Testsoftware Manual

MDG1

TSW-Version: 1.1.46.0

DGS-EC/EHT5

Confidential

Testsoftware Manual:

MDG1

TSW-Version: 1.1.46.0

DGS-EC/EHT5

Confidential

Table of Contents

List of Figures

List of Tables

General Notes

Reject Codes of BootCtrl-Services

1	2	3
RRP	**SID**	**RRC**
7F	...	...

Details

Name	Length	Fmt	Description
RRP	**1**	**xx**	**Reject response prefix**
SID	**1**	**xx**	**Service ID of request**
RRC	**1**	**xx**	**Reject response code**

Code	Description
0x10	General Reject
0x11	Service Not Supported
0x12	SubService not supported
0x22	Conditions not correct
0x31	Out of range
0x33	Security Access Denied
0x40	Download not accepted
0x41	Downloaded TSW is to old
0x43	Can't Download number of Bytes requested
0x72	Transfer Aborted
0x90	Invalid byte count
0x91	One or more arguments have invalid values
0x92	Signature invalid

Representation of values

All parameter values of transmit and receive data are represented in hexadecimal or binary values (marked as bitcoded, eg. b10 which represents the value 0x2).

Values in command or response strings, which need more than one byte, are always represented in high-to-low byte order, eg. the bytes of the value 0x123456 are counted starting with the high-order byte 0x12 followed by 0x34 and then 0x56 which is for example byte 4, 5 and 6 of a response string:

3	4	5	6	7
…	12	34	56	…

If certain bit positions of a value which embraces more than one byte, are explained in detail, the bit positions always refer to the 'natural' representation, i.e. the bit 0 corresponds to the most right bit (weight 2^0).

Table 1, “Comparison of byte and bit numbering within byte strings” illustrates the byte and bit numbering of the example value 0x123456 which starts on byte 4 within a communication string.

Table 1. Comparison of byte and bit numbering within byte strings

Bit position	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit value	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	0	1	0	1	0	1	1	0
Hex value	1				2				3				4				5				6			
Byte position	4								5								6							

TSW-Version information in PAV

The TSW version is stored as a string and as an integral value in TSW container readable from PAV by function

ECUMemDataStr(TSW, DATA, PARAMETER, VERSIONSTRING, s, 0)

to get the TSW-Version as string (eg. 1.2.3.4) valuerange 0 .. 255 each.

ECUMemDataStr(TSW, DATA, PARAMETER, VERSION, i, 0)

to get the TSW-Version as integer value (eg. 1002003004).

Resolution indicator of physical values

Representation

The upper 4 bit of a hex value given as TSW parameter indicates the resolution of the physical value. The user is free to choose a resolution according to his needs. The only limitation is the value range as described in the TSW documentation. The length of the value (8, 16, 24, 32 bit) depends on the TSW usage and is shown in the documentation. Returned physical values (e.g. time) use always a fixed resolution indicator as described in the TSW documentation. Commonly used physical values in TSW parameters are typically time, baudrate, frequencies, etc.

Table 2. Composition of resolution indicator within a 16 bit value

Bit															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R	R	R	R	V	V	V	V	V	V	V	V	V	V	V	V

R = 4bit: resolution indicator

V = remaining bits: value

Table 3. Resolution indicator coding

decimal	hex	resolution factor		
0	0	10^0	1	
1	1	10^1	10	
2	2	10^2	100	
3	3	10^3	1.000	1K

decimal	hex	resolution factor		
4	4	10^4	10.000	10K
5	5	10^5	100.000	100K
6	6	10^6	1.000.000	1M
7	7	10^{-9}	0,000.000.001	1n
8	8	10^{-8}	0,000.000.01	10n
9	9	10^{-7}	0,000.000.1	100n
10	A	10^{-6}	0,000.001	1µ
11	B	10^{-5}	0,000.01	10µ
12	C	10^{-4}	0,000.1	100µ
13	D	10^{-3}	0,001	1m
14	E	10^{-2}	0,01	10m
15	F	10^{-1}	0,1	100m

Examples:

phys. value	=	value	*	10^x	=>	R	V	=>	value to transmit
380µs	=	380	*	10^{-6}	=>	0xA	0x17C	=>	0xA17C
380µs	=	38	*	10^{-5}	=>	0xB	0x26	=>	0xB026
1.000.000bps	=	1000	*	10^3	=>	0x3	0x3E8	=>	0x33E8
1.000.000bps	=	1	*	10^6	=>	0x6	0x1	=>	0x6001

Trap system error message

```
description now only for IFX
7F FF 02 : system error message "Trap occured"
90       : TSW
04       : Trap-ID
01       : TIN
Start lower context :
00 1D 00 0C : PCXI
00 00 00 00 : A11 (Trap address)
FF FF FF FF : A2
D8 00 44 A8 : A3
FF FF FF CD : D0
00 00 00 00 : D1
00 00 00 01 : D2
00 00 00 02 : D3
D8 00 44 AC : A4
D8 00 44 B8 : A5
D8 00 44 C4 : A6
D8 00 44 AC : A7
00 00 07 03 : D4
00 00 00 00 : D5
10 62 4D D3 : D6
ED B8 83 20 : D7
Start Upper Context
00 1D 00 0D : PCXI
08 00 0B 8B : PSW
```

```
D8 00 44 A8 : A10 (Stackpointer)
D8 00 4D 9C : A11 (Returnaddress)
00 00 80 00 : D8
00 00 00 03 : D9
00 20 80 01 : D10
00 00 00 00 : D11
D8 00 4D 3C : A12
D8 00 47 72 : A13
D8 00 45 D4 : A14
00 00 00 00 : A15
00 00 20 10 : D12
00 00 00 04 : D13
00 00 60 00 : D14
00 00 00 05 : D15



description now only for JDP
7F FF 02 : system error message "Trap occured"
90       : TSW
Some Core register:
00 1D 00 0C : ESR
00 00 00 00 : MSCR
FF FF FF FF : MCSRR0
D8 00 44 A8 : MCSRR1
FF FF FF CD : SRR0
00 00 00 00 : SRR1
00 00 00 01 : CSRR0
00 00 00 02 : CSRR1
Start Upper Context
D8 00 44 AC : R16
D8 00 44 B8 : R17
D8 00 44 C4 : R18
D8 00 44 AC : R19
00 00 07 03 : R20
00 00 00 00 : R21
10 62 4D D3 : R22
ED B8 83 20 : R23
00 1D 00 0D : R24
08 00 0B 8B : R25
D8 00 44 A8 : R26
D8 00 4D 9C : R27
00 00 80 00 : R28
00 00 00 03 : R29
00 20 80 01 : R30
00 00 00 00 : R31
00 00 00 00 :
00 00 00 00 :
00 00 00 00 :
00 00 00 00 :
00 00 00 00 :
00 00 00 00 :
00 00 00 00 :
00 00 00 00 :
```

Overview of GTM resources used by TSW/ BootCtrl

The table below gives an overview of TSWs using Generic Timer Module (GTM) resources and lists which GTM resources are used by the respective TSW or by BootCtrl.

TIM ... Timer Input Module

TOM ... Timer Output Module

ATOM ... ARU-connected Timer Output Module

Ch ... Channel (of the respective module)

Table 4. Overview of GTM resources used by TSW/BootCtrl

BC/TSW	GTM res	Description
BootCtrl	ATOM[0].Ch[0]	ATOM0_0
TSW_05	TIM[X].Ch[Y] TOM[X].Ch[Y] ATOM[X].CH[Y]	X ... selected by user; Y ... selected by user
TSW_09	TOM[X].Ch[Y] ATOM[X].CH[Y]	X ... selected by user; Y ... selected by user
TSW_75	ATOM[C].Ch[D] (3 resources)	C ... selected by user; D ... selected by user; Note: There are 3 ATOM resources used within this TSW
TSW_C0	TOM[B].Ch[0-7] or TOM[0].CH[0-7] TOM[1].CH[0-7]	B ... 0 or 1 or both; All channels (0-7) of the TOM[B] (or of TOM[0] and TOM[1]) are occupied and CANNOT be used by other TSWs
TSW_C3	TIM[X].Ch[Y] (1 resource)	X ... selected by user; Y ... selected by user; (one out of TIM[0..3].Ch[0..7])
TSW_C4	TIM[X].Ch[Y] (1 resource)	X ... selected by user; Y ... selected by user; (one out of TIM[0..3].Ch[0..7])
TSW_C7	TOM[X].Ch[Y] ATOM[X].Ch[Y] (1 resource)	X ... selected by user; Y ... selected by user;
TSW_CA	TOM[X].Ch[Y] ATOM[X].Ch[Y] (up to 7 resources)	X ... selected by user; Y ... selected by user;
TSW_CB	TIM[X].Ch[Y] and ATOM[Z].Ch[A] and MCS[B].Ch[C]	X,Z,B ... selected by user; Y,A,C ... selected by user;

BC/TSW	GTM res	Description
TSW_D5	TIM[A].Ch[A] (up to 4 resources)	A ... automatic; (respective to source available on used pin).
TSW_D6	TIM[A].Ch[A] (up to 4 resources)	A ... automatic; (respective to source available on used pin).
TSW_D8	TIM[A].Ch[A] (up to 2 resources)	A ... automatic; (respective to source available on used pin). Workaround for JDP (cut1.x)
TSW_DF	TIM[A].Ch[A] (up to 4 resources)	A ... automatic; (respective to source available on used pin).
TSW_DF	TOM[A].Ch[A] (up to 4 resources)	A ... automatic; (respective to source available on used pin).
TSW_DF	ATOM[A].Ch[A] (up to 4 resources)	A ... automatic; (respective to source available on used pin).

Description of test mode

Test mode activation

The ECU can be switched to test mode by sending defined patterns at the CAN lines of the ECU, called "CAN-ESB"'s. The patterns do not represent a valid CAN message for avoiding an unintentional activation of STIL routines by normal CAN communication. The patterns are transmitted without any CAN frame (ASC@CAN).

STIL entry cycle with CAN-ESB

- Transmission of CAN-ESB pattern must be started before ECU is switched on. The patterns must be repeated periodically.
- The STIL part at boot control (BootCtrl) of ECU software scans the CAN line for the patterns.
- The receiption of valid patterns is answered by the ECU with a defined response (acknowledge).
- When no valid patterns were received within a defined time the execution of normal startup procedure is continued.
- After successful detection of valid entry condition, the ECU waits for further communication in KP2000 protocol (e.g. Start Communication Request).

Exit from STIL routines

- While STIL routines are executed, the correct entry via entry conditions is checked periodically. When invalid entry is detected, STIL routine is left via reset.
- No return from STIL routines to application software possible. Always restart with PowerOn reset is necessary.
- Therefore no effect at application software by remains of STIL routines (e.g. at RAM) possible.

Baud rate of CANESB's: 250kbps.

example CfiTerm:

COMINIT RAW,SERIAL,NO,250000,0,1,0,0,0,0

CANESB MDG1,TSW_1,8,10000

COMINIT RAW,SERIAL,NO,250000,20,100000,0,0,50,25

Test mode communication

Hardware implementation

Communication in test mode is done using a serial communication protocol. Implemented is ASC over CAN: differencial connection with 2.5V idle level and 0V/5V active level (ASC@CAN)

The differencial connection can be used with speeds up to 5Mbps [1] . Default baud rate is same as for CANESB's: 250kbps.

The serial protocol has following definition:

- 8 databits
- 1 stopbit
- no parity

Software implementation

The test mode interpreter uses the KWP2000 [2] protocol for communicating with the world (eg. tester). A master (eg. tester) sends a service request and the slave (ECU) has to answer this request with its response. The slave may not start a transmission on its own.

For first programming a special low-level protocol is implemented which is used when the ECU is in bootstrap mode (IFX) / serialboot mode (JDP/FSL) [3] . That is, when no valid software is found in flash during startup. It is used to download the operating system for the first-programming which uses then KWP2000 communication protocol.

KWP2000 protocol implementation

The actual implementation of KWP2000 uses a serial communication interface as described in the section called “Hardware implementation”. Each transmission consists of a requestframe or responseframe, depending on the direction.

KWP2000 supports different frame formats but only those which are shown in Table 1, “Supported KWP2000 Frame Formats” (the "•" symbol marks used bytes):

Table 1. Supported KWP2000 Frame Formats

Byte:	FMT	LEN	TGA	SRA	SID	
	81		FE	F1	81	Start communication request
	00	•			•	KWP2000 format used in TSW communication after Start communication request has been executed.

Byte description

FMT Format byte, describes the frame format and contains the framelength in case, if no additional length byte is used.

LEN Additional length byte; extends the framelength up to 255 payload bytes.

TGA Target address of slave, 0xFE means: all slaves

SRA Source address of master: must be 0xF0 or 0xF1 in test mode

SID Service identifier; contains the service number to be requested by the master or the SID+0x40 in the response frame of the slave.

KWP2000 communication timing

KWP2000 defines the timing parameters as shown in Table 2, “KWP2000 Timing Parameters” and Figure 1, “Illustration of the KWP2000 timing parameters”. The values which

[1] depending on the cpu clock, the maximum communication speed can be different
[2] KeyWord Protocol 2000; see ISO-14230
[3] refer to the cpu manuals for detailed information

are valid after the 'start communication'-request are shown in Table 3, “KWP2000 Timing Parameters - Default values”.

Table 2. KWP2000 Timing Parameters

Parameter	Description
P1	Time between 2 consecutive bytes within the transmitted data of the ECU.
P2	Time between tester request and ECU response.
P3	Time between the end of the ECU response and a consecutive tester request.
P4	Time between 2 consecutive bytes within the transmitted data of the tester.

Figure 1. Illustration of the KWP2000 timing parameters

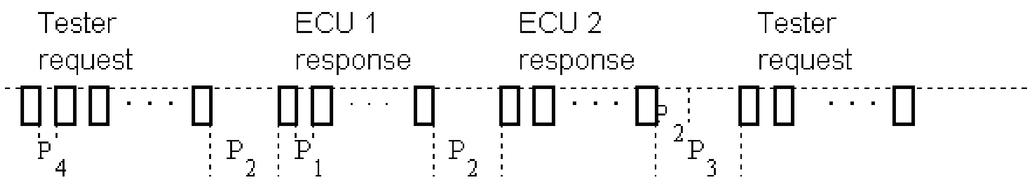


Table 3. KWP2000 Timing Parameters - Default values

Parameter	Default value	Remark
P1min	0 ms	ECU transmits a new byte as soon as the transmitter signals that it is ready. The time needed is typically less than 10 µs.
P1max	20 ms	
P2min	1 ms	Deviation to KWP2000 standard, because for slow communication hardware.
P2max	1000 ms	
P3min	0 ms	
P3max	5000ms	
P4min	5 ms	The ECU only checks for P4max.
P4max	20 ms	

Implemented KWP2000 Services

Implemented services

- ECUReset (ID: 0x11)
- ReadDataByLocalIdentifier (ID: 0x21)
- SecurityAccess (ID 0x27)
- TransferData (ID 0x36)
- WriteDataByLocalIdentifier (ID 0x3B)
- EscapeCode (ID 0x80)

- StartCommunication (ID 0x81)
- StopCommunication (ID 0x82)

Reset ECU (Service ID: 0x11)

information follows ...

Command

0	1
SID	**...**
11	...

Parameters

1	+0
(1)	RSTMODE

Details

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>RSTMODE</td><td>1</td><td>xx</td><td>Reset mode
<table>
<tr><td>0x00</td><td>Systemreset</td></tr>
<tr><td>0x01</td><td>Reset and start factory test mode without CANESB</td></tr>
<tr><td>0x02</td><td>Destructive Reset</td></tr>
</table>
</td></tr>
</table>

Read data by local identifier (Service ID: 0x21)

information follows ...

Command

0	1
SID	**...**
21	...

Parameters

subfunction service ID

1	+0
(1)	SSID

subfunction service ID data

2	+0
(*)	CMDDATA

Details

Name	Length	Fmt	Description
SSID	**1**	**xx**	**Subfunction Service-ID** refer to the section called “subfunction service ID's of RDBLID” for details.
CMDDATA	**1**	**xx..**	**Data** if necessary for specific subfunction.

Response

see response sub chapters of the section called “subfunction service ID's of RDBLID”

subfunction service ID's of RDBLID

Available sub ID's

ID	description
00	number of ID's
01	read controller ID
02	read controller serial number
03	read BootCtrl version
04	list of SSID's of loaded TSW services
05	read start address of specific TSW service
06	read state info of DMA and SCI
07	read version number of specific TSW service
08	Power On Selftest
09	ECUSecu Version
0A	Info about download RAM Usage
0B	Get Ecu mode
0C	Get BootCtrl state
0D	Get security access type
0E	Read cpu related informations

number of ID's (SID:0x21 Sub Service ID: 0x00)

returns number of available subfunction services

sub command

1
SSID
00

Response

1	2
SVC	**...**
61	...

Parameters

2	+0
(1)	NUMID

Details

Name	Length	Fmt	Description
NUMID	**1**	**xx**	**number of local ID's** follow-up chapters describes available local sub functions

read controller ID (SID:0x21 Sub Service ID: 0x01)

read specific controller identification registers

sub command

1
SSID
01

Response

1	2
SVC	**...**
61	...

Parameters

2	+0	+1	+5
(1)	MANU	CPUID1	CPUID2

Details

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>MANU</td><td>1</td><td>xx</td><td>manufacturer
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x01</td><td>IFX</td></tr>
<tr><td>0x02</td><td>JDP</td></tr>
</table>
FSL: not exist</td></tr>
<tr><td>CPUID1</td><td>4</td><td>xxxxxxxx</td><td>CPU specific ID 1
<table>
<tr><th></th><th>Register</th><th>Description</th></tr>
<tr><td>IFX</td><td>SCU.CHIPID</td><td>Chip Identification Register</td></tr>
<tr><td>JDP</td><td>SIUL2.MIDR1</td><td>SIUL2 MCU ID Register #1</td></tr>
<tr><td>FSL</td><td>SIU.MIDR</td><td>MCU ID register</td></tr>
</table>
</td></tr>
</table>

Name	Length	Fmt	Description
CPUID2	**4**	**xxxxxxxx**	**CPU specific ID 2** | | Register | Description | IFX: SCU.MANID — Manufacturer Identification Register JDP: SIUL2.MIDR2 — SIUL2 MCU ID Register #2 FSL: not exist

read controller serial number (SID:0x21 Sub Service ID: 0x02)

read controller seriennummer from a specific memory area

sub command

1
SSID
02

Response

1	2
SVC	**...**
61	...

Parameters

2	+0
(1)	CPUSN

Details

Name	Length	Fmt	Description
CPUSN	**16/32**	**xxxxxxxx...**	**seriennummer** **Device** — **Description** IFX, FSL: read 16 byte from addresse 0xAF101000 JDP: read 32 byte from addresse 0x004000A0

read BootCtrl version (SID:0x21 Sub Service ID: 0x03)

read BootCtrl version

sub command

1
SSID
03

Response

BootCtrl version number

1	2
SVC	**...**
61	...

Parameters

2	+0	+1	+2	+3	+4	+5	+6	+7
(1)	RSRVD	BCVER	BCREV	BCFIX	RSRVD	MODVER	MODREV	MODFIX

...	+8	+9	+10	+11	+12
	VARID	RSRVD	INTVER	INTREV	INTFIX

Details

Name	Length	Fmt	Description
RSRVD	**1**	**xx**	**0x00 (reserved)**
BCVER	**1**	**xx**	**BootCtrl Version Id (BASD)**
BCREV	**1**	**xx**	**BootCtrl Revision Id (BASD)**
BCFIX	**1**	**xx**	**BootCtrl Bugfix Id (BASD)**
RSRVD	**1**	**xx**	**0x00 (reserved)**
MODVER	**1**	**xx**	**TSW Module Version Id**
MODREV	**1**	**xx**	**TSW Module Revision Id**
MODFIX	**1**	**xx**	**TSW Module Bugfix Id**
VARID	**1**	**xx**	**BootCtrl Variant ID**
RSRVD	**1**	**xx**	**0x00 (reserved)**
INTVER	**1**	**xx**	**BootCtrl Interface Version**
INTREV	**1**	**xx**	**BootCtrl Interface Revision**
INTFIX	**1**	**xx**	**BootCtrl Interface Bugfix**

list of SSID's of loaded TSW services (SID:0x21 Sub Service ID: 0x04)

returns ID's of loaded teststeps

sub command

1
SSID
04

Response

sub service ID's of loaded TSW's

1	2
SVC	**...**
61	...

Parameters

Number of bytes of the response corresponds to the number of TSW Services in DL RAM

2	+0	...
(1)	TSWID	...

Details

Name	Length	Fmt	Description
TSWID	**1**	**xx**	**TSW Service in DL RAM**

read start address of specific TSW service (SID:0x21 Sub Service ID: 0x05)

returns start address of specific TSW service in DL RAM

sub command

1	2
SSID	**...**
05	...

sub parameters

2	+0
(1)	TSWID

sub Details

Name	Length	Fmt	Description
TSWID	**1**	**xx**	**TSW identification**

Response

start address (4 bytes) [HEX] in RAM of loaded TSW

1	2
SVC	**...**
61	...

Parameters

2	+0
(1)	ADDR

Details

Name	Length	Fmt	Description
ADDR	**4**	**xxxxxxxx**	**Start address of TSW Service in DL RAM**

read state info of DMA and SCI (SID:0x21 Sub Service ID: 0x06)

read state info from specific register

sub command

1
SSID
06

Response

1	2
SVC	**...**
61	...

Parameters

2	+0	+4	+8	+9
(1)	LINES	DMAES	LSTERR	ERRCNT

Details

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>LINES</td><td>4</td><td>xxxxxxxx</td><td>LIN error status
<table>
<tr><th></th><th>Register</th><th>Description</th></tr>
<tr><td>IFX</td><td>FLAGS</td><td>ASCLIN Flags Register</td></tr>
<tr><td>JDP</td><td>LINESR</td><td>LIN Error Status Register</td></tr>
<tr><td>FSL</td><td>ESCI.SR</td><td>Interrupt Flag and Status Register</td></tr>
</table></td></tr>
<tr><td>DMAES</td><td>4</td><td>xxxxxxxx</td><td>DMA error status
<table>
<tr><th></th><th>Register</th><th>Description</th></tr>
<tr><td>IFX</td><td>DMA.ERRSR</td><td>Error Status Register</td></tr>
<tr><td>JDP</td><td>DMA.ES</td><td>Error Status Register</td></tr>
<tr><td>FSL</td><td>EDMA.ESR</td><td>Error Status Register</td></tr>
</table></td></tr>
<tr><td>LSTERR</td><td>1</td><td>xx</td><td>Type of last communication error</td></tr>
<tr><td>ERRCNT</td><td>7</td><td>xxxxxxxx xxxxxx</td><td>Number of occurred communication errors according to type</td></tr>
</table>

read version number of specific TSW service (SID:0x21 Sub Service ID: 0x07)

read version information of the specific TSW service.

sub command

1	2
SSID	**...**
07	...

sub parameters

2	+0

(1)	TSWID

sub Details

Name	Length	Fmt	Description
TSWID	**1**	**xx**	**TSW identification**

Response

1	2
SVC	**...**
61	...

Parameters

2	+0	+4	+6
(1)	VERSION	INT	MOD

Details

Name	Length	Fmt	Description
VERSION	**4**	**xxxxxxxx**	**Version of TSW service** The version is 4 digits long. aabbccdd = aa.bb.cc.dd . The first digit is not used. This version is equal the delivery version. It is 00.00.00.00 if the TSW service was not delivered officially.
INT	**2**	**xxxx**	**Integration number** Number of the last integration.
MOD	**1**	**xx**	**Modification identifier** This Byte is 0xAA in case of any modified files not available in Version Control.

Power On Selftest (SID:0x21 Sub Service ID: 0x08)

Start Power On Selftest and get result.

sub command

1	2
SSID	**...**
08	...

sub parameters

2	+0
(1)	Mask

sub Details

Name	Length	Fmt	Description
Mask	**1**	**xx**	**Controls the depth of the Power On Selftest.** If it is not defined a full scan will be performed. Values in table can be linked logically(OR). **Value** / **Function** 0x01 — Check blocks type "after flash process" 0x02 — Check blocks type "in the Ini Task" 0x04 — Check blocks type "cyclic in the drive Task" 0x08 — Check blocks type "in the PostDrive Task" 0x10 — Check blocks type "SB" 0x20 — Check blocks type "MAC at secure boot" 0x40 — Check blocks type "MAC before any RB programming is started" 0x80 — Check blocks type "MAC before any CB programming is started"

Response

1	2
SVC	**...**
61	...

Parameters

2	+0	+4	+4	+4
(1)	BLOCK	ADR	STATE	ERROR

Details

Name	Length	Fmt	Description
BLOCK	**4**	**xxxxxxxx**	**Block ID** The id of the block with errors
ADR	**4**	**xxxxxxxx**	**Block start address** The start address of the block with errors
STATE	**4**	**xxxxxxxx**	**State flags** Status flag of tests they have been executed. Correspond to the Mask bits.
ERROR	**4**	**xxxxxxxx**	**Error flags** **Value** / **Description** 0x00000001 — LinkChain is in an endless loop.

Name	Length	Fmt	Description	
			Value	**Description**
			0x00000002	Incorrect Prolog CRC
			0x00000004	Incorrect valid pattern
			0x00000008	Wrong value in checksum block
			0x00000010	Wrong compatibility value in linked block
			0x00000020	Incorrect address in compatibility link chain
			0x00000040	Incorrect address to a checksum block
			0x00000080	Incorrect address in link chain
			0x00000100	Wrong number of checksum entries
			0x00000200	Checksum algorithm not supported
			0x00000400	Incorrect address of Prolog lays for valid pattern check
			0x00000800	Incorrect address of Prolog lays for CRC check
			0x00001000	Incorrect Prolog end address
			0x00002000	ECC failure in Prolog or Epilog

read ECUSecu Version (SID:0x21 Sub Service ID: 0x09)

ECUSecu version number

sub command

1
SSID
09

Response

BootCtrl version number

1	2
SVC	**...**
61	...

Parameters

2	+0	+1	+2	+3
(1)	VER	REV	BUGFIX	CPU_VER

Details

Name	Length	Fmt	Description
VER	**1**	**xx**	**EcuSecu Version Id (BASD)**

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>REV</td><td>1</td><td>xx</td><td>EcuSecu Revision Id (BASD)</td></tr>
<tr><td>BUGFIX</td><td>1</td><td>xx</td><td>EcuSecu Bugfix Id (BASD)</td></tr>
<tr><td>CPU_VER</td><td>1</td><td>xx</td><td>Microcontroller type
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x00</td><td>IFX DEV2</td></tr>
<tr><td>0x01</td><td>IFX DEV3</td></tr>
<tr><td>0x02</td><td>IFX DEV3B</td></tr>
<tr><td>0x03</td><td>IFX DEV3C</td></tr>
<tr><td>0x04</td><td>IFX DEV4</td></tr>
<tr><td>0x10</td><td>JDP DEV2</td></tr>
<tr><td>0x11</td><td>JDP DEV2C2</td></tr>
<tr><td>0x12</td><td>JDP DEV3</td></tr>
<tr><td>0x13</td><td>JDP DEV3C2</td></tr>
<tr><td>0x14</td><td>JDP DEV4</td></tr>
<tr><td>0x15</td><td>JDP DEV4C2</td></tr>
</table>
</td></tr>
</table>

Info about download RAM Usage (SID:0x21 Sub Service ID: 0x0A)

info about available and free RAM size

sub command

1
SSID
0A

Response

1	2
SVC	**...**
61	...

Parameters

2	+0	+4	+8
(1)	AVAILABLE	FREE	LRG_BLK

Details

Name	Length	Fmt	Description
AVAILABLE	**4**	**xxxxxxxx**	**Size of available download RAM**
FREE	**4**	**xxxxxxxx**	**Size of free download Ram**
LRG_BLK	**4**	**xxxxxxxx**	**Size of largest free RAM block**

Get Ecu mode (SID:0x21 Sub Service ID: 0x0B)

Get Ecu Mode

sub command

1
SSID
0B

Response

1	2
SVC	**...**
61	...

Parameters

2	+0	+1	+2	+3
(1)	STATE_HSM	STATE_SRV	STATE_EXT	ECU_MODE

Details

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>STATE_HSM</td><td>1</td><td>xx</td><td>State
<table>
<tr><th>value</th><th>state</th></tr>
<tr><td>0x00</td><td>STHSM_IDLE
or not available</td></tr>
<tr><td>0x01</td><td>STHSM_INIT</td></tr>
<tr><td>0x02</td><td>STHSM_ACTIVE</td></tr>
<tr><td>0x03</td><td>STHSM_FINAL</td></tr>
<tr><td>0x04</td><td>STHSM_PROG_ASW</td></tr>
<tr><td>0x05</td><td>STHSM_PROG_HSM</td></tr>
</table></td></tr>
<tr><td>STATE_SRV</td><td>1</td><td>xx</td><td>State
<table>
<tr><th>value</th><th>state</th></tr>
<tr><td>0x00</td><td>STSRV_IDLE
or not available</td></tr>
<tr><td>0x01</td><td>STSRV_INIT</td></tr>
<tr><td>0x02</td><td>STSRV_ACTIVE</td></tr>
<tr><td>0x03</td><td>STSRV_FINAL</td></tr>
<tr><td>0x04</td><td>STSRV_STOPPED</td></tr>
</table></td></tr>
<tr><td>STATE_EXT</td><td>1</td><td>xx</td><td>State
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7-4</td><td>not used, is set to 0</td></tr>
<tr><td>3</td><td>CFGMISMATCH: 1 if an EcuSecu configuration mismatch was detected, 0 otherwise</td></tr>
<tr><td>2</td><td>TRUSTED: 1 if the Trusted mode is</td></tr>
</table></td></tr>
</table>

Name	Length	Fmt	Description	
			Bit(s)	**Description**
				active (software is authenticated), 0 otherwise
			1	TSTSER: 1 if the Test Serial mode is active, 0 otherwise
			0	SWOVR: 1 if the switch-over mode is active, 0 otherwise
ECU_MODE	**1**	**xx**	**State**	
			value	**state**
			0x00	Application
			0x01	Serial

Get BootCtrl state (SID:0x21 Sub Service ID: 0x0C)

Get BooCtrl state

sub command

1
SSID
0C

Response

1	**2**
SVC	**...**
61	...

Parameters

2	+0	
(1)	STATE	

Details

Name	Length	Fmt	Description	
STATE	**1**	**xx**	**State**	
			value	**state**
			0x00	default
			0x01	BootCtrl is running on "Copy Page"

Get security access type (SID:0x21 Sub Service ID: 0x0D)

Get security access type.

In the EcuSecu can be configured which security access is at least needed to get access to STIL mode. This can be different for serial ECUs and application ECUs. The results of needed algorithm are ordered by their weight. The weakest type is listed on the first position.

sub command

1
SSID
0D

Response

1	2
SVC	**...**
61	...

Parameters

2 (1..n*)	+0	+1
	ALGO	KEYID

Details

Name	Length	Fmt	Description
ALGO	**1**	**xx**	**Algo Type** value: 0x00 = MDG1 Seed & Key value: 0x02 = BBM Security Access value: 0xFF = No EcuSecu found. Crypto Security Access not active.
KEYID	**1**	**xx**	**KeySlot ID** KeySlot ID

Read cpu related informations (SID:0x21 Sub Service ID: 0x0E)

Read cpu related informations.

sub command

1
SSID
0E

Response

1	2
SVC	**...**
61	...

Parameters

2	+0	
(1)	VERSION	

Details

Name	Length	Fmt	Description
VERSION	**4**	**xxxxxxxx**	**VERSION** **device** / **version** IFX / undef JDP/ST / BAF Version

Security access (Service ID: 0x27)

Request Seed

Command

0	1
SID	**...**
27	...

Parameters

1	+0	+1
(1)	LEVEL	ROUNDS

Details

Name	Length	Fmt	Description
LEVEL	**1**	**xx**	**Required Access level** This value selects the required access privilege level and must be odd. If granted, TSW which is classified by a certain privilege level, will be allowed to be executed. **Level** / **Description** 0x01 / Generic access privileges (no flash programming, eeprom access, ...) ... / Privilege level for certain customer specific TSWs 0x7F / Privilege level for RB factory access
ROUNDS	**1**	**xx**	**Number of cypher rounds**

Response

1	2
SVC	**...**
67	...

Parameters

2	+0	+1
(1)	LEVEL	SEED

Details

Name	Length	Fmt	Description
LEVEL	**1**	**xx**	**Requested access level**
SEED	**4...20**	**xxxxxxxx...**	**Seed values** Depending on the required access level, the number of received seed bytes is either 4 or 20.

Send Key

Command

0	1
SID	**...**
27	...

Parameters

1	+0	+1
(1)	LEVEL+1	KEY

Details

Name	Length	Fmt	Description
LEVEL+1	**1**	**xx**	**Required access level + 1** Must be an even value.
KEY	**4...128**	**xxxxxxxx...**	**Key** According to the the requested access level, the key has either a length of 4 bytes or 128 bytes.

Response

1	2
SVC	**...**
67	...

Parameters

2	+0	+1
(1)	LEVEL+1	AGA

Details

Name	Length	Fmt	Description
LEVEL	**1**	**xx**	**Requested access level + 1**

Name	Length	Fmt	Description
AGA	**1**	**xx**	**Access Granted Acknowledge** 0x34

Transfer data (Service ID: 0x36)

Command

0	1
SID	**...**
36	...

Parameters

1 (1)	+0	+1
	BLKCNT	DATA

Details

Name	Length	Fmt	Description
BLKCNT	**1**	**xx**	**information follows ...**
DATA	**1**	**xx..**	**Data** to download

Response

1	2	3	4
SVC	**ESVC**	**ETSW**	**VER**
76	xx	00	01

Details

ESCV	Description
00	Successful.
04	Init function returns error.
05	Not supported TSW type.
06	CRC failure.
07	Wrong CPU family.
08	HAL / TSW version mismatch.
09	Signature invalid.
0A	Certificate invalid.

Write data by local identifier (Service ID: 0x3B)

Command

0	1
SID	**...**
3B	...

Parameters

subfunction service ID

1	+0
(1)	SSID

subfunction service ID data

2	+0
(*)	CMDDATA

Details

Name	Length	Fmt	Description
SSID	**1**	**xx**	**Subfunction Service-ID** refer to the section called “subfunction service ID's of WDBLID” for details.
CMDDATA	**1**	**xx..**	**Data** for specific subfunction, if necessary.

subfunction service ID's of WDBLID

ID	description
00	number of ID's
01	set baud rate
02	Free Memory by service ID
03	Set Ecu Mode "Serial"

number of ID's

returns number of aviable subfunction services

sub command

1
SSID
00

Response

number of aviable subfunction services

1	2
SVC	**...**
7B	...

Parameters

1	+0
(1)	NUM

Details

Name	Length	Fmt	Description
NUM	**1**	**xx**	**Number of available ID's**

set baud rate

Set ECU baud rate for communication

Table 4. Proposed baud rates for JDP (in baud)

250000	320000	400000	500000	640000	800000	1000000	1250000
1600000	2000000	2500000	3200000	4000000	5000000	8000000	

Table 5. Proposed baud rates for IFX (in baud)

290000	580000	760000	870000	1000000	1450000	1850000	2000000
2500000	2900000	3000000	3700000	3800000	4000000	4200000	4800000
5000000	6000000	7000000	8000000				

sub command

1	2
SSID	**...**
01	...

sub parameters

1	+0
(1)	BAUD

sub Details

Name	Length	Fmt	Description
BAUD	**1**	**xxxxxxxx**	**Baud rate** to set. Resolution: BAUD = (value * 10^0)[HEX].

Response

New baud rate will be set from ECU after this response. Baud rate = return value[HEX] * 10^0.

1	2
SVC	**...**
7B	...

Parameters

1	+0	+1
(1)	RSVD	BAUD

Details

Name	Length	Fmt	Description
RSVD	**1**	**xx**	**0x00 (reserved)**
BAUD	**4**	**xxxxxxxx**	**new baud rate**

Free Memory by service ID

Free the Memory used by a service. The memory can be used by another service after deletion.

Available with BootCtl Version 1.4.3

sub command

1	2
SSID	**...**
02	...

sub parameters

1	+0
(1)	ServiceID

sub Details

Name	Length	Fmt	Description
ServiceID	**1**	**xx**	**Service ID** Delete the specified service and free the memory.

Response

code	description
00	Deletion successful
01	Service ID not found

Set Ecu Mode "Serial"

Set Ecu Mode virtual to "Serial" to activate certificate and signature check on ECUs with HSM.

Available with BootCtl Version 1.6.2

sub command

1	2
SSID	**...**
03	...

Escape code (Service ID: 0x80)

Command

0	1
SID	**...**
80	...

Parameters

1	+0	+1
(1)	DLSID	DLPARA

Details

Name	Length	Fmt	Description
DLSID	**1**	**xx**	**Service-ID of Download-TSW**
DLPARA	**1**	**xx..**	**Parameter** for Download-TSW (refer to specific TSW)

Response

refer to specific TSW documentation

Start communication (Service ID: 0x81)

The KWP2000 service “Start communication Request” is needed to initially connect to an ECU. The initial baud rate depends on the used physical communication layer:
K-Line 10400bps (according to ISO standard)
ASCCAN 57600bps (deviation to ISO standard because of dominant timeout supervision of CAN transceivers)

Command

0
SID
81

Parameters

1	+0	+1	+2	+3
(1)	TADR	SADR	SID	CHS

Details

Name	Length	Fmt	Description
TADR	**1**	**xx**	**Target Address** TADR specifies the target address of the ECU which shall be connected. Must be set to 0xFE.

Name	Length	Fmt	Description
SADR	**1**	**xx**	**Source Address** SADR specifies the source address of the communication master (e.g tester). This byte must be set to one of 0xF0. . . 0xF1.
SID	**1**	**xx**	**Service ID:** Start Communication Request 0x81
CHS	**1**	**xx**	**Frame Checksum**

Response

1	2
SVC	**...**
00	...

Parameters

2	+0	+1	+2	+3	+4
(1)	LEN	RES	KEY1	KEY2	CHS

Details

Name	Length	Fmt	Description
LEN	**1**	**xx**	**Length byte**
RES	**1**	**xx**	**Service ID + 0x40** 0xC1: Start Communciaton Request – positive response
KEY1	**1**	**xx**	**Key Byte 1** Refer to ISO14230–2 for a detailed description.
KEY2	**1**	**xx**	**Key Byte 2** Refer to ISO14230–2 for a detailed description.
CHS	**1**	**xx**	**Frame Checksum**

Stop Communication (Service ID: 0x82)

This service is used to stop an established communication to an ECU. The ECU will remain in factory test mode, but certain running services may be stopped (eg. monitoring service).

Command

0
SID
82

Response

Positive response: Service ID + 0x40

1
SVC
C2

TSW_01 - ADC conversion

Test step description

Date: 01.06.2016

Initialisation of the onchip ADC modules, configuration of the channels to be converted, start a conversion and read conversion results.

Table 1. Analog channel mapping for IFX 65nm (Group 8-10 only available for Dev4)

ADC Group	Channel	Virtual Channel(CHA)
Group 0 - 5	AN00 - AN47	0 - 47
Group 6 - 6	P0.12 - P0.07	176 - 181
Group 7 - 7	P0.06 - P0.01	184 - 189
Group 8 - 10	AN48 - AN71	48 - 71

Table 2. Analog channel mapping for IFX 40nm (Dev5)

ADC Group	Channel	Virtual Channel(CHA)
Group 0 - 3	AN00 - AN31	0 - 31
Group 4 - 4	--- NOT USED ---	--- NOT USED ---
Group 5 - 7	AN48 - AN71	48 - 71
NOT USED	--- NOT USED ---	72 - 79
Group 8 - 8	AN32 - AN47	32 - 47
Group 9 - 9	P0.12 - P0.1 / P1.5 - P1.3 / P2.11	208 - 223 (80 - 95)
Group 10-10	P33.7 - P33.0 / P34.4 - P34.1	224 - 235 (96 - 107)
Group 11-11	--- NOT USED ---	--- NOT USED ---

Note

Means if a port is used as a channel(P0.xx), Bit 7 of numeric channel number has to be set!

Test step commands

.

01 - Initialise the ADC

This command may be used to initialise and start the ADC.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	01	01	01	...

Parameters

ADC

5	+0
(1*)	**ADC**

Details

ADC
Selection of the ADC group

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>ADC</td><td>1</td><td>xx</td><td>ADC index to select the ADC group
Table 3. ADC index table for IFX
<table>
<tr><th>ADC</th><th>Index to ADC Group</th></tr>
<tr><td>All ADC groups</td><td>0xFF</td></tr>
<tr><td>ADC Group 0</td><td>0x00</td></tr>
<tr><td>ADC Group 1</td><td>0x01</td></tr>
<tr><td>ADC Group 2</td><td>0x02</td></tr>
<tr><td>ADC Group 3</td><td>0x03</td></tr>
<tr><td>ADC Group 4</td><td>0x04</td></tr>
<tr><td>ADC Group 5</td><td>0x05</td></tr>
<tr><td>ADC Group 6</td><td>0x06</td></tr>
<tr><td>ADC Group 7</td><td>0x07</td></tr>
<tr><td>ADC Group 8</td><td>0x08</td></tr>
<tr><td>ADC Group 9</td><td>0x09</td></tr>
<tr><td>ADC Group 10</td><td>0x0A</td></tr>
</table>
Note
Not all modules are available for all devices. Please refer to controller manual for details of assignment analog input channel and ADC Groups.</td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	01	xx	xx	01

.

02 - Setup ADC

Setup ADC

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	01	02	01	...

Parameters

Parameter set for ADC group selection

5	+0
(1*)	**ADC**

Parameter set for ADC channel selection

6	+0	+1	+2	+3	+5
(1*)	**RES**	**UREF**	**EMUX**	**TSA**	**TPCH**

Details

Parameter set for ADC group selection
Parameter set for ADC setup.

Name	Length	Fmt	Description
ADC	**1**	**xx**	**ADC index to select the ADC group**

Table 4. ADC index table for IFX

ADC	Index to ADC Group
All ADC groups	0xFF
ADC Group 0	0x00
ADC Group 1	0x01
ADC Group 2	0x02
ADC Group 3	0x03
ADC Group 4	0x04
ADC Group 5	0x05
ADC Group 6	0x06
ADC Group 7	0x07
ADC Group 8	0x08
ADC Group 9	0x09
ADC Group 10	0x0A

Note

Not all modules are available for all devices. Please refer to controller manual for details of assignment of analog input channels to ADC Groups.

Parameter set for ADC channel selection
Parameter set for ADC setup.

Name	Length	Fmt	Description
RES	**1**	**xx**	**Resolution Configuration**

Table 5. FSL: Resolution of ADC

Value	Resolution
0x00	12 Bit

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Value</th><th>Resolution</th></tr>
<tr><td>0x01</td><td>10 Bit</td></tr>
<tr><td>0x02</td><td>8 Bit</td></tr>
</table>
For DEV3+ there are also ADCs with a maximum resolution of 10 bit. In this case 0x00 = 10 bit, 0x01 = 8 bit. For details please consult the controller manual.
</td></tr>
<tr><td>UREF</td><td>1</td><td>xx</td><td>Reference Voltage
Table 6. IFX: Reference Input Selection
<table>
<tr><th>Value</th><th>Function</th></tr>
<tr><td>0x00</td><td>Standard reference input Varef</td></tr>
<tr><td>0x01</td><td>Alternate reference input from CH0</td></tr>
</table>
</td></tr>
<tr><td>EMUX</td><td>1</td><td>xx</td><td>Configuration of external multiplexer
External analog multiplexers will be switched to the value provided via MUX parameter.

Not used, the value shall be set to 0x00.</td></tr>
<tr><td>TSA</td><td>2</td><td>xxxx</td><td>Sample capacitor charge time

Sample time. Format: RVVV where R=Resolution, V=Value</td></tr>
<tr><td>TPCH</td><td>2</td><td>xxxx</td><td>Precharge Time

Not used, the value shall be set to 0x0000.</td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	01	xx	xx	01

.

10 - Start conversion and read result

Start conversion and read result

HINT: Call Init and Setup before calling this command!

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	01	10	01	...

Parameters

Parameter set for ADC conversion request

5	+0	+1
(1..n*)	**CHA**	**MUX**

Details

Parameter set for ADC conversion request
Repeatable parameter set for each Analog input(AN) to be measured in parallel. Inputs are arranged on different converter for shorter conversion time if possible. Repetition of the same input will not trigger a second conversion.

Name	Length	Fmt	Description
CHA	**1**	**xx**	**Index of the channel** Virtual analog input number, which is converted internally to (ANx). The measurement of the same analog input(AN) with different multiplexer input selections is not supported at the moment!
MUX	**1**	**xx**	**Channel of the external multiplexer (IFX/JDP) / Pull Up/Down Resistor selection (FSL)** **Bit(s)** – **Description** **7:4** – **EMUX** **0_H:** EMUX off **1_H:** EMUX on **3:0** – **Selection of Multiplexer input (AN0-AN7)**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	01	xx	xx	01	...

Parameters

Result of channel conversion

6	+0
(1..n*)	**RSLT**

Details

Result of channel conversion

Name	Length	Fmt	Description
RSLT	**2**	**xxxx**	**Result of conversion** For every conversion a result is responded

.

11 - Start conversion and read result without initialization

Start conversion and read result without initialization on ADC channel defined with command "Start conversion and read result"

HINT: Call command "Start conversion and read result" before calling this command!

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	01	11	01	...

Parameters

Parameter set for ADC conversion request. For details refer to command: Start conversion and read result

5	+0	+1
(1..n*)	**CHA**	**MUX**

Details

Parameter set for ADC conversion request. For details refer to command: Start conversion and read result
Note: Number of values and values itself must be the same as with command: "Start conversion and read result"

Name	Length	Fmt	Description
CHA	**1**	**xx**	**Index of the channel**
MUX	**1**	**xx**	**Channel of the external multiplexer (IFX/JDP) / Pull Up/Down Resistor selection (FSL)**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	01	xx	xx	01	...

Parameters

Result of channel conversion. For details refer to command: Start conversion and read result

6	+0
(1..n*)	**RSLT**

Details

Result of channel conversion. For details refer to command: Start conversion and read result

Name	Length	Fmt	Description
RSLT	**2**	**xxxx**	**Result of conversion**

.

20 - Average ADC conversions

Do a defined number of conversions and build an average.

HINT: Call Init and Setup before calling this command!

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	01	20	01	...

Parameters

Parameter set for ADC average conversion request

5	+0	+1
(1*)	**NSA**	**STI**

Parameter set for ADC average conversion request

8	+0	+1
(1..n*)	**CHA**	**MUX**

Details

Parameter set for ADC average conversion request
Parameter set.

Name	Length	Fmt	Description
NSA	**1**	**xx**	**Number of Samples**
STI	**2**	**xxxx**	**Time between samples** Time in µs. Format: RVVV where R=Resolution, V=Value [s]

Parameter set for ADC average conversion request
Repeatable parameter.

Name	Length	Fmt	Description
CHA	**1**	**xx**	**Index of the channel** For details regarding channel parameter values please refer to CMD10 parameter "CHA".
MUX	**1**	**xx**	**Channel of the external multiplexer (IFX/JDP) / Pull Up/Down Resistor selection (FSL)** For details regarding multiplexer parameter values please refer to CMD10 parameter "MUX".

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	01	xx	xx	01	...

Parameters

Result of channel conversion

6	+0
(1..n*)	**RSLT**

Details

Result of channel conversion

Name	Length	Fmt	Description
RSLT	**2**	**xxxx**	**Result of conversion** For every conversion a result is responded

.

21 - Average ADC conversions without initialization

Do a defined number of conversions and build an average without initialization.

HINT: Call "Average ADC conversions" before calling this command!

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	01	21	01	...

Parameters

Parameter set for ADC average conversion request. For details refer to command: Average ADC conversions

5	+0	+1
(1*)	**NSA**	**STI**

Parameter set for ADC average conversion request. For details refer to command: Average ADC conversions

8	+0	+1
(1..n*)	**CHA**	**MUX**

Details

Parameter set for ADC average conversion request. For details refer to command: Average ADC conversions
Parameter set.

Note: Values must be the same as with command: "Average ADC conversions"

Name	Length	Fmt	Description
NSA	**1**	**xx**	**Number of Samples**

Name	Length	Fmt	Description
STI	**2**	**xxxx**	**Time between samples**

Parameter set for ADC average conversion request. For details refer to command: Average ADC conversions
Repeatable parameter.

Note: Number of values and values must be the same as with command: "Average ADC conversions"

Name	Length	Fmt	Description
CHA	**1**	**xx**	**Index of the channel**
MUX	**1**	**xx**	**Channel of the external multiplexer (IFX/JDP) / Pull Up/Down Resistor selection (FSL)**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	01	xx	xx	01	...

Parameters

Result of channel conversion. For details refer to command: Average ADC conversions

6	+0
(1..n*)	**RSLT**

Details

Result of channel conversion. For details refer to command: Average ADC conversions

Name	Length	Fmt	Description
RSLT	**2**	**xxxx**	**Result of conversion**

.

40 - Min/Max evaluation of ADC conversion

Set function of ADC min/max evaluation.

HINT: Call Init and Setup before calling this command!

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	01	40	01	...

Parameters

Parameter set for ADC min/max evaluation request

5	+0	+1
(1*)	**NSA**	**STI**

Parameter set for ADC min/max evaluation request

8	+0	+1
(1..n*)	**CHA**	**MUX**

Details

Parameter set for ADC min/max evaluation request
Parameter set.

Name	Length	Fmt	Description
NSA	**1**	**xx**	**Number of samples**
STI	**2**	**xxxx**	**Time between samples** Time in µs. Format: RVVV where R=Resolution, V=Value [s]

Parameter set for ADC min/max evaluation request
Repeatable parameter.

Name	Length	Fmt	Description
CHA	**1**	**xx**	**Index of the channel** For details regarding channel parameter values please refer to CMD10 parameter "CHA".
MUX	**1**	**xx**	**Channel of the external multiplexer (IFX/JDP) / Pull Up/Down Resistor selection (FSL)** For details regarding multiplexer parameter values please refer to CMD10 parameter "MUX".

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	01	xx	xx	01	...

Parameters

Result of channel conversion

6	+0	+2
(1..n*)	**MIN**	**MAX**

Details

Result of channel conversion

Name	Length	Fmt	Description
MIN	**2**	**xxxx**	**Result of conversion** For every conversion a result is responded
MAX	**2**	**xxxx**	**Result of conversion** For every conversion a result is responded

.

41 - Min/Max evaluation of ADC conversion without initialization

Set function of ADC min/max evaluation without initialization.

HINT: Call "Min/Max evaluation of ADC conversion" before calling this command!

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	01	41	01	...

Parameters

Parameter set for ADC min/max evaluation request. For details refer to command: Min/Max evaluation of ADC conversion

5	+0	+1
(1*)	**NSA**	**STI**

Parameter set for ADC min/max evaluation request. For details refer to command: Min/Max evaluation of ADC conversion

8	+0	+1
(1..n*)	**CHA**	**MUX**

Details

Parameter set for ADC min/max evaluation request. For details refer to command: Min/Max evaluation of ADC conversion
Parameter set.

Note: Values must be the same as with command: "Min/Max evaluation of ADC conversion"

Name	Length	Fmt	Description
NSA	**1**	**xx**	**Number of samples**
STI	**2**	**xxxx**	**Time between samples**

Parameter set for ADC min/max evaluation request. For details refer to command: Min/Max evaluation of ADC conversion
Repeatable parameter.

Note: Number of values and values must be the same as with command: "Min/Max evaluation of ADC conversion"

Name	Length	Fmt	Description
CHA	**1**	**xx**	**Index of the channel**
MUX	**1**	**xx**	**Channel of the external multiplexer (IFX/JDP) / Pull Up/Down Resistor selection (FSL)**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	01	xx	xx	01	...

Parameters

Result of channel conversion. For details refer to command: Min/Max evaluation of ADC conversion

6	+0	+2
(1..n*)	**MIN**	**MAX**

Details

Result of channel conversion. For details refer to command: Min/Max evaluation of ADC conversion

Name	Length	Fmt	Description
MIN	**2**	**xxxx**	**Result of conversion**
MAX	**2**	**xxxx**	**Result of conversion**

.

50 - Reference measurement

This command supports a reference voltage correction with measured ADC values.

HINT: Call Init and Setup before calling this command!

The measured ADC value Uadc_measured can be corrected by calculating the measured reference voltage Uref_measured and the calibrated reference value Uref_calibrated into the output value Uadc_corrected. The correction is done using a calibrateable adjustment value COR from 0% to 100% where 0% means no voltage correction and 100% means full voltage correction. This is necessary because in case of some resistors at the input network of the analog input pin it might be necessary to limit the voltge correction to a certain percentage. The calculation is as follows:

Uadc_corrected = Uadc_measured*(1 + COR*((Uref_calibrated / Uref_measured) -1))

This command supposes command 0x1 and command 0x2.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	01	50	01	...

Parameters

Parameter set for the reference measurement

5	+0	+1
(1*)	**NSA**	**STI**

Parameter set for the reference measurement

8	+0	+1	+2	+3
...	**REFCHA**	**REFMUX**	**COR**	**UREF**

Parameter set for the reference measurement

13	+0	+1
(1..n*)	**CHA**	**MUX**

Details

Parameter set for the reference measurement
Parameter set.

Name	Length	Fmt	Description
NSA	**1**	**xx**	**Number of Samples**
STI	**2**	**xxxx**	**Time between samples** Time in µs. Format: RVVV where R=Resolution, V=Value [s]

Parameter set for the reference measurement
Parameter set.

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>REFCHA</td><td>1</td><td>xx</td><td>Index of the reference channel
For details regarding channel parameter values please refer to CMD10 parameter "CHA".</td></tr>
<tr><td>REFMUX</td><td>1</td><td>xx</td><td>Channel of the external multiplexer of the Reference Channel
For details regarding channel parameter values please refer to CMD10 parameter "MUX".</td></tr>
<tr><td>COR</td><td>1</td><td>xx</td><td>Correction of ref. voltage
Correction of the reference voltage in percent. Resolution is 0.5%.

0 -> 0%

200 -> 100%</td></tr>
<tr><td>UREF</td><td>2</td><td>xxxx</td><td>Reference voltage
Reference voltage.

Table 7. Examples for reference voltage values
<table>
<tr><th>voltage</th><th>8Bit@5.0V</th><th>10Bit@5.0V</th><th>12Bit@5.0V</th></tr>
<tr><td>1.0V</td><td>0x033</td><td>0x0CC</td><td>0x333</td></tr>
<tr><td>2.5V</td><td>0x07F</td><td>0x1FF</td><td>0x7FF</td></tr>
<tr><td>3.3V</td><td>0x0A8</td><td>0x2A3</td><td>0xA8E</td></tr>
<tr><td>5.0V</td><td>0x0FF</td><td>0x3FF</td><td>0xFFF</td></tr>
</table></td></tr>
</table>

Parameter set for the reference measurement

Repeatable parameter.

Name	Length	Fmt	Description
CHA	**1**	**xx**	**Index of the channel** For details regarding channel parameter values please refer to CMD10 parameter "CHA".
MUX	**1**	**xx**	**Channel of the external multiplexer (IFX/JDP) / Pull Up/Down Resistor selection (FSL)** For details regarding multiplexer parameter values please refer to CMD10 parameter "MUX".

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	01	xx	xx	01	...

Parameters

Result of channel conversion

6	+0	+2	+4
(1*)	**RSLT**	**REFRSLT**	**UADCCOR**

Details

Result of channel conversion

Name	Length	Fmt	Description
RSLT	**2**	**xxxx**	**Result of conversion** For every conversion a result is responded
REFRSLT	**2**	**xxxx**	**Result of reference conversion** For every conversion a result is responded
UADCCOR	**2**	**xxxx**	**Value of the corrected measured voltage** Uadc_corrected = Uadc_measured(1+COR((Uref_calibrated / Uref_measured) -1))

.

51 - Reference measurement without initialization

This command supports a reference voltage correction with measured ADC values without initialization.

HINT: Call "Reference measurement" before calling this command!

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	01	51	01	...

Parameters

Parameter set for the reference measurement. For details refer to command: Reference measurement

5 (1*)	+0	+1
	NSA	STI

Parameter set for the reference measurement. For details refer to command: Reference measurement

8 ...	+0	+1	+2	+3
	REFCHA	REFMUX	COR	UREF

Parameter set for the reference measurement. For details refer to command: Reference measurement

13 (1..n*)	+0	+1
	CHA	MUX

Details

Parameter set for the reference measurement. For details refer to command: Reference measurement
Parameter set.

Note: Values must be the same as with command: "Reference measurement"

Name	Length	Fmt	Description
NSA	**1**	**xx**	**Number of Samples**
STI	**2**	**xxxx**	**Time between samples**

Parameter set for the reference measurement. For details refer to command: Reference measurement
Parameter set.

Note: Values must be the same as with command: "Reference measurement"

Name	Length	Fmt	Description
REFCHA	**1**	**xx**	**Index of the reference channel**
REFMUX	**1**	**xx**	**Channel of the external multiplexer of the Reference Channel**
COR	**1**	**xx**	**Correction of ref. voltage**
UREF	**2**	**xxxx**	**Reference voltage**

Parameter set for the reference measurement. For details refer to command: Reference measurement
Repeatable parameter.

Note: Number of values and values must be the same as with command: "Reference measurement"

Name	Length	Fmt	Description
CHA	**1**	**xx**	**Index of the channel**

Name	Length	Fmt	Description
MUX	**1**	**xx**	**Channel of the external multiplexer**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	01	xx	xx	01	...

Parameters

Result of channel conversion. For details refer to command: Reference measurement

6	+0	+2	+4
(1*)	**RSLT**	**REFRSLT**	**UADCCOR**

Details

Result of channel conversion. For details refer to command: Reference measurement

Name	Length	Fmt	Description
RSLT	**2**	**xxxx**	**Result of conversion**
REFRSLT	**2**	**xxxx**	**Result of reference conversion**
UADCCOR	**2**	**xxxx**	**Value of the corrected measured voltage**

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)

Code	Description
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0xFF	**HAL busy**
0xFE	**Generic HAL error**
0xFD	**HAL not compatible to current CPU**
0xFC	**Invalid params**
0xFB	**HAL module not implemented**
0xFA	**HAL function not implemented**
0xF9	**HAL timeout**
0xEF	**ADC channel not supported.**
0xEE	**Conversion result not valid.**
0xED	**ADC group not supported.**

TSW_02 - SPI communication

Test step description

Date 08.08.2013

This TSW module provides low level SPI communication

High speed SPI initialisation:

To initialize HSSPI after execution of TSW04 CMD07 the negative pins of differential outputs should be set with TSW04 CMD04 to SPI alternate function. This should be followed by TSW04 CMD06 where the LVDS mode is set for differential inputs and outputs.

Test step commands

.

00 - Initialise/Reset specific SPI controller

This command may be used to reset and initialise global SPI controller parameters. This command is optional and only needed to reset/re-initialise a specific SPI controller. All controllers are enabled after download of HAL.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	02	00	01	...

Parameters

Parameter set for selecting a SPI controller by port/pin numbers

5	+0	+1	+2	+3	+4	+5
(1*)	**PORTMT**	**PINMT**	**PORTMR**	**PINMR**	**PORTCK**	**PINCK**

...	+6	+7
(1*)	**PORTSL**	**PINSL**

Details

Parameter set for selecting a SPI controller by port/pin numbers

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal

Name	Length	Fmt	Description
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal. If MSB bit is set, emulated SLSO is used with given port (only for IFX and JDP).
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal. If MSB bit is set, emulated SLSO is used with given pin (only for IFX and JDP)

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	02	xx	xx	01

.

01 - Setup channel parameters

Command to configure the clock phase, clock polarity, shift direction and the active level of the slave select signal.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	02	01	01	...

Parameters

Parameter set for SLSO operation

5 (1..n*)	+0	+1	+2	+3	+4	+5
	PORTMT	**PINMT**	**PORTMR**	**PINMR**	**PORTCK**	**PINCK**

... (1..n*)	+6	+7	+8
	PORTSL	**PINSL**	**CFG**

Details

Parameter set for SLSO operation

Parameter set for defining channel parameters. Multiple parameter sets may be added to command line to configure SLSOs sequentially.

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>PORTMT</td><td>1</td><td>xx</td><td>Port number of MTSR signal
Port number of MTSR (Master Transmit) signal</td></tr>
<tr><td>PINMT</td><td>1</td><td>xx</td><td>Pin number of MTSR signal
Pin number within PORTMT of MTSR (Master Transmit) signal</td></tr>
<tr><td>PORTMR</td><td>1</td><td>xx</td><td>Port number of MRST signal
Port number of MRST (Master Receive) signal</td></tr>
<tr><td>PINMR</td><td>1</td><td>xx</td><td>Pin number of MRST signal
Pin number withing PORTMR of MRST (Master Receive) signal</td></tr>
<tr><td>PORTCK</td><td>1</td><td>xx</td><td>Port number of SCK signal
Port number of SCK (Serial Clock) signal</td></tr>
<tr><td>PINCK</td><td>1</td><td>xx</td><td>Pin number of SCK signal
Pin number within PORTCK of SCK (Serial Clock) signal</td></tr>
<tr><td>PORTSL</td><td>1</td><td>xx</td><td>Port number of SLSO signal
Port number of SLSO (Slave Select) signal. If MSB bit is set, emulated SLSO is used with given port (only for IFX and JDP).</td></tr>
<tr><td>PINSL</td><td>1</td><td>xx</td><td>Pin number of SLSO signal
Pin number within PORTSL of SLSO (Slave Select) signal. If MSB bit is set, emulated SLSO is used with given pin (only for IFX and JDP).</td></tr>
<tr><td>CFG</td><td>1</td><td>xx</td><td>Configuration flags
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:5</td><td>not used</td></tr>
<tr><td>4</td><td>Byte shift direction
0_B: Low byte first
1_B: High byte first</td></tr>
<tr><td>3</td><td>Slave select active level
0_B: SLSO is active low
1_B: SLSO is active high</td></tr>
<tr><td>2</td><td>Clock phase
0_B: Shift on leading edge
1_B: Shift on trailing edge</td></tr>
<tr><td>1</td><td>Clock polarity
0_B: Idle state is low
1_B: Idle state is high</td></tr>
<tr><td>0</td><td>Shift direction
0_B: LSB first</td></tr>
</table>
</td></tr>
</table>

Name	Length	Fmt	Description	
			Bit(s)	**Description**
				1_B: MSB first

Response

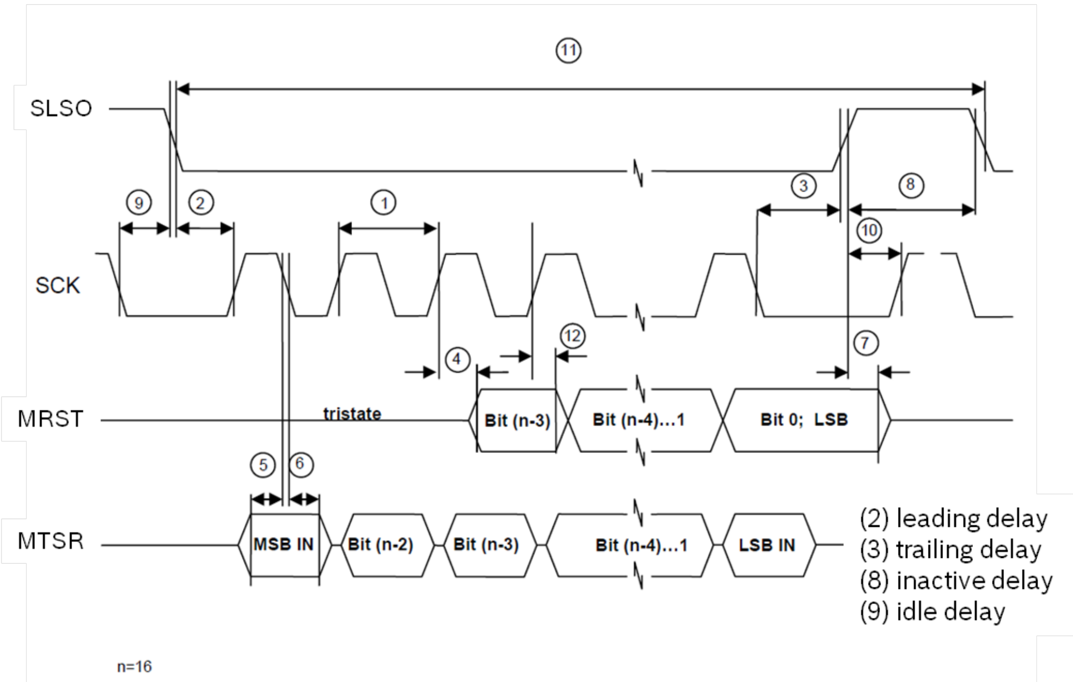
1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	02	xx	xx	01

.

02 - Setup channel timing parameters

Configuration of the slave select timing parameters like leading- and trailing delay, baud rate and the necessary delay between switching SCK signal to idle and activating the SLSO signal.

Figure 1. Slave select timing parameter



Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	02	02	01	...

Parameters

Parameter set for SLSO timing configuration

5	+0	+1	+2	+3	+4	+5
(1..n*)	**PORTMT**	**PINMT**	**PORTMR**	**PINMR**	**PORTCK**	**PINCK**

...	+6	+7
(1..n*)	**PORTSL**	**PINSL**

Channel timings

13	+0	+2	+4	+6	+8
...	**BR**	**LEAD**	**TRAIL**	**INACT**	**IDLE**

Details

Parameter set for SLSO timing configuration

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal0xFF: PINMT is interpreted as SPI handle ID for faster access
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signalIf Port was 0xFF, this is used as SPI handle ID. These IDs are in the same order the SPI channel parameters where set up using CMD_01, starting with 0. maximum: STIL 4 entries, ValSW 32 entries.
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal. If MSB bit is set, emulated SLSO is used with given port (only for IFX and JDP).
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal. If MSB bit is set, emulated SLSO is used with given pin (only for IFX and JDP).

Channel timings

Name	Length	Fmt	Description
BR	**2**	**xxxx**	**Baudrate** Baud rate for communication. Format: RVVV where R=Resolution, V=Value
LEAD	**2**	**xxxx**	**Leading delay** Time delay between activating the SLSO signal and first edge of SCK signal.

Name	Length	Fmt	Description
			Format: RVVV where R=Resolution, V=Value
TRAIL	**2**	**xxxx**	**Trailing delay** Time delay after last edge of SCK signal and deactivating the SLSO signal. Format: RVVV where R=Resolution, V=Value
INACT	**2**	**xxxx**	**Inactive delay** Time delay between consecutive frames. Format: RVVV where R=Resolution, V=Value
IDLE	**2**	**xxxx**	**Idle delay** Time delay for setting the SCK signal idle state and activating the SLSO signal. Format: RVVV where R=Resolution, V=Value

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	02	xx	xx	01	...

Parameters

Response of command SetChannelTimingParameters

6	+0
(1..n*)	**ACTBR**

Details

Response of command SetChannelTimingParameters

Name	Length	Fmt	Description
ACTBR	**4**	**xxxxxxxx**	**Real baudrate**

.

03 - Transmit SPI data

Command to transmit SPI data to one or more SPI slaves. Received data are responded for each transmitted frame.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	02	03	01	...

Parameters

Parameter set for transmitting SPI data

5 (1..n*)	+0 PORTMT	+1 PINMT	+2 PORTMR	+3 PINMR	+4 PORTCK	+5 PINCK

... (1..n*)	+6 PORTSL	+7 PINSL

Transmit configuration data

13 ...	+0 NUM	+1 CFG	+2 DATA

Details

Parameter set for transmitting SPI data

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal.0xFF: PINMT is interpreted as SPI handle ID for faster access
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signalIf Port was 0xFF, this is used as SPI handle ID. These IDs are in the same order the SPI channel parameters where set up using CMD_01, starting with 0. maximum: STIL 4 entries, ValSW 32 entries.
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number within PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal. If MSB bit is set, emulated SLSO is used with given port (only for IFX and JDP).
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal. If MSB bit is set, emulated SLSO is used with given pin (only for IFX and JDP).

Transmit configuration data

Name	Length	Fmt	Description
NUM	**1**	**xx**	**Number of Frames to transmit**
CFG	**1**	**xx**	**Frame configuration**

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td></td><td>Bit(s)</td><td>Description</td></tr>
<tr><td></td><td></td><td></td><td>7</td><td>Burst mode enable
0_B: SLSO signal returns to inactive between frames
1_B: SLSO signal stays active</td></tr>
<tr><td></td><td></td><td></td><td>6</td><td>not used, always set to 0</td></tr>
<tr><td></td><td></td><td></td><td>5:0</td><td>Number of bits per frame
Selection of number of bits to shift by the QSPI. Supported values are 2...32 (0x02...0x20) bits.</td></tr>
<tr><td>DATA</td><td>1</td><td>xx..</td><td colspan="2">Transmit Data
Data to be transmitted. Data which consists of more than one byte must be provided in high-to-low byteorder.</td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	02	xx	xx	01	...

Parameters

Response of TransmitData request.

6	+0
(1..n*)	**DATA**

Details

Response of TransmitData request.

Name	Length	Fmt	Description
DATA	**1**	**xx..**	**Received SPI data**

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**

Code	Description
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0xFF	**HAL busy**
0xFE	**Generic HAL error**
0xFD	**HAL not compatible to current CPU**
0xFC	**Invalid params**
0xFB	**HAL module not implemented**
0xFA	**HAL function not implemented**
0xF9	**HAL timeout**
0xEF	**No SPI controller on given ports found**
0xEE	**SPI function or option not supported**
0xED	**SPI baudrate not adjustable**
0xEC	**SPI HAL parameter error**

TSW_03 - MSC communication

Test step description

This TSW modul allows bus communication to external peripherals via MSC.

Created 18.01.2017 09:12.

Test step commands

.

01 - Initialise MSC port of specific MSC controller

This command may be used to initialise global MSC controller parameters. Later you will configure with other commands according to the MSC chip. Remember that some chips (E.g. CJxxx) have specific command sequences to unlock the power stages.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	03	01	01	...

Parameters

Parameter set for selecting a MSC controller by port/pin numbers (PO=PORT, PI=PIN)

5	+0	+1	+2	+3	+4	+5
(1..n*)	**POFCLP**	**PIFCLP**	**POFCLN**	**PIFCLN**	**POEN**	**PIEN**

...	+6	+7	+8	+9	+10	+11
(1..n*)	**POSOP**	**PISOP**	**POSON**	**PISON**	**POSDI**	**PISDI**

Parameter set for configuration of MSC controller

17	+0	+1	+3	+4	+5	+6
(1..n*)	**CLK**	**BAUD**	**PPL**	**NPF**	**MODE**	**POL**

Details

Parameter set for selecting a MSC controller by port/pin numbers (PO=PORT, PI=PIN)

Name	Length	Fmt	Description
POFCLP	**1**	**xx**	**Port number of FCLP signal** Port number of FCLP (Clock positiv) signal
PIFCLP	**1**	**xx**	**Pin number of FCLP signal** Pin number within POFCLP of FCLP (Clock positiv) signal
POFCLN	**1**	**xx**	**Port number of FCLN signal** Port number of FCLN (Clock negativ) signal. Set 0xFF if single ended is used.

Name	Length	Fmt	Description
PIFCLN	**1**	**xx**	**Pin number of FCLN signal** Pin number within POFCLN of FCLN (Clock negativ) signal. Set 0xFF if single ended is used.
POEN	**1**	**xx**	**Port number of EN signal** Port number of EN (Enable) signal
PIEN	**1**	**xx**	**Pin number of EN signal** Pin number within POEN of EN (Enable) signal
POSOP	**1**	**xx**	**Port number of SOP signal** Port number of SOP (Serial Output Positiv) signal.
PISOP	**1**	**xx**	**Pin number of SOP signal** Pin number within POSOP of SOP (Serial Output Positiv) signal.
POSON	**1**	**xx**	**Port number of SON signal** Port number of SON (Serial Output Negativ) signal. Set 0xFF is single ended is used.
PISON	**1**	**xx**	**Pin number of SON signal** Pin number within POSON of SON (Serial Output Negativ) signal. Set 0xFF is single ended is used.
POSDI	**1**	**xx**	**Port number of SDI signal** Port number of SDI (Serial Data In) signal
PISDI	**1**	**xx**	**Pin number of SDI signal** Pin number within POSDI of SDI (Serial Data In) signal

Parameter set for configuration of MSC controller

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>CLK</td><td>1</td><td>xx</td><td>MSC clock mode
Mode of MSC and FSC clock.
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:4</td><td>MSC clock
0_H: NORMAL clock divider
1_H: FRACTIONAL clock divider</td></tr>
<tr><td>3:0</td><td>FSC clock
0_H: FSC active on downstream frames only
1_H: FSC permanently active</td></tr>
</table></td></tr>
<tr><td>BAUD</td><td>2</td><td>xxxx</td><td>MSC baud rate [Hz]
MSC baud rate [Hz].

Format: RVVV where R=Resolution, V=Value

Note
Please note that when NORMAL clock mode is selected, the effective baud rate may differ from the selected since the effecive baud rate will adjust to a integer divider of the system frequency.</td></tr>
</table>

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>Tip
The frequency is entered as a integer value in RVVV format. If the frequency of BAUD is a float value (including decimal point) please enter the value in integer format and round down the last digit to get the correct frequency. e.g. 16.67MHz => 1666 x 10kHz (RVVV: 0x4682)</td></tr>
<tr><td>PPL</td><td>1</td><td>xx</td><td>Passive phase length of dataframes [SysTimer ticks]
Passive phase length of dataframes [SysTimer ticks].

Range 2..31</td></tr>
<tr><td>NPF</td><td>1</td><td>xx</td><td>Number of passive frames between two data frames
Number of passive frames between two data frames [n]</td></tr>
<tr><td>MODE</td><td>1</td><td>xx</td><td>Transfer mode
Selection of MSC transfer mode:

0=triggered, 1=Continuous</td></tr>
<tr><td>POL</td><td>1</td><td>xx</td><td>MSC pin polarities
Polarities of MSC pins.
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:4</td><td>Not used</td></tr>
<tr><td>3</td><td>SDIx

0_B: Low
1_B: High</td></tr>
<tr><td>2</td><td>ENAx

0_B: Low
1_B: High</td></tr>
<tr><td>1</td><td>SDO

0_B: Low
1_B: High</td></tr>
<tr><td>0</td><td>FCL

0_B: Low
1_B: High</td></tr>
</table>
</td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	03	xx	xx	01

.

02 - Configure MSC frames

This command is used to configure MSC data and command frames. ENx and SDIx line for selected frame type will be choosen automatically from EN/SDI Port/Pin parameter.

Important: Configure DATA frame(s) before COMMAND frame to avoid unexpected behavior! Also configure DATA LOW frame first and then DATA HIGH frame.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	03	02	01	...

Parameters

Parameter set for selecting a MSC controller by port/pin numbers (PO=PORT, PI=PIN)

5 (1..n*)	+0	+1	+2	+3	+4	+5
	POFCLP	**PIFCLP**	**POFCLN**	**PIFCLN**	**POEN**	**PIEN**

... (1..n*)	+6	+7	+8	+9	+10	+11
	POSOP	**PISOP**	**POSON**	**PISON**	**POSDI**	**PISDI**

Parameter set for MSC frame configuration

17 (1..n*)	+0	+1	+2	+3	+7	+8
	TYP	**BITS**	**PHA**	**DSRC**	**RXBIT**	**RXBR**

... (1..n*)	+9
	RXPA

Details

Parameter set for selecting a MSC controller by port/pin numbers (PO=PORT, PI=PIN)

Name	Length	Fmt	Description
POFCLP	**1**	**xx**	**Port number of FCLP signal** Port number of FCLP (Clock positiv) signal0xFF: PI FCLP is interpreted as MSC handle ID for faster access
PIFCLP	**1**	**xx**	**Pin number of FCLP signal** Pin number within POFCLP of FCLP (Clock positiv) signalIf Port was 0xFF, this is used as MSC handle ID. These IDs are in the same order the MSC modules where initialized using CMD_01, starting with 0. maximum: STIL 2 entries, ValSW 8 entries
POFCLN	**1**	**xx**	**Port number of FCLN signal** Port number of FCLN (Clock negativ) signal. Set 0xFF if single ended is used.
PIFCLN	**1**	**xx**	**Pin number of FCLN signal** Pin number within POFCLN of FCLN (Clock negativ) signal. Set 0xFF if single ended is used.

Name	Length	Fmt	Description
POEN	**1**	**xx**	**Port number of EN signal** Port number of EN (Enable) signal
PIEN	**1**	**xx**	**Pin number of EN signal** Pin number within POEN of EN (Enable) signal
POSOP	**1**	**xx**	**Port number of SOP signal** Port number of SOP (Serial Output Positiv) signal.
PISOP	**1**	**xx**	**Pin number of SOP signal** Pin number within POSOP of SOP (Serial Output Positiv) signal.
POSON	**1**	**xx**	**Port number of SON signal** Port number of SON (Serial Output Negativ) signal. Set 0xFF is single ended is used.
PISON	**1**	**xx**	**Pin number of SON signal** Pin number within POSON of SON (Serial Output Negativ) signal. Set 0xFF is single ended is used.
POSDI	**1**	**xx**	**Port number of SDI signal** Port number of SDI (Serial Data In) signal
PISDI	**1**	**xx**	**Pin number of SDI signal** Pin number within POSDI of SDI (Serial Data In) signal

Parameter set for MSC frame configuration

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>TYP</td><td>1</td><td>xx</td><td>Frame type to be configured
Following selections are supported:
<table>
<tr><th>Selection</th><th>Frame type</th></tr>
<tr><td>0</td><td>DATA LOW frame</td></tr>
<tr><td>1</td><td>DATA HIGH frame</td></tr>
<tr><td>2</td><td>COMMAND frame</td></tr>
</table>
Configure DATA frame(s) before COMMAND frame to avoid unexpected behavior! Also configure DATA LOW frame first and then DATA HIGH frame.

For JDP low and high frame are mapped to 16 bit registers. For smaller data frames (E.g. CJ960 with 13 bits) if it's on the low frame the first bits of the high frame (13-15) must be configured as low frame bits.</td></tr>
<tr><td>BITS</td><td>1</td><td>xx</td><td>Number of bits to transfer
Number of bits to be transferred in selected frame type.

Range: 0..32</td></tr>
<tr><td>PHA</td><td>1</td><td>xx</td><td>Enable/Disable active phase selection bit
Enable/Disable active phase selection bit for selected frame type. 0=disable, 1=enable</td></tr>
<tr><td>DSRC</td><td>4</td><td>xxxxxxxx</td><td>Frame data source mask</td></tr>
</table>

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>Selection of data source for each bit of data frame.

Each bit of data frame is defined by 2 bits within data source mask with binary settings shown below:
<table>
<tr><th>MaskBits</th><th>Bit in data frame</th></tr>
<tr><td>0..1</td><td>0</td></tr>
<tr><td>2..3</td><td>1</td></tr>
<tr><td>4..5</td><td>2</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>26..27</td><td>13</td></tr>
<tr><td>28..29</td><td>14</td></tr>
<tr><td>30..31</td><td>15</td></tr>
</table>
Following values may be selected for each data frame bit
<table>
<tr><th>Value (binary)</th><th>Data source</th></tr>
<tr><td>00</td><td>DATA register value</td></tr>
<tr><td>01</td><td>not used</td></tr>
<tr><td>10</td><td>Alternative input (i.e. GTM or ETPU/EMIOS for E84/C55) Alternative input configuration is done in PWM.</td></tr>
<tr><td>11</td><td>Alternative input inverted (i.e. GTM) [currently NOT implemented for JDP]</td></tr>
</table>
</td></tr>
<tr><td>RXBIT</td><td>1</td><td>xx</td><td>number of bits for RX
Number of bits for upstream frame.

Range: 0..16</td></tr>
<tr><td>RXBR</td><td>1</td><td>xx</td><td>Baud rate divider for RX
RX Baud rate divider. RX baud rate = MSC baud rate / RxBaudDiv.

Range: 2..128

Selection of 0 disables upstream frame reception</td></tr>
<tr><td>RXPA</td><td>1</td><td>xx</td><td>Parity for RX
Expected parity of upstream RX frames.

0=EVEN, 1=ODD</td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	03	xx	xx	01

.

03 - Set/Update data for data frame

This command is used to set / update MSC data frames.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	03	03	01	...

Parameters

Parameter set for selecting a MSC controller by port/pin numbers (PO=PORT, PI=PIN)

5	+0	+1	+2	+3	+4	+5
(1..n*)	**POFCLP**	**PIFCLP**	**POFCLN**	**PIFCLN**	**POEN**	**PIEN**

...	+6	+7	+8	+9	+10	+11
(1..n*)	**POSOP**	**PISOP**	**POSON**	**PISON**	**POSDI**	**PISDI**

Parameter set to set TX data for MSC data frame

17	+0	+4
(1*)	**MSK**	**VAL**

Details

Parameter set for selecting a MSC controller by port/pin numbers (PO=PORT, PI=PIN)

Name	Length	Fmt	Description
POFCLP	**1**	**xx**	**Port number of FCLP signal** Port number of FCLP (Clock positiv) signal0xFF: PI FCLP is interpreted as MSC handle ID for faster access
PIFCLP	**1**	**xx**	**Pin number of FCLP signal** Pin number within POFCLP of FCLP (Clock positiv) signalIf Port was 0xFF, this is used as MSC handle ID. These IDs are in the same order the MSC modules where initialized using CMD_01, starting with 0. maximum: STIL 2 entries, ValSW 8 entries
POFCLN	**1**	**xx**	**Port number of FCLN signal** Port number of FCLN (Clock negativ) signal. Set 0xFF if single ended is used.
PIFCLN	**1**	**xx**	**Pin number of FCLN signal** Pin number within POFCLN of FCLN (Clock negativ) signal. Set 0xFF if single ended is used.
POEN	**1**	**xx**	**Port number of EN signal** Port number of EN (Enable) signal
PIEN	**1**	**xx**	**Pin number of EN signal** Pin number within POEN of EN (Enable) signal

Name	Length	Fmt	Description
POSOP	**1**	**xx**	**Port number of SOP signal** Port number of SOP (Serial Output Positiv) signal.
PISOP	**1**	**xx**	**Pin number of SOP signal** Pin number within POSOP of SOP (Serial Output Positiv) signal.
POSON	**1**	**xx**	**Port number of SON signal** Port number of SON (Serial Output Negativ) signal. Set 0xFF is single ended is used.
PISON	**1**	**xx**	**Pin number of SON signal** Pin number within POSON of SON (Serial Output Negativ) signal. Set 0xFF is single ended is used.
POSDI	**1**	**xx**	**Port number of SDI signal** Port number of SDI (Serial Data In) signal
PISDI	**1**	**xx**	**Pin number of SDI signal** Pin number within POSDI of SDI (Serial Data In) signal

Parameter set to set TX data for MSC data frame

This command doesn't explicitly configure a low and high frame. It assumes a maximum frame size of 32 bits. So if you have e.g. a CJ960 (Data frame has 13 bits only) in your low frame you must take into account that the first bit of the high frame chip is bit13.

Name	Length	Fmt	Description
MSK	**4**	**xxxxxxxx**	**Frame data update mask** Bit mask selection of data bits which have to be updated.
VAL	**4**	**xxxxxxxx**	**Frame data update value** Value of data to be sent (masked by above bitmask).

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	03	xx	xx	01

.

04 - Send command frame

This command is used to send command frames.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	03	04	01	...

Parameters

Parameter set for selecting a MSC controller by port/pin numbers (PO=PORT, PI=PIN)

5	+0	+1	+2	+3	+4	+5
(1..n*)	**POFCLP**	**PIFCLP**	**POFCLN**	**PIFCLN**	**POEN**	**PIEN**

...	+6	+7	+8	+9	+10	+11
(1..n*)	**POSOP**	**PISOP**	**POSON**	**PISON**	**POSDI**	**PISDI**

Parameter set to send TX data for MSC data frames

17	+0	+2
(1..n*)	**CMD**	**RXN**

Details

Parameter set for selecting a MSC controller by port/pin numbers (PO=PORT, PI=PIN)

Name	Length	Fmt	Description
POFCLP	**1**	**xx**	**Port number of FCLP signal** Port number of FCLP (Clock positiv) signal0xFF: PI FCLP is interpreted as MSC handle ID for faster access
PIFCLP	**1**	**xx**	**Pin number of FCLP signal** Pin number within POFCLP of FCLP (Clock positiv) signalIf Port was 0xFF, this is used as MSC handle ID. These IDs are in the same order the MSC modules where initialized using CMD_01, starting with 0. maximum: STIL 2 entries, ValSW 8 entries
POFCLN	**1**	**xx**	**Port number of FCLN signal** Port number of FCLN (Clock negativ) signal. Set 0xFF if single ended is used.
PIFCLN	**1**	**xx**	**Pin number of FCLN signal** Pin number within POFCLN of FCLN (Clock negativ) signal. Set 0xFF if single ended is used.
POEN	**1**	**xx**	**Port number of EN signal** Port number of EN (Enable) signal
PIEN	**1**	**xx**	**Pin number of EN signal** Pin number within POEN of EN (Enable) signal
POSOP	**1**	**xx**	**Port number of SOP signal** Port number of SOP (Serial Output Positiv) signal.
PISOP	**1**	**xx**	**Pin number of SOP signal** Pin number within POSOP of SOP (Serial Output Positiv) signal.
POSON	**1**	**xx**	**Port number of SON signal** Port number of SON (Serial Output Negativ) signal. Set 0xFF is single ended is used.
PISON	**1**	**xx**	**Pin number of SON signal** Pin number within POSON of SON (Serial Output Negativ) signal. Set 0xFF is single ended is used.
POSDI	**1**	**xx**	**Port number of SDI signal** Port number of SDI (Serial Data In) signal
PISDI	**1**	**xx**	**Pin number of SDI signal**

Name	Length	Fmt	Description
			Pin number within POSDI of SDI (Serial Data In) signal

Parameter set to send TX data for MSC data frames

Name	Length	Fmt	Description
CMD	**2**	**xxxx**	**Command frame value** Value of command to be sent.
RXN	**1**	**xx**	**number of expected response frames** Range: 0..8

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	03	xx	xx	01	...

Parameters

Command response data

6	+0	+1	+2
(1*)	**RXN**	**STA**	**DATA**

Details

Command response data
Command response data. This response block only appears, if any data is received from MSC port.

Name	Length	Fmt	Description
RXN	**1**	**xx**	**Number of received frames** number of received upstream frames.
STA	**1**	**xx**	**RX Frame status** Bit mask of upstream frame parity errors. BIT0=Parity status Frame0 ... Bit7=Parity status Frame7
DATA	**1**	**xx..**	**RX frame data bytes ...** Received upstream data bytes [0...numRxFrms]

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**

Code	Description
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0xFF	**HAL busy**
0xFE	**Generic HAL error**
0xFD	**HAL not compatible to current CPU**
0xFC	**Invalid params**
0xFB	**HAL module not implemented**
0xFA	**HAL function not implemented**
0xF9	**HAL timeout**
0xEF	**No MSC controller on given ports found**
0xEE	**MSC function not available**
0xED	**MSC illegal parameter**
0xEC	**MSC timeout occured**

Code	Description
0xEB	**MSC feature not implemented**
0xEA	**MSC end of queue missing**

TSW_04 - GPIO port pin operations

Test step description

This test step allows operation of digital I/O ports.

Since I/O ports may be used for other functions as well, the TSW also allows to change functionality of a port/pin.

P40 is usable only for analog ADC input or for SENT input.

Created 18.01.2017 09:12.

Port/Pins and Freescale microcontrollers

Freescale does not support a Port/Pin concept like Infineon, but just number their 'Pads' from 0...511. Please translate this 'Pad' number into Port/Pin with the following formula:

Port = Pad / 16 (round down)

Pin = Pad modulo 16

Test step commands

.

01 - Port/Pin Operations

This command may be used to set static output pins to specified pin level, read and report current value of digital input ports(s), or to wait a specified time. Because of code runtime small wait times can`t be used and also this code runtime must be considered!

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	04	01	01	...

Parameters

Parameter set for port set operation

5	+0	+1	+2	+4
(1..n*)	**CFG**	**PORT**	**PINS**	**STA**

Parameter set for port read operation

11	+0	+1	+2
(1..n*)	**CFG**	**PORT**	**PINS**

Parameter set for port wait operation

15	+0	+1
(1..n*)	**CFG**	**WAIT**

Details

Parameter set for port set operation
Multiple parameter sets may be appended to create a complete sequence of port/pin settings

Name	Length	Fmt	Description
CFG	**1**	**xx**	**Type of port operation** In this case Type "0x80" for WRITE
PORT	**1**	**xx**	**GPIO port number** Port number of digital output port.
PINS	**2**	**xxxx**	**Bit mask selection of port pins** Pin selection bit mask, 1 bit for each pin within selected port. Bit0=Pin0...Bit15=Pin15. Set bit(s) to 1 if pin(s) of selected port shall be operated
STA	**2**	**xxxx**	**Bit mask selection of port pin status** Pin output status bit mask, 1 bit for each pin within selected port. Bit0=Pin0...Bit15=Pin15. Set bit(s) to 1 if pin(s) output status shall be set to HIGH state. Set bit(s) to 0 if pin(s) output status shall be set to LOW state. This setting will only change a pin state.

Parameter set for port read operation
Multiple parameter sets may be appended to create a complete sequence of port/pin settings

Name	Length	Fmt	Description
CFG	**1**	**xx**	**Type of port operation** In this case "0x00" for READ
PORT	**1**	**xx**	**GPIO port number** Port number of digital output port.
PINS	**2**	**xxxx**	**Bit mask selection of port pins** Pin selection bit mask, 1 bit for each pin within selected port. Bit0=Pin0...Bit15=Pin15. Set bit(s) to 1 if pin(s) of selected port shall be operated

Parameter set for port wait operation
Multiple parameter sets may be appended to create a complete sequence of port/pin settings

Name	Length	Fmt	Description
CFG	**1**	**xx**	**Type of port operation** In this case "0xFF" for WAIT
WAIT	**2**	**xxxx**	**Wait time [s] before next port operation** Waiting time BEFORE next pin operation is performed. Format: RVVV where R=Resolution, V=Value [s]

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	04	xx	xx	01	...

Parameters

Response set of SetPort operation

6	+0
(1..n*)	**STA**

Details

Response set of SetPort operation

Name	Length	Fmt	Description
STA	**2**	**xxxx**	**I/O status of port operation matching request number n** I/O state of pins in port which was requested by request parameter set number n

.

03 - Read Pin Function(s)

Read current setting of pin function

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	04	03	01	...

Parameters

Parameter set for pin function request

5	+0	+1
(1..n*)	**PORT**	**PINS**

Details

Parameter set for pin function request
Parameter set for reading pin function.

Name	Length	Fmt	Description
PORT	**1**	**xx**	**GPIO port number** Port number of digital I/O port.
PINS	**2**	**xxxx**	**Bit mask selection of port pins** Pin selection bit mask, 1 bit for each pin within selected port. Bit0=Pin0...Bit15=Pin15. Set bit(s) to 1 if pin(s) of selected port shall be read

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	04	xx	xx	01	...

Parameters

Response set of GetPinFunc operation

6	+0	+1
(1..n*)	**DIR**	**ALTFUNC**

Details

Response set of GetPinFunc operation

Name	Length	Fmt	Description
DIR	**1**	**xx**	**Selection of pull and pin direction** Pull and direction ID of pin specified within pin mask. This value is repeated for each pin which is specified within bit mask for pin selection.
ALTFUNC	**1**	**xx**	**Alternate function ID of pin n within bit mask** Alternate function ID of pin specified within pin mask. This value is repeated for each pin which is specified within bit mask for pin selection.

.

04 - Set Pin Function(s)

Set function of port pin(s). This command sets the function of all specified pins on specified port. This command has the same functions as CMD 07. CMD 07 usable as alternative command instead of CMD 04.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	04	04	01	...

Parameters

Parameter set for pin function request

5	+0	+1	+3	+4
(1..n*)	**PORT**	**PINS**	**DIR**	**ALTFUNC**

Details

Parameter set for pin function request
Parameter set for reading pin function. This parameter set may be repeated for each port to be modified.

Name	Length	Fmt	Description
PORT	**1**	**xx**	**GPIO port number** Port number of digital I/O port.
PINS	**2**	**xxxx**	**Bit mask selection of port pins**

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>Pin selection bit mask, 1 bit for each pin within selected port. Bit0=Pin0...Bit15=Pin15. Set bit(s) to 1 if pin(s) of selected port shall be read.</td></tr>
<tr><td>DIR</td><td>1</td><td>xx</td><td>Selection of pull and pin direction
Pull selection and pin direction to selected pin(s).
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:4</td><td>Pull selection
0_H: Don´t change
1_H: No pull
2_H: Pulldown
3_H: Pullup</td></tr>
<tr><td>3:0</td><td>Direction
0_H: Input
1_H: Output</td></tr>
</table></td></tr>
<tr><td>ALTFUNC</td><td>1</td><td>xx</td><td>Alternate function ID of pin n within bit mask
TSW alternate function ID of pin specified within pin mask. This value is repeated for each pin which is specified within bit mask for pin selection.
Table 1. Alternate function table for IFX
<table>
<tr><th>Value</th><th>Function</th></tr>
<tr><td>0x00</td><td>GPIO</td></tr>
<tr><td>0x01</td><td>Alt func 1</td></tr>
<tr><td>0x02</td><td>Alt func 2</td></tr>
<tr><td>0x03</td><td>Alt func 3</td></tr>
<tr><td>0x04</td><td>Alt func 4</td></tr>
<tr><td>0x05</td><td>Alt func 5</td></tr>
<tr><td>0x06</td><td>Alt func 6</td></tr>
<tr><td>0x07</td><td>Alt func 7</td></tr>
<tr><td>0xFF</td><td>Don't change</td></tr>
</table></td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	04	xx	xx	01	...

Parameters

Response set of SetPinFunc operation

6	+0	+1
(1..n*)	**DIR**	**ALTFUNC**

Details

Response set of SetPinFunc operation

Name	Length	Fmt	Description
DIR	**1**	**xx**	**Selection of pull and pin direction** Pull and direction ID of pin specified within pin mask. This value is repeated for each pin which is specified within bit mask for pin selection.
ALTFUNC	**1**	**xx**	**Alternate function ID of pin n within bit mask** TSW alternate function ID of pin specified within pin mask. This value is repeated for each pin which is specified within bit mask for pin selection.

.

05 - Read Pin Signal Properties

Read properties of pin/pad signal. The signal properties are read from all pins of the given port and pin mask. For details of your microcontroller please refer to manual.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	04	05	01	...

Parameters

Parameter set for pin properties setting

5	+0	+1
(1..n*)	**PORT**	**PINS**

Details

Parameter set for pin properties setting
Parameter set for reading pin properties. This parameter set may be repeated for each port to be modified.

Name	Length	Fmt	Description
PORT	**1**	**xx**	**GPIO port number** Port number of digital I/O port.
PINS	**2**	**xxxx**	**Bit mask selection of port pins** Pin selection bit mask, 1 bit for each pin within selected port. Bit0=Pin0...Bit15=Pin15. Set bit(s) to 1 if pin(s) of selected port shall be set

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	04	xx	xx	01	...

Parameters

Response set of GetPinProperties operation

6	+0	+1	+2	+3
(1..n*)	**STSP**	**INLEV**	**OUTLEV**	**SPCONF**

Details

Response set of GetPinProperties operation
This parameter set will be repeated for each pin selected by bmPins.

Name	Length	Fmt	Description
STSP	**1**	**xx**	**Pin driver strength value** Selection value for driver strength
INLEV	**1**	**xx**	**Signal level for input pin** Selection value for input level. The register setting of Automotive and CMOS level is same. Therefore the input level depends on pad level (5V or 3.3V), too. This command will always return Automotive even if the pad type indicates that CMOS level is selected. Reason is that Pad level can not be distiguished by TSW. Please refer to uC manual.
OUTLEV	**1**	**xx**	**Signal level for output pin** Selection value for output level
SPCONF	**2**	**xxxx**	**Special configuration** Under discussion

.

06 - Set Pin Signal Properties

Set properties of pin/pad signal. All pins which are specified with the selected PinMask (bmPins) will be set to desired values. For details of your microcontroller please refer to manual. Note!The TSW uses either INLEVEL oder OUTLEVEL depending on the direction of the selected port pin. So please configure the port pin direction (CMD 04 or CMD 07) before(!) sending CMD 06.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	04	06	01	...

Parameters

Parameter set for pin properties setting

5	+0	+1	+3	+4	+5	+6
(1..n*)	**PORT**	**PINS**	**STSP**	**INLEV**	**OUTLEV**	**SPCONF**

Details

Parameter set for pin properties setting

Parameter set for setting pin properties. This parameter set may be repeated for each port to be modified.

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>PORT</td><td>1</td><td>xx</td><td>GPIO port number
Port number of digital I/O port.</td></tr>
<tr><td>PINS</td><td>2</td><td>xxxx</td><td>Bit mask selection of port pins
Pin selection bit mask, 1 bit for each pin within selected port. Bit0=Pin0...Bit15=Pin15. Set bit(s) to 1 if pin(s) of selected port shall be set</td></tr>
<tr><td>STSP</td><td>1</td><td>xx</td><td>Pin driver strength and speed value
<h3>Table 2. Possible value for driver strength and speed</h3>
<table>
<tr><th>Value</th><th>Strength</th></tr>
<tr><td>0x00</td><td>Weak</td></tr>
<tr><td>0x01</td><td>Medium</td></tr>
<tr><td>0x02</td><td>Strong (FSL: reserved at some pins)</td></tr>
<tr><td>0x03</td><td>Very Strong</td></tr>
</table></td></tr>
<tr><td>INLEV</td><td>1</td><td>xx</td><td>Signal level for input pin
<h3>Note</h3>
The register setting of Automotive and CMOS level is same. Therefore the input level depends on pad level (5V or 3.3V), too. This command will always return Automotive even if the pad type indicates that CMOS level is selected. Reason is that Pad level can not be distiguished by TSW. Please refer to uC manual.
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:6</td><td>Hysteresis
00_B: Don´t change
01_B: Off
10_B: On</td></tr>
<tr><td>5:0</td><td>Level
0_H: Don´t change
1_H: Automotive
2_H: TTL
3_H: CMOS
4_H: LVDS
5_H: LVDS HSCT</td></tr>
</table></td></tr>
<tr><td>OUTLEV</td><td>1</td><td>xx</td><td>Signal level for output pin
<h3>Note</h3>
LVDS HSCT not used at the moment.</td></tr>
</table>

Name	Length	Fmt	Description
			Bit(s) / **Description** **7:4** **Pull selection** **0_H:** Don´t change **1_H:** Push-Pull **2_H:** Open-Drain **3:0** **Level** **0_H:** Don´t change **1_H:** CMOS **2_H:** LVDS MSC **3_H:** LVDS HSCT
SPCONF	**2**	**xxxx**	**Special configuration** 0x0000Intended for future enhancements.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	04	xx	xx	01	...

Parameters

Response set of SetPinProperties operation

6	+0	+1	+2	+3
(1..n*)	**STSP**	**INLEV**	**OUTLEV**	**SPCONF**

Details

Response set of SetPinProperties operation

Name	Length	Fmt	Description
STSP	**1**	**xx**	**Pin driver strength value** Selection value for driver strength
INLEV	**1**	**xx**	**Signal level for input pin** Selection value for input level
OUTLEV	**1**	**xx**	**Signal level for output pin** Selection value for output level
SPCONF	**2**	**xxxx**	**Special configuration** Intended for future enhancements.

.

07 - Set Port/Pins To Module Function

Set pins for specified peripheral module functions i.e. SPI, MSC ... This command has similar functionality as command CMD 04 but uses a module type instead of raw alternate function configuration. For the command to be able to find the correct function in case of a pin having multiple functionalities for a module (SPI, FLX, ...) it should be given a complete pin set (For

FLX e.g. out, enable and in). In contrast to CMD 04 CMD 07 is capable to set alternate function values in "input" MSCR registers which is necessary for JDP. If multiple chips are connected to a HW module (E.g. SPI or MSC) they use the same data and clock pins but different chip selects. In this case you have to call command 07 for every connected chip with the complete pin set. Only the chip select will differ.

For modules where not all pins are on the same port multiple ports can be specified. E.g. "80040701 03 0E 00C4 0044 0F 0100 0100".

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	04	07	01	...

Parameters

Static parameters.

5	+0
(1*)	**MOD**

Repeatable parameters.

6	+0	+1	+3
(1..n*)	**PORT**	**PINS**	**DIR**

Optional parameters.

11	+0
(1*)	**INST**

Details

Static parameters.

Name	Length	Fmt	Description

MOD	**1**	**xx**	**Module to set function pins to.**
			HAL-ID of HAL-Module to set pins for.
			
			If bit7 (0x80) is set for JDP only SSS is changed, for IFX IOCR only. Otherwise there are default settings (push/pull) set for output pins. For IFX without bit7 pull-ups which are set after reset are turned off.
			
			`Supported modules are:`
			`0x01: SENT`
			`0x02: I2C`
			`0x03: SPI`
			`0x05: MSC`
			`0x06: EXTCLOCK`
			`0x07: PWM`
			`0x08: CAN`
			`0x09: FLX`

Name	Length	Fmt	Description
			`0x0C: ETH`

Repeatable parameters.

Name	Length	Fmt	Description
PORT	**1**	**xx**	**GPIO port number** Port number of digital output port.
PINS	**2**	**xxxx**	**Bit mask selection of port pins** Pin selection bit mask, 1 bit for each pin within selected port. Bit0=Pin0...Bit15=Pin15. Set bit(s) to 1 if pin(s) of selected port shall be operated
DIR	**2**	**xxxx**	**Bit mask selection of port pin direction** Pin direction bit mask, 1 bit for each pin within selected port. Bit0=Pin0...Bit15=Pin15. Set bit(s) to 1 if pin(s) shall be set as module OUTPUT pin(s).

Optional parameters.

Name	Length	Fmt	Description
INST	**1**	**xx**	**Instance to set function pins to.** Instance of HAL-Module to set pins for. Necessary to set if pin set is ambiguous.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	04	xx	xx	01	...

Parameters

Module instance selected by pin set.

6	+0
(1*)	**INST**

Details

Module instance selected by pin set.

Name	Length	Fmt	Description
INST	**1**	**xx**	**Selected module instance** For some pin sets multiple instances are possible. So the selected module instance is returned for further use.

.

08 - Debounced Port/Pin Operations

This command may be used to set a static output pin to specified pin level (STAB), wait specified time and set the output pin again to next specified level (STAA). After this, wait specified time and read and report current value of a given digital input pin and set the output pin again (STA). A specified sequence with this command can be used only for one pin and not for a pin group!

Note

Watchdog service is deactivated for the whole sequence, so the sequence time must be shorter than the max. possible watchdog servicing time. Also the code runtime must be considered only in the wait times of the repeatable blocks. Wait time must be 1-2µs lower than expected.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	04	08	01	...

Parameters

Static parameter set for debounced port operation

5	+0	+1	+3	+5	+7	+9
(1*)	**SPORT**	**SPINS**	**STAB**	**WAIT**	**STAA**	**RPORT**

...	+10
(1*)	**RPINS**

Repeatable parameters for debounced port operation.

17	+0	+2
(1..n*)	**WAIT**	**STA**

Details

Static parameter set for debounced port operation

Name	Length	Fmt	Description
SPORT	**1**	**xx**	**GPIO port number for set action** Port number of digital output port for set action.
SPINS	**2**	**xxxx**	**Bit mask selection for port pins for set operation** Pin number of digital output pin for set action.
STAB	**2**	**xxxx**	**Bit mask selection for port pin state before wait** Requested Pin state for given digital output pin.
WAIT	**2**	**xxxx**	**Wait time [s] before next pin operation** Waiting time BEFORE next pin operation is performed. Format: RVVV where R=Resolution, V=Value [s]
STAA	**2**	**xxxx**	**Bit mask selection for port pin state after wait** Requested Pin state for given digital output pin.
RPORT	**1**	**xx**	**GPIO port number for read operation** Port number of digital input port.
RPINS	**2**	**xxxx**	**Bit mask selection for port pins for read operation**

Name	Length	Fmt	Description
			Pin number of digital input port.

Repeatable parameters for debounced port operation.

Name	Length	Fmt	Description
WAIT	**2**	**xxxx**	**Wait time [s] before next pin operation** Waiting time BEFORE next pin operation is performed. Format: RVVV where R=Resolution, V=Value [s]
STA	**2**	**xxxx**	**Bit mask selection for port pin state after wait** Requested Pin state for given digital output pin.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	04	xx	xx	01	...

Parameters

Response set of SetPort operation

6	+0
(1..n*)	**STA**

Details

Response set of SetPort operation

Name	Length	Fmt	Description
STA	**2**	**xxxx**	**I/O status of port operation matching request number n** I/O state of pins in port which was requested by request parameter set number n

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**

Code	Description
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0xFF	**HAL busy**
0xFE	**Generic HAL error**
0xFD	**HAL not compatible to current CPU**
0xFC	**Invalid params**
0xFB	**HAL module not implemented**
0xFA	**HAL function not implemented**
0xEF	**GPIO port not found**
0xEE	**GPIO pin not found**
0xED	**GPIO function not supported**
0xEC	**GPIO port.pin / module not connectable**

TSW_05 - PWM operations

Test step description

Created 18.01.2017 09:12.

This test step provides functionality to generate a PWM signal and output it to an output port/pin or insert it into a MSC frame. Additionally, this test step can analyze a PWM signal that is present at an input pin and return its period and duty cycle.

Command 0x01 ("Setup PWM OUT") is necessary to configure the PWM source (e.g. TPU or GTM module), signal properties (period, duty cycle) and the pins (or bit positions of an MSC frame), where the signal shall be output.

Note 1 Any combination of ATOM and MSC is NOT supported!

Note 2 Due to limitations in timer to port mapping, some ports may have active signals if a timer associated with this port is running. Make sure that IOCR registers of unwanted ports are NOT connected to GTM with teststep GPIO.

You can use command 0x02 ("Link pins") to connect an already running signal to an additional output pin or MSC bit. This command also allows to disconnect a pin or MSC bit from a running signal.

Important note: A PWM timer cannot be deleted after it is configured with command 0x01. In other words, a timer must always be connected to at least 1 output pin. But, you can use command 0x02 to add a second pin to the timer and unlink the first pin (to use it for other purposes). For exceptions please refer to command 0x30 "Disable Ports".

If you unlink the timer that was first connected to a MSC bit using command 0x01, all other MSC bits loose their connection as well.

Command 0x03 ("Modify PWM OUT") can be used to modify the timer properties (period or duty cycle) of an already running signal.

Command 0x04 ("Modify PWM OUT One Shot Mode") can be used to modify the timer properties to one shot mode of an already running signal (started with command 0x1).

Command 0x05 ("Calculate timer registers for PWM OUT One Shot Mode") can be used to calculate the timer registers of a timer configured in one shot mode (started with command 0x1, modified with command 0x4).

Command 0x06 ("Trigger PWM OUT One Shot Mode") can be used to trigger an output of a timer configured in one shot mode of an already running signal (started with command 0x1, modified with command 0x4 and 0x5).

Command 0x10 ("PWM In") analyzes a signal that is present at an input pin and returns its properties (period, duty cycle etc.).

Command 0x11 ("PWM IN - with trigger signal output") is same as Command 0x10, but offers an additional trigger signal (only for IFX/JDP).

Command 0x20 ("PWM OUT - User defined PWM signal from FIFO") may be used to initialise GTM FIFO with defined PWM signal data.

Command 0x30 ("Disable Ports") can be used to completely disable port/pins. Also registered ressources will be cleared. Timers will NOT be disabled.

Port numbers greater or equal than 0x80 are interpreted as MSC ports according to the following table:

Table 1. MSC ports

"PORT"	IFX / JDP	E84 FSL
0x80	MSC0_L (low 16 bit)	DSPI_A (low 16 bit)
0x81	MSC0_H (high 16 bit)	DSPI_A (high 16 bit)
0x82	MSC1_L (low 16 bit)	DSPI_B (low 16 bit)
0x83	MSC1_H (high 16 bit)	DSPI_B (high 16 bit)
0x84	MSC2_L (low 16 bit)	DSPI_C (low 16 bit)
0x85	MSC2_H (high 16 bit)	DSPI_C (high 16 bit)
0x86	MSC3_L (low 16 bit)	DSPI_D (low 16 bit)
0x87	MSC3_H (high 16 bit)	DSPI_D (high 16 bit)
0xC0	MSC0_LE (low extended 16 bit)	not used
0xC1	not used	not used
0xC2	MSC1_LE (low extended 16 bit)	not used
0xC3	not used	not used
0xC4	MSC2_LE (low extended 16 bit)	not used
0xC5	not used	not used
0xC6	MSC3_LE (low extended 16 bit)	not used
0xC7	not used	not used

All physical parameters like period, timeouts etc. use the RVVV format, where R is the resolution (see following table) and VVV is the value of the parameter. For example 2ms would be 0xD002.

Table 2. Resolution table

Value	Resolution
0x7	1ns
0x8	10ns
0x9	100ns
0xA	1µs
0xB	10µs
0xC	100µs
0xD	1ms
0xE	10ms
0xF	100ms

The following Timer Resources (GTM, TPU, EMIOS etc.) are available:

Table 3. Timer Resources

Value	IFX/JDP	FSL
0x00	TIM0, TOM0	TPU_A
0x01	TIM1, TOM1	TPU_B
0x02	TIM2, TOM2	TPU_C
0x03	TIM3, TOM3	MIOS_0

Value	IFX/JDP	FSL
0x04	TIM4, TOM4	MIOS_1
0x0x	TIMx, TOMx	not supported
0x10	ATOM0	not supported
0x11	ATOM1	not supported
0x12	ATOM2	not supported
0x13	ATOM3	not supported
0x14	ATOM4	not supported
0x1x	ATOMx	not supported

Note: ATOMs are NOT supported in combination with MSC!

The following Timer resources can be mapped into MSC frames:

Table 4. MSC mapping

MSC bit	MSCx_L/ LE/H	MSCx_L/ LE/H	MSCx_L/ LE/H	MSC2_LE/H	MSC2_L (only Dev5)	MSC2_L (only Dev5)
	x = (0..2)	x = (0..1)	x = (0..2)		MSC3_L/ LE/H	MSC3_L/LE/ H
0	TOM0_0	TOM1_0	TOM2_0	TOM3_0	TOM4_0	TOM5_0
1	TOM0_1	TOM1_1	TOM2_1	TOM3_1	TOM4_1	TOM5_1
2	TOM0_2	TOM1_2	TOM2_2	TOM3_2	TOM4_2	TOM5_2
3	TOM0_3	TOM1_3	TOM2_3	TOM3_3	TOM4_3	TOM5_3
4	TOM0_4	TOM1_4	TOM2_4	TOM3_4	TOM4_4	TOM5_4
5	TOM0_5	TOM1_5	TOM2_5	TOM3_5	TOM4_5	TOM5_5
6	TOM0_6	TOM1_6	TOM2_6	TOM3_6	TOM4_6	TOM5_6
7	TOM0_7	TOM1_7	TOM2_7	TOM3_7	TOM4_7	TOM5_7
8	TOM0_8	TOM1_8	TOM2_8	TOM3_8	TOM4_8	TOM5_8
9	TOM0_9	TOM1_9	TOM2_9	TOM3_9	TOM4_9	TOM5_9
10	TOM0_10	TOM1_10	TOM2_10	TOM3_10	TOM4_10	TOM5_10
11	TOM0_11	TOM1_11	TOM2_11	TOM3_11	TOM4_11	TOM5_11
12	TOM0_12	TOM1_12	TOM2_12	TOM3_12	TOM4_12	TOM5_12
13	TOM0_13	TOM1_13	TOM2_13	TOM3_13	TOM4_13	TOM5_13
14	TOM0_14	TOM1_14	TOM2_14	TOM3_14	TOM4_14	TOM5_14
15	TOM0_15	TOM1_15	TOM2_15	TOM3_15	TOM4_15	TOM5_15

This table is valid for all MSC busses (MSC0_L(E), MSC0_H, MSC1_L(E), MSC1_H, MSC2_L(E), MSC2_H, MSC3_L(E) and MSC3_H).

Test step commands

.

01 - PWM OUT - Setup

This command is used to setup a PWM output on a specified port/pin. The timer can be selected automatically (TIMRES = 0xFFFF) or manually.

Note 1 Any combination of ATOM and MSC is NOT supported!

Note 2 Due to limitations in timer to port mapping, some ports may have active signals if a timer associated with this port is running. Make sure that IOCR registers of unwanted ports are NOT connected to GTM with teststep GPIO.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	05	01	01	...

Parameters

Ports/Pins set for PWM set operation.

5	+0	+1	+2	+4	+6
(1..n*)	**PORT**	**PIN**	**PERIOD**	**DUTY**	**TIMRES**

Details

Ports/Pins set for PWM set operation.
Multiple parameter sets may be appended to create a complete sequence of port/pin and PWM settings.

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>PORT</td><td>1</td><td>xx</td><td>PWM port number
Port number of PWM output port.If "PORT" is >= 0x80 the port is an MSC port. See 'MSC Port Table'.</td></tr>
<tr><td>PIN</td><td>1</td><td>xx</td><td>PWM pin number
Pin number of PWM output. For MSC output, this is the bit position within the MSC frame.</td></tr>
<tr><td>PERIOD</td><td>2</td><td>xxxx</td><td>Period and resolution PWM
Format: RVVV where R=Resolution, V=Value of period. See 'Resolution Table'</td></tr>
<tr><td>DUTY</td><td>2</td><td>xxxx</td><td>Duty cycle of PWM or signal from FIFO
Resolution of duty cycle is 0.01%.

The output is active-low (i.e. DutyCycle=10% means 10% low and 90% high). Setting to 0% or 100% will force the pin to a static low/high level.

Note1: Before using PWM signal from FIFO, please see CMD 20!

Note2: Meaning of bits 14:0 depends on bit15.

Example1: 0x2710 means 100% duty cycle.

Example2: 0x8001 means PWM signal from FIFO with user defined PWM signal 2.

<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>15</td><td>Function bit</td></tr></table></td></tr>
</table>

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td><table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td></td><td>0_B: Duty cycle as given by bits 14:0
1_B: PWM signal from FIFO</td></tr><tr><td>14:0</td><td>Duty cycle or signal selection depends on bit 15
0_H: User defined PWM signal 1
1_H: User defined PWM signal 2</td></tr></table></td></tr>
<tr><td>TIMRES</td><td>2</td><td>xxxx</td><td>Desired timer resource
Timer resource that generates the signal. With TIMRES=0xFFFF the timer resource will be selected automatically within the limits of the current µC. See table 'MSC mapping'

Example: TIMRES = 0x0105: --> Select TPU_B channel 5 (FSL) resp. TOM1_5 (IFX/JDP).<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>15:8</td><td>Timer module (see table 'Timer Resources')</td></tr><tr><td>7:0</td><td>Timer channel (0-15)</td></tr></table></td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	05	xx	xx	01	...

Parameters

Response set of PwmOut operation

6	+0
(1..n*)	**TIMRES**

Details

Response set of PwmOut operation

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>TIMRES</td><td>2</td><td>xxxx</td><td>Desired timer resource
Timer module that generates the signal. With TIMRES=0xFFFF the timer resource will be selected automatically.<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>15:8</td><td>Timer module (see table 'Timer Resources')</td></tr></table></td></tr>
</table>

Name	Length	Fmt	Description	
			Bit(s)	**Description**
			7:0	**Timer channel (0-15)**

.

02 - PWM OUT - Link pins

Another pin or MSC bit will be connected or disconnected to an already running timer.

Most permutations between a pin and a MSC bit are possible (µC-specific limitations!). If both ports - the output port and reference port - are MSC, command 2 will enable another bit in the 32 bit MSC bus frame. Linking of MSC outputs is only working if both ports are running on the same MSC bus.

Note 1 Any combination of ATOM and MSC is NOT supported!

Note 2 Due to limitations in timer to port mapping, some ports may have active signals if a timer associated with this port is running. Make sure that IOCR registers of unwanted ports are NOT connected to GTM with teststep GPIO.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	05	02	01	...

Parameters

5	+0	+1	+2	+3	+4
(1..n*)	**REFPORT**	**REFPIN**	**PORT**	**PIN**	**UNLINK**

Details

Multiple parameter sets may be appended to create a complete sequence of PWM settings.

Name	Length	Fmt	Description
REFPORT	**1**	**xx**	**reference port** Port which is already running.
REFPIN	**1**	**xx**	**reference pin** Port which is already running.
PORT	**1**	**xx**	**port** Port which will be connected to the refport.
PIN	**1**	**xx**	**pin** Port which will be connected to the refport.
UNLINK	**1**	**xx**	**link/unlink selection** **0_H:** Link portpin **1_H:** Unlink portpin

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	05	xx	xx	01

.

03 - PWM OUT - Modify

Command 0x3 modifies the timer properties (period or duty cycle) of an already running signal.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	05	03	01	...

Parameters

5	+0	+2	+4
(1..n*)	**TIMRES**	**PERIOD**	**DUTY**

Details

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>TIMRES</td><td>2</td><td>xxxx</td><td>Desired timer resource
Timer module that generates the signal. With TIMRES=0xFFFF the timer resource will be selected automatically.

Example: TIMRES = 0x0105: --> Select TPU_B channel 5 (FSL) resp. TOM1_5 (IFX/JDP).

<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>15:8</td><td>Timer module (see table 'Timer Resources')</td></tr>
<tr><td>7:0</td><td>Timer channel (0-15)</td></tr>
</table></td></tr>
<tr><td>PERIOD</td><td>2</td><td>xxxx</td><td>Period
Period of the PWM signalFormat: RVVV where R=Resolution, V=Value of period. See 'Resolution Table'</td></tr>
<tr><td>DUTY</td><td>2</td><td>xxxx</td><td>Duty cycle
Duty cycle of the PWM signal (resolution is 0.01%).Duty cycle = 0% or 100% will force the pin to a static low/high level.</td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	05	xx	xx	01

.

04 - PWM OUT - Modify for One Shot Mode

Command 0x4 modifies the timer properties of an already running signal (started with command 0x1) for one shot mode operation.

Note 1: As a reset of the timer will be executed, the signal running on this timer will be interrupted.

Note 2: In case of positive pulses (IDLEV = LOW) a small unavoidable glitch (500-600ns) will occur during reset of timer.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	05	04	01	...

Parameters

5	+0	+1	+3	+4	+6
(1..n*)	**NUMOSM**	**TIMRES**	**IDLEV**	**DELAY**	**PULSE**

Details

Name	Length	Fmt	Description
NUMOSM	**1**	**xx**	**Number of signal** Number of one shot mode signal: PAV [0..1], ValSW [0..7].
TIMRES	**2**	**xxxx**	**Desired timer resource** Timer module that generates the signal. Example: TIMRES = 0x0105: --> Select TPU_B channel 5 (FSL) resp. TOM1_5 (IFX/JDP). Bit(s) – Description **15:8** – **Timer module (see table 'Timer Resources')** **7:0** – **Timer channel (0-15)**
IDLEV	**1**	**xx**	**Idle level** Idle level of signal. Pulse is inverted to this level.

Name	Length	Fmt	Description
			0x00: Idle level is LOW. 0x01: Idle level is HIGH.
DELAY	**2**	**xxxx**	**Delay** Delay of signalFormat: RVVV where R=Resolution, V=Value of delay. See 'Resolution Table' Note 1: Value "R" must be the same with DELAY and PULSE! Note 2: Values given here will be used for clock setting calculation. So suitable values (relating to values given with command 0x5) are necessary for correct generation of signals. Please do new clock setting by calling command 0x4 again with desired values.
PULSE	**2**	**xxxx**	**Pulse length** Pulse length of signalFormat: RVVV where R=Resolution, V=Value of delay. See 'Resolution Table' Note 1: Value "R" must be the same with DELAY and PULSE! Note 2: Values given here will be used for clock setting calculation. So suitable values (relating to values given with command 0x5) are necessary for correct generation of signals. Please do new clock setting by calling command 0x4 again with desired values.

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	05	xx	xx	01

.

05 - PWM OUT - Calculate registers for One Shot Mode

Command 0x5 calculates the registers for a timer (started with command 0x1 and modified with command 0x4) for one shot mode operation.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	05	05	01	...

Parameters

5	+0	+1	+3
(1..n*)	**NUMOSM**	**DELAY**	**PULSE**

Details

Name	Length	Fmt	Description
NUMOSM	**1**	**xx**	**Number of signal** Number of one shot mode signal: PAV [0..1], ValSW [0..7].
DELAY	**2**	**xxxx**	**Delay** Delay of signalFormat: RVVV where R=Resolution, V=Value of delay. See 'Resolution Table' Note 1: Value "R" must be the same with DELAY and PULSE! Note 2: If value leads to different clock setting (relating to values given in command 0x4), an error will occur. Please do new clock setting by calling command 0x4 again with desired value.
PULSE	**2**	**xxxx**	**Pulse length** Pulse length of signalFormat: RVVV where R=Resolution, V=Value of delay. See 'Resolution Table' Note 1: Value "R" must be the same with DELAY and PULSE! Note 2: If value leads to different clock setting (relating to values given in command 0x4), an error will occur. Please do new clock setting by calling command 0x4 again with desired value.

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	05	xx	xx	01

.

06 - PWM OUT - Trigger for One Shot Mode

Command 0x6 triggers the timer properties for one shot mode operation.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	05	06	01	...

Parameters

5	+0
(1..n*)	**NUMOSM**

Details

Name	Length	Fmt	Description
NUMOSM	**1**	**xx**	**Number of signal** Number of one shot mode signal: PAV [0..1], ValSW [0..7].

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	05	xx	xx	01

.

10 - PWM IN - Analyze a PWM signal

This command analyzes a PWM signal that is present at an input pin and returns its properties (period, duty cycle etc.).

Measurement mode can be switched between 'Normal' and 'Background' mode.

In 'Normal' mode measurement results will be returned immediately after successful acquisition of a signal. 'Init' mode must be called at least once, but can be called multiple times.

In 'Background' mode (only IFX and JDP) no measurement results will be returned. Results must be polled using 'Get result'. With 'Invalidate result' the last result could be invalidated.

Note: If signal period is longer than 3.35s or a signal dropout occurs a timer overflow error may be set. In 'Background' mode you should call 'Background measurement with init' (MODE = 0x3) to get new measurements with valid signals again.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	05	10	01	...

Parameters

5	+0	+1	+2	+4	+5	+6
(1..n*)	**PORT**	**PIN**	**TIMRES**	**MODE**	**AVG**	**TEDGE**

...	+7	+8	+9	+13	+17	+18
(1..n*)	**FMODE-R**	**FMODE-F**	**TIME-R**	**TIME-F**	**NO_ERR**	**TIMEOUT**

Details

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>PORT</td><td>1</td><td>xx</td><td>Port
Port number of PWM in.</td></tr>
<tr><td>PIN</td><td>1</td><td>xx</td><td>Pin
Pin number of PWM in.</td></tr>
<tr><td>TIMRES</td><td>2</td><td>xxxx</td><td>Timer resource
Timer resource that should analyze the signal. With TIMRES=0xFFFF the timer resource will be selected automatically.

Example: TIMRES = 0x0105: --> Select TPU_B channel 5 (FSL) resp. TIM1_5 (IFX/JDP).

<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>15:8</td><td>Timer module (see table 'Timer Resources')</td></tr>
<tr><td>7:0</td><td>Timer channel (0-15)</td></tr>
</table>
</td></tr>
<tr><td>MODE</td><td>1</td><td>xx</td><td>Mode selection

Measurement mode can be switched between 'Normal' and 'Background' mode.

In 'Normal' mode measurement results will be returned immediately after successful acquisition of a signal. 'Init' mode must be called at least once, but can be called multiple times.

In 'Background' mode (only IFX and JDP) no measurement results will be returned. Results must be polled using 'Get result'. With 'Invalidate result' the last result could be invalidated.

With the bit combinations given below, valid values for MODE are:

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x0</td><td>Normal measurement without init</td></tr>
<tr><td>0x1</td><td>Normal measurement with init</td></tr>
<tr><td>0x3</td><td>Background measurement with init</td></tr>
<tr><td>0x4</td><td>Get result from background measurement</td></tr>
<tr><td>0xC</td><td>Get result from background measurement and invalidate last result</td></tr>
</table>

<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:4</td><td>not used, should be 0</td></tr>
<tr><td>3</td><td>Invalidate last result (use only for measurements started in background measure mode)

0_B: No action</td></tr>
</table>
</td></tr>
</table>

Name	Length	Fmt	Description
			Bit(s) / **Description** (continued) **1_B:** Invalidate last result **2** — **Get result (use only for measurements started in background measure mode)** **0_B:** No action **1_B:** Get result of background measurement **1** — **Measure mode** **0_B:** Normal measure mode **1_B:** Background measure mode **0** — **Init mode** **0_B:** No init will be done **1_B:** Init will be done
AVG	**1**	**xx**	**Averaging** Note: Value will be ignored in 'Background' mode! **Bit(s)** / **Description** **7** — **Averaging** **0_B:** Return all samples **1_B:** Return only average **6:5** — **reserved** **4:0** — **Number of samples to take (0-23)**
TEDGE	**1**	**xx**	**Trigger edge selection** **0_H:** Trigger on falling edge **1_H:** Trigger on rising edge
FMODE-R	**1**	**xx**	**Filter mode for rising edge** IFX/JDP: for details see the respective manual of the GTM module (General Timer Module) **Bit(s)** / **Description** **7:2** — **reserved** **1** — **Filter counter mode** (only applicable in 'Individual de-glitch mode') **0_B:** Up/Down Counter **1_B:** Hold Counter **0** — **Filter mode** **0_B:** Immediate edge propagation mode

Name	Length	Fmt	Description
			Bit(s) / **Description** 1_B: Individual de-glitch mode
FMODE-F	**1**	**xx**	**Filter mode for falling edge** Filter mode for falling edge (see description for FMODE-R).
TIME-R	**4**	**xxxxxxxx**	**Filter parameter for rising edge** Set TIME-R and TIME-F = 0 to deactivate the filter stage. Note that TIME-R and TIME-F should be smaller than the pulse widths of the signal you try to analyze - otherwise you won't measure anything! Quantisation is in 12,5ns. Note: This register has different meanings in the various filter modes. 'Immediate edge propagation mode': acceptance time for rising edge. 'Individual deglitch time mode': deglitch time for rising edge
TIME-F	**4**	**xxxxxxxx**	**Filter parameter for falling edge** Filter parameter for falling edge (see description for TIME-R).
NO_ERR	**1**	**xx**	**Error behaviour** 0_H: Return error if measurement is invalid 1_H: Return 0xFF if measurement is invalid
TIMEOUT	**2**	**xxxx**	**Measurement timeout** Note 1: Value will be ignored in 'Background' mode! Note 2: Timeout should be long enough to cover time duration of all configured samples! Format: RVVV where R=Resolution, V=Value of timeout. See "Resolution table" in the introduction. See 'Resolution Table'

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	05	xx	xx	01	...

Parameters

6	+0	+2	+4	+5
(1..n*)	**TIMRES**	**PERIOD**	**DUTY**	**PDUR**

Details

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>TIMRES</td><td>2</td><td>xxxx</td><td>Timer resource
Timer module that analyzes the signal. With TIMRES=0xFFFF the timer resource will be selected automatically.

Example: TIMRES = 0x0105: --> Select TPU_B channel 5 (FSL) resp. TIM1_5 (IFX/JDP).
<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>15:8</td><td>Timer module (see table 'Timer Resources')</td></tr><tr><td>7:0</td><td>Timer channel (0-15)</td></tr></table></td></tr>
<tr><td>PERIOD</td><td>2</td><td>xxxx</td><td>Measured signal period
Format: RVVV where R=Resolution, V=Value of period. See "Resolution table" in the introduction. See 'Resolution Table'</td></tr>
<tr><td>DUTY</td><td>1</td><td>xx</td><td>Measured duty cycle
Resolution of duty cycle is 0.5%. Note: The duty cycle depends on the selection of TEDGE (if TEDGE='rising edge' then duty cycle = SignalHighTime / TotalPeriod, otherwise duty cycle = SignalLowTime / TotalPeriod)</td></tr>
<tr><td>PDUR</td><td>2</td><td>xxxx</td><td>Measure pulse duration
Note: The pulse duration depends on the selection of TEDGE (if TEDGE='rising edge' then it's the HighTime of the signal). Format: RVVV where R=Resolution, V=Value of pulsduration. See "Resolution table" in the introduction. See 'Resolution Table'</td></tr>
</table>

.

11 - PWM IN - Analyze a PWM signal with trigger signal output (only for IFX/JDP)

This command analyzes a PWM signal that is present at an input pin and returns its properties (period, duty cycle etc.).

Additional to command "PWM IN - Analyze a PWM signal" a output trigger port should be defined. Signal level of this port changes at start and end of measurement beginning with the level specified.

Note: If signal period is longer than 3.35s or a signal dropout occurs a timer overflow error may be set.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	05	11	01	...

Parameters

5 (1..n*)	+0 PORT	+1 PIN	+2 TIMRES	+4 MODE	+5 AVG	+6 TEDGE
... (1..n*)	+7 FMODE-R	+8 FMODE-F	+9 TIME-R	+13 TIME-F	+17 NO_ERR	+18 TIMEOUT
... (1..n*)	+20 PORTTRG	+21 PINTRG	+22 LVLTRG			

Details

Name	Length	Fmt	Description
PORT	**1**	**xx**	**Port** Port number of PWM in.
PIN	**1**	**xx**	**Pin** Pin number of PWM in.
TIMRES	**2**	**xxxx**	**Timer resource** Please refer to command "PWM IN - Analyze a PWM signal" for details.
MODE	**1**	**xx**	**Mode selection** In 'Normal' mode measurement results will be returned immediately after successful acquisition of a signal. 'Init' mode must be called at least once, but can be called multiple times. Note: Only "Normal measurement with or without init" is suitable here. With the bit combinations given below, valid values for MODE are: **Value** / **Description**: 0x0 — Normal measurement without init 0x1 — Normal measurement with init Please refer to command "PWM IN - Analyze a PWM signal" for details.
AVG	**1**	**xx**	**Averaging** Please refer to command "PWM IN - Analyze a PWM signal" for details.
TEDGE	**1**	**xx**	**Trigger edge selection** Please refer to command "PWM IN - Analyze a PWM signal" for details.
FMODE-R	**1**	**xx**	**Filter mode for rising edge** Please refer to command "PWM IN - Analyze a PWM signal" for details.
FMODE-F	**1**	**xx**	**Filter mode for falling edge**

Name	Length	Fmt	Description
			Please refer to command "PWM IN - Analyze a PWM signal" for details.
TIME-R	**4**	**xxxxxxxx**	**Filter parameter for rising edge** Please refer to command "PWM IN - Analyze a PWM signal" for details.
TIME-F	**4**	**xxxxxxxx**	**Filter parameter for falling edge** Please refer to command "PWM IN - Analyze a PWM signal" for details.
NO_ERR	**1**	**xx**	**Error behaviour** Please refer to command "PWM IN - Analyze a PWM signal" for details.
TIMEOUT	**2**	**xxxx**	**Measurement timeout** Please refer to command "PWM IN - Analyze a PWM signal" for details.
PORTTRG	**1**	**xx**	**Trigger port** Port number of trigger signal port.
PINTRG	**1**	**xx**	**Trigger pin** Pin number of trigger signal pin.
LVLTRG	**1**	**xx**	**Level of trigger signal** Level of trigger signal. **Value** / **Description** 0x0: Pin is switched low during measurement. 0x1: Pin is switched high during measurement.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	05	xx	xx	01	...

Parameters

6	+0	+2	+4	+5
(1..n*)	**TIMRES**	**PERIOD**	**DUTY**	**PDUR**

Details

Name	Length	Fmt	Description
TIMRES	**2**	**xxxx**	**Timer resource** Please refer to command "PWM IN - Analyze a PWM signal" for details.
PERIOD	**2**	**xxxx**	**Measured signal period**

Name	Length	Fmt	Description
			Please refer to command "PWM IN - Analyze a PWM signal" for details.
DUTY	**1**	**xx**	**Measured duty cycle** Please refer to command "PWM IN - Analyze a PWM signal" for details.
PDUR	**2**	**xxxx**	**Measure pulse duration** Please refer to command "PWM IN - Analyze a PWM signal" for details.

.

20 - PWM OUT - User defined PWM signal according to data from FIFO

This command may be used to initilaise GTM FIFO with defined PWM signal data. The given data below (period duration and duty cycles) are stored to FIFO, and will be send continuously to selected ATOM timer at CMD 01. There is possibility to use two FIFO channels independently. It is possible to fill FIFO with default or user defined PWM signal. Default PWM signal: 2us period duration (500kHz) and 50 different duty cycles vary between 100% and 0%.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	05	20	01	...

Parameters

5	+0	+1
(1*)	**FSTP**	**FPERIOD**

64 times repeatable parameter for FIFO

8	+0
(1..n*)	**FDUTY**

Details

Name	Length	Fmt	Description
FSTP	**1**	**xx**	**FIFO setup** **Bit(s)** / **Description** **7:4** — **FIFO channel** **0_H:** Channel 1 **1_H:** Channel 2 **3:0** — **PWM signal type**

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td><table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td></td><td>0_H: Default
1_H: User defined</td></tr></table></td></tr>
<tr><td>FPERIOD</td><td>2</td><td>xxxx</td><td>Period duration
Format: RVVV where R=Resolution, V=Value of period. See 'Resolution Table'This value shall be set to 0x0000 if default PWM signal selected.</td></tr>
</table>

64 times repeatable parameter for FIFO

Name	Length	Fmt	Description
FDUTY	**2**	**xxxx**	**Duty cycle** Duty cycle of the user defined PWM signal. Max 64 times repeatable.Resolution is 0.01%. The output is active-high. (i.e. DutyCycle=10% means 10% high and 90% low). Duty cycle = 0% or 100% will force the pin to a static low/high level.This value shall be set to 0x0000 if default PWM signal selected.

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	05	xx	xx	01

.

30 - PWM - Disable port

Running Port/Pin or MSC bit will be disabled.

In contrast to the disable function within command 0x2 "PWM OUT - Link pins", this command may lead to a complete disable of a port/pin (orphaned timer). Also registered ressources will be cleared. Timers will NOT be disabled!

Note: If any linked MSC bits are found, no disabling of MSC bits is possible. Reason: Memory includes no timer information for these linked MSC bits. So probably disconnecting of main MSC bit leads to non assignable data in memory. Please disconnect any linked MSC bits first with command 0x2 "PWM OUT - Link pins".

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	05	30	01	...

Parameters

Parameter set for port disable

5	+0	+1	+2	+4
(1..n*)	**PORT**	**PIN**	**TIMRES**	**DIR**

Details

Parameter set for port disable
Multiple parameter sets may be appended to create a complete sequence of PWM settings.

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>PORT</td><td>1</td><td>xx</td><td>PWM port number
Port which should be disabled.</td></tr>
<tr><td>PIN</td><td>1</td><td>xx</td><td>PWM pin number
Pin which should be disabled.</td></tr>
<tr><td>TIMRES</td><td>2</td><td>xxxx</td><td>Used timer resource
Timer that generates/ measures the signal on port/ pin.
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>15:8</td><td>Timer module (see table 'Timer Resources')</td></tr>
<tr><td>7:0</td><td>Timer channel (0-15)</td></tr>
</table></td></tr>
<tr><td>DIR</td><td>1</td><td>xx</td><td>Direction of signal
Direction of port/pin (PWM Out, PWM In).
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>15:1</td><td>not used</td></tr>
<tr><td>0</td><td>1: PWM In</td></tr>
<tr><td>0</td><td>0: PWM Out</td></tr>
</table></td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	05	xx	xx	01

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)

Code	Description
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**TSW: All timers occupied.**
0x02	**TSW: Requested timer occupied.**
0x03	**TSW: Requested portpin occupied**
0x04	**TSW: Combination of portpin and timer unplausible.**
0x05	**TSW: Timer resource is not running.**
0x06	**TSW: Ports on different timers.**
0x07	**TSW: Port not connected to timer.**
0x08	**TSW: Ports not connected.**
0x09	**TSW: PWM-In via MSC not possible.**
0x0A	**TSW: PWM-In port not connectable.**
0x0B	**TSW: Timer already connected to a PWM-In channel.**
0x0C	**TSW: Buffer full. Max (20[MDG1]/64[FSL])) number of timer connections reached. This limit includes timer assignments and links. It doesn't apply to VALSW.**
0x0D	**TSW: Requested portpin is a PWM-Out channel.**
0x0E	**TSW: Requested mode not supported.**
0x0F	**TSW: Enhanced output (IFX/JDP: ATOM) not supported.**

Code	Description
0x10	TSW: Duty value is over the limit.
0x11	TSW: Amount of FIFO FDUTY data is over the limit.
0x12	TSW: Any MSC linked pins found. Please disconnect with CMD_02.
0x13	TSW: Value 'R' of 'RVVV' must be the same for DELAY and PULSE.
0x14	TSW: Timer clock different.
0x15	TSW: Invalid one shot mode signal number.
0x16	TSW: Command not supported.
0xFF	HAL busy
0xFE	Generic HAL error
0xFD	HAL not compatible to current CPU
0xFC	Invalid params
0xFB	HAL module not implemented
0xFA	HAL function not implemented
0xF9	HAL timeout
0xED	Invalid MSC port
0xEE	FIFO is empty
0xEF	Timer overflow

TSW_06 - I2C communication

Test step description

Date: 20.02.2013

This TSW module provides low level I2C (Inter-Integrated Circuit) communication.

(only for MDG1, not for E84)

Test step commands

.

00 - Initialize/Reset I2C module

This command resets and initialize the I2C (Inter-Integrated Circuit) module.

Note: This command is optional. The module is initalized on load of the teststep.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	06	00	01

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	06	xx	xx	01

.

01 - Define i2c pins

This command defines two controller pins used for i2c communication.

Note: the used pins have to be initialized with TSW_04.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	06	01	01	...

Parameters

parameters to select a I2C controller for the device

5	+0	+1	+2	+3
(1*)	**POSCL**	**PISCL**	**POSDA**	**PISDA**

Details

parameters to select a I2C controller for the device
Port and Pin parameters to select the I2C controller where the device is connected

Name	Length	Fmt	Description
POSCL	**1**	**xx**	**Port number of SCL signal** Port number of SCL (serial clock) signal
PISCL	**1**	**xx**	**Pin number of SCL signal** Pin number within POSCL of SCL (serial clock) signal
POSDA	**1**	**xx**	**Port number of SDA signal** Port number of SDA (serial data) signal
PISDA	**1**	**xx**	**Pin number of SDA signal** Pin number within POSDA of SDA (serial data) signal

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	06	xx	xx	01

.

02 - Set baudrate

Command to set the baudrate on the i2c bus.

Note: This command is optional. A baudrate of approximately 100kHz is set on load of the teststep. If this is sufficient there is no need to use this command.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	06	02	01	...

Parameters

parameters to set baudrate

5	+0
(1*)	**RATE**

Details

parameters to set baudrate

Name	Length	Fmt	Description
RATE	**2**	**xxxx**	**baudrate** the baudrate in kHz (1-400)

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	06	xx	xx	01

.

10 - Read i2c data

Command to read data from i2c devices.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	06	10	01	...

Parameters

parameters to read data

5	+0	+1
(1..n*)	**ADDR**	**DLEN**

Details

parameters to read data

Name	Length	Fmt	Description
ADDR	**1**	**xx**	**device address** i2c address of the device
DLEN	**1**	**xx**	**number of data to read** number of data to read (1-4)

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	06	xx	xx	01	...

Parameters

response of read request

6	+0	+1
(1..n*)	**DLEN**	**DATA**

Details

response of read request

Name	Length	Fmt	Description
DLEN	**1**	**xx**	**number of received data**

Name	Length	Fmt	Description
			number of received data
DATA	**1**	**xx..**	**received data [0-4]**

.

11 - Write i2c data

Command to write data to i2c devices.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	06	11	01	...

Parameters

parameters to write data

5	+0	+1
(1..n*)	**DLEN**	**DATA**

Details

parameters to write data

Name	Length	Fmt	Description
DLEN	**1**	**xx**	**number of data to write** number of data to write (1-32)
DATA	**1**	**xx..**	**data to write** 1 to 32 bytes to transmitt to the i2c devices.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	06	xx	xx	01	...

Parameters

response of write request

6	+0
(1..n*)	**DLEN**

Details

response of write request

Name	Length	Fmt	Description
DLEN	**1**	**xx**	**number of transmitted data**

Name	Length	Fmt	Description
			number of transmitted data

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**

Code	Description
0xFF	**HAL busy**
0xFE	**Generic HAL error**
0xFD	**HAL not compatible to current CPU**
0xFC	**Invalid params**
0xFB	**HAL module not implemented**
0xFA	**HAL function not implemented**
0xF9	**HAL timeout**
0xEF	**i2c-bus busy (3, slave mode)**
0xEE	**i2c-bus busy (2, master mode)**
0xED	**i2c-bus busy (1, start condition detected)**
0xEC	**pin not available for i2c**

TSW_07 - Delta Sigma ADC

Test step description

This test step is used to read a analog values with Delta Sigma converters.

Table 1. IFX Device 4 input mapping

IFX Input name	Input Number(ANx)
P00.8	52
P00.7	53
P00.2	60
P00.1	61
AN52	68
AN53	69
AN54	70
AN55	71
AN60	76
AN61	77
AN62	78
AN63	79
AN68	84
AN69	85
AN70	86
AN71	87

Table 2. IFX Device 5 input mapping

IFX Input name	Input Number(ANx)
P00.8	52
P00.7	53
P00.10	54
P00.9	55
P00.2	60
P00.1	61
P00.4	62
P00.3	63
AN50	66
AN51	67
AN52	68
AN53	69
AN54	70
AN55	71
AN58	74

IFX Input name	Input Number(ANx)
AN59	75
AN60	76
AN61	77
AN62	78
AN63	79
AN66	82
AN67	83
AN68	84
AN69	85
AN70	86
AN71	87
AN72	88
AN73	89

Test step commands

.

01 - Initialize DS ADC module

If you don´t need the DSADC module anymore, call the terminate command (CMD20). This is necessary to free bus resources.

Configuration table for output datarate of SDADC conversion.

Use only the green marked settings. Other values can be used, but resulting output rates are not reachable by all devices.

Output sample rate = Input sample rate / oversampling rate

Note

This command will delete all settings made by a previous Initialize call.

Note

One DS ADC module can convert one analog input only. If you like to read multiple inputs (not for differential signals) of the same DS-ADC module, you have to initialize one, read it and after that initialize the second one and read it.

Note

Conversion result range in the table below gives the maximum measurable decimal value for the corresponding sample rate. This maximum value represents 5 Volts and has to be used for calculating the measured voltage. Example: Conversion result range is 18317 (@208,333kHz), the measured decimal value is 10276. Measured voltage is: 10276 * 5 V / 18317 = 2,805 V

Table 3. IFX

Input sample rate (MHz)	oversampling	output sample rate (kHz)	conversion result range
20	16	1250	21709
20	20	1000	21200
20	24	833,3333333	18317

Input sample rate (MHz)	**oversampling**	**output sample rate (kHz)**	**conversion result range**
20	28	714,2857143	14543
20	32	625	21709
20	36	555,5555556	15455
20	40	500	21200
20	44	454,5454545	14109
20	48	416,6666667	18317
20	52	384,6153846	11644
20	56	357,1428571	14543
20	60	333,3333333	17887
20	64	312,5	21709
20	68	294,1176471	13019
20	72	277,7777778	15455
20	76	263,1578947	18176
20	80	250	21200
20	84	238,0952381	12271
20	88	227,2727273	14109
20	92	217,3913043	16121
20	96	208,3333333	18317
20	100	200	20703
20	104	192,3076923	23288
20	108	185,1851852	13040
20	112	178,5714286	14543
20	116	172,4137931	16158
20	120	166,6666667	17887
20	124	161,2903226	19736
20	128	156,25	21709
20	132	151,5151515	11904
20	136	147,0588235	13019
20	140	142,8571429	14202
20	144	138,8888889	15455
20	148	135,1351351	16779
20	152	131,5789474	18176
20	156	128,2051282	19649
20	160	125	21200
20	164	121,9512195	11415
20	168	119,047619	12271
20	172	116,2790698	13168
20	176	113,6363636	14109
20	180	111,1111111	15093
20	184	108,6956522	16121

Input sample rate (MHz)	**oversampling**	**output sample rate (kHz)**	**conversion result range**
20	188	106,3829787	17196
20	192	104,1666667	18317
20	196	102,0408163	19486
20	200	100	20703
20	204	98,03921569	10985
20	208	96,15384615	11644
20	212	94,33962264	12329
20	216	92,59259259	13040
20	220	90,90909091	13778
20	224	89,28571429	14543
20	228	87,71929825	15336
20	232	86,20689655	16158
20	236	84,74576271	17008
20	240	83,33333333	17887
20	244	81,96721311	18797
20	248	80,64516129	19736
20	252	79,36507937	20707
20	256	78,125	21709
20	260	76,92307692	11371
20	264	75,75757576	11904
20	268	74,62686567	12453
20	272	73,52941176	13019
20	276	72,46376812	13602
20	280	71,42857143	14202
20	284	70,42253521	14820
20	288	69,44444444	15455
20	292	68,49315068	16108
20	296	67,56756757	16779
20	300	66,66666667	17468
20	304	65,78947368	18176
20	308	64,93506494	18903
20	312	64,1025641	19649
20	316	63,29113924	20415
20	320	62,5	21200
20	324	61,72839506	11002
20	328	60,97560976	11415
20	332	60,24096386	11838
20	336	59,52380952	12271
20	340	58,82352941	12714
20	344	58,13953488	13168

Input sample rate (MHz)	**oversampling**	**output sample rate (kHz)**	**conversion result range**
20	348	57,47126437	13633
20	352	56,81818182	14109
20	356	56,17977528	14595
20	360	55,55555556	15093
20	364	54,94505495	15601
20	368	54,34782609	16121
20	372	53,76344086	16653
20	376	53,19148936	17196
20	380	52,63157895	17750
20	384	52,08333333	18317
20	388	51,54639175	18895
20	392	51,02040816	19486
20	396	50,50505051	20088
20	400	50	20703
20	404	49,5049505	21330
20	408	49,01960784	10985
20	412	48,54368932	11311
20	416	48,07692308	11644
20	420	47,61904762	11983
20	424	47,16981132	12329
20	428	46,72897196	12681
20	432	46,2962963	13040
20	436	45,87155963	13406
20	440	45,45454545	13778
20	444	45,04504505	14157
20	448	44,64285714	14543
20	452	44,24778761	14936
20	456	43,85964912	15336
20	460	43,47826087	15743
20	464	43,10344828	16158
20	468	42,73504274	16579
20	472	42,37288136	17008
20	476	42,01680672	17444
20	480	41,66666667	17887
20	484	41,32231405	18338
20	488	40,98360656	18797
20	492	40,6504065	19263
20	496	40,32258065	19736
20	500	40	20218
20	504	39,68253968	20707

Input sample rate (MHz)	**oversampling**	**output sample rate (kHz)**	**conversion result range**
20	508	39,37007874	21204
20	512	39,0625	21709
10	260	38,46153846	11371
10	264	37,87878788	11904
10	268	37,31343284	12453
10	272	36,76470588	13019
10	276	36,23188406	13602
10	280	35,71428571	14202
10	284	35,21126761	14820
10	288	34,72222222	15455
10	292	34,24657534	16108
10	296	33,78378378	16779
10	300	33,33333333	17468
10	304	32,89473684	18176
10	308	32,46753247	18903
10	312	32,05128205	19649
10	316	31,64556962	20415
10	320	31,25	21200

Table 4. IFX Dev5

Input sample rate (MHz)	**oversampling**	**output sample rate (kHz)**	**conversion result range**
40	120	333,3	31129
40	160	250	31129
40	200	200	31129
40	240	166,67	31129
40	280	142,857143	31129
40	320	125	31129
40	360	111,1	31129
40	400	100	31129
40	440	90,9090909	31129
40	480	83,3	31129
40	640	62,5	31129
40	720	55,5	31129
40	800	50	31129
40	880	45,4545455	31129
40	960	41,6	31129
20	60	333,3	31129
20	80	250	31129
20	100	200	31129
20	120	166,6	31129

Input sample rate (MHz)	**oversampling**	**output sample rate (kHz)**	**conversion result range**
20	140	142,857143	31129
20	160	125	31129
20	180	111,1	31129
20	200	100	31129
20	220	90,9090909	31129
20	240	83,3	31129
20	280	71,4285714	31129
20	320	62,5	31129
20	360	55,5	31129
20	400	50	31129
20	440	45,4545455	31129
20	480	41,6	31129
20	560	35,7142857	31129
20	640	31,25	31129
20	1280	15,625	31129
16	48	333,3	31129
16	56	285,714286	31129
16	64	250	31129
16	72	222,2	31129
16	80	200	31129
16	88	181,818182	31129
16	96	166,6	31129
16	128	125	31129
16	144	111,1	31129
16	160	100	31129
16	176	90,9090909	31129
16	192	83,3	31129
16	224	71,4285714	31129
16	256	62,5	31129
16	288	55,5	31129
16	320	50	31129
16	352	45,4545455	31129
16	384	41,6	31129
16	448	35,7142857	31129
16	512	31,25	31129
16	2048	7,8125	31129

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	07	01	01	...

Parameters

Parameter set for initialization of DS ADC module

5	+0	+1	+2	+3	+4	+5
(1*)	**INP**	**INN**	**MODE**	**GAIN**	**INSR**	**OSR**

Details

Parameter set for initialization of DS ADC module

Name	Length	Fmt	Description
INP	**1**	**xx**	**Input positiv** ADC Input P(ANx). Please refer table "IFX Device input mapping".
INN	**1**	**xx**	**Input negative** ADC Input N(ANx). (Only relevant for differential input mode) Please refer table "IFX Device input mapping".
MODE	**1**	**xx**	**Input mode** Mode selection 0 – Differential input mode selected 1 – Single-ended input mode selected
GAIN	**1**	**xx**	**Gain** This field selects the gain to be applied to the analog input stage of the SDADC. The effective analog input becomes the input voltage level multiplied by the gain factor. IFX Dev3 - Dev4: 1,2,4,8,16 IFX Dev5: 1,2,4
INSR	**1**	**xx**	**Input sample rate** Not all values are possible for input sample rate. Maximum input sample rate is 40 MHz. Unit is MHz. Common: 20 MHz
OSR	**2**	**xxxx**	**Oversampling rate** Not all values are possible for oversampling rate. Common: 200 Max Dev3-Dev4: 1024 Max Dev5: 2048

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	07	xx	xx	01	...

Parameters

Command response data

6	+0	+1
(1*)	**INSR**	**OSR**

Details

Command response data
Command response data.

Name	Length	Fmt	Description
INSR	**1**	**xx**	**Input sample rate** Used input sample rate in Mhz.
OSR	**2**	**xxxx**	**Oversampling rate** Used Oversampling rate.

.

10 - Average DS ADC conversions

Do a defined number of conversions and build an average.

Interpreting the result:

The result value is not scaled or calibrated! The range of the value depends on the underlaying platform type. Following information is from "DS-ADC Verwendung in µC1 – Controllern, Version 0.7" with date 2014-02-10.

Note

Negativ values are stored in a 16-bit two’s complement format.

DEV2 B-Step,DEV3 D-Step, DEV4 B_Step: 5 V on the positive input will result in 20703 dec with default output sample rate of 200 kHz.

DEV5: 5 V on the positive input will result in 31129 dec with default output sample rate of 200 kHz.

OTHER: 5 V on the positive input will result in 21613 dec with default output sample rate of 200 kHz.

Check table in command 01 description for other output sample rates.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	07	10	01	...

Parameters

Parameter set for DS ADC messurement

5	+0	+1
(1*)	**PIN**	**NSA**

Details

Parameter set for DS ADC messurement

Name	Length	Fmt	Description
PIN	**1**	**xx**	**DS ADC Input pin** Pin(ANx) of the DSADC module, which should be used. In differential input mode any of both pins can be used! Please refer table "IFX Dev4 / Dev5 input mapping".
NSA	**1**	**xx**	**Number of Samples**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	07	xx	xx	01	...

Parameters

Command response data

6	+0	+1	+3	+5
(1*)	**RES**	**RSLT**	**MIN**	**MAX**

Details

Command response data

Name	Length	Fmt	Description
RES	**1**	**xx**	**Reserved** reserved for future use.
RSLT	**2**	**xxxx**	**Result of conversion** Average result of conversion. Note: Value is not scaled. Value is exactly the conversion result.
MIN	**2**	**xxxx**	**Minimum of all conversion results** Minimum of all conversion results. Note: Value is not scaled. Value is exactly the conversion result.
MAX	**2**	**xxxx**	**Maximum of all conversion results**

Name	Length	Fmt	Description
			Maximum of all conversion results. Note: Value is not scaled. Value is exactly the conversion result.

.

11 - All sample measurement

Do a defined number of conversions and return all sample results.

The DS ADC starts convertion after the "Init". With this command a snapshot of a defined number of these values is taken.

Interpreting the result: see command "Average DS ADC conversions" (10)

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	07	11	01	...

Parameters

Parameter set for DS ADC messurement

5	+0	+1
(1*)	**PIN**	**NSA**

Details

Parameter set for DS ADC messurement

Name	Length	Fmt	Description
PIN	**1**	**xx**	**DS ADC Input pin** Pin(ANx) of the DSADC module, which should be used. In differential input mode any of both pins can be used! Please refer table "IFX Device 4 input mapping".
NSA	**1**	**xx**	**Number of Samples** Sampling of DS ADC starts after Init. Specify the number of values to be read. Max. 0x7C because KWP2000 restrictions. Larger values are set to 0x7C internally.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	07	xx	xx	01	...

Parameters

Command response data.

6	+0
(1*)	**RES**

Conversion results

7	+0
(1..n*)	**RSLT**

Details

Command response data.

Name	Length	Fmt	Description
RES	**1**	**xx**	**Reserved** reserved for future use.

Conversion results

Name	Length	Fmt	Description
RSLT	**2**	**xxxx**	**Result of conversion** Result of conversion. Note: Value is not scaled. Value is exactly the conversion result.

.

20 - Terminate DS ADC module

This command is used to free bus resources.

Note

only for JDP devices

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	07	20	01	...

Parameters

Parameter set for termination of DS ADC module

5	+0
(1*)	**PIN**

Details

Parameter set for termination of DS ADC module

Name	Length	Fmt	Description
PIN	**1**	**xx**	**DS ADC Input pin** Pin(ANx) of the used DSADC module, or 0xFF to terminate all DSADC modules.

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	07	xx	xx	01

.

30 - Average DS ADC conversions with scaled values

Do a defined number of conversions and build an average.

Interpreting the result:

The conversion results are scaled to maximum value of 0x7FFF. The result for voltages greater 5 V will be set to the maximum value 0x7FFF. Negativ values are not stored in a 16-bit two’s complement format. They are the positiv value with the first Bit set.

Example: 0x80A0 = -160d

Note

The scaling has a small error because of floating point operations.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	07	30	01	...

Parameters

Parameter set for DS ADC messurement

5	+0	+1
(1*)	**PIN**	**NSA**

Details

Parameter set for DS ADC messurement

Name	Length	Fmt	Description
PIN	**1**	**xx**	**DS ADC Input pin** Pin(ANx) of the DSADC module, which should be used. In differential input mode any of both pins can be used!

Name	Length	Fmt	Description
			Please refer table "IFX Dev4 / Dev5 input mapping".
NSA	**1**	**xx**	**Number of Samples**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	07	xx	xx	01	...

Parameters

Command response data

6	+0	+1	+3	+5
(1*)	**RES**	**RSLT**	**MIN**	**MAX**

Details

Command response data

Name	Length	Fmt	Description
RES	**1**	**xx**	**Reserved** reserved for future use.
RSLT	**2**	**xxxx**	**Result of conversion** Average result of conversion. Note: Value is scaled to maximum value of 0x7FFF.
MIN	**2**	**xxxx**	**Minimum of all conversion results** Minimum of all conversion results. Note: Value is scaled to maximum value of 0x7FFF.
MAX	**2**	**xxxx**	**Maximum of all conversion results** Maximum of all conversion results. Note: Value is scaled to maximum value of 0x7FFF.

.

31 - All sample measurement with scaled values

Do a defined number of conversions and return all sample results.

The DS ADC starts convertion after the "Init". With this command a snapshot of a defined number of these values is taken.

Interpreting the result: see command "Average DS ADC conversions" (30)

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	07	31	01	...

Parameters

Parameter set for DS ADC messurement

5	+0	+1
(1*)	**PIN**	**NSA**

Details

Parameter set for DS ADC messurement

Name	Length	Fmt	Description
PIN	**1**	**xx**	**DS ADC Input pin** Pin(ANx) of the DSADC module, which should be used. In differential input mode any of both pins can be used! Please refer table "IFX Device 4 input mapping".
NSA	**1**	**xx**	**Number of Samples** Sampling of DS ADC starts after Init. Specify the number of values to be read. Max. 0x7C because KWP2000 restrictions. Larger values are set to 0x7C internally.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	07	xx	xx	01	...

Parameters

Command response data.

6	+0
(1*)	**RES**

Conversion results

7	+0
(1..n*)	**RSLT**

Details

Command response data.

Name	Length	Fmt	Description
RES	**1**	**xx**	**Reserved** reserved for future use.

Conversion results

Name	Length	Fmt	Description
RSLT	**2**	**xxxx**	**Result of conversion** Result of conversion. Note: Value is scaled to maximum value of 0x7FFF.

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)

Code	Description
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0xEF	**Invalid Parameter**
0xEE	**Pin not supported**
0xED	**Timeout**
0xEC	**DMA error**

TSW_08 - C2C Communication

Test step description

Date: 20.04.2016

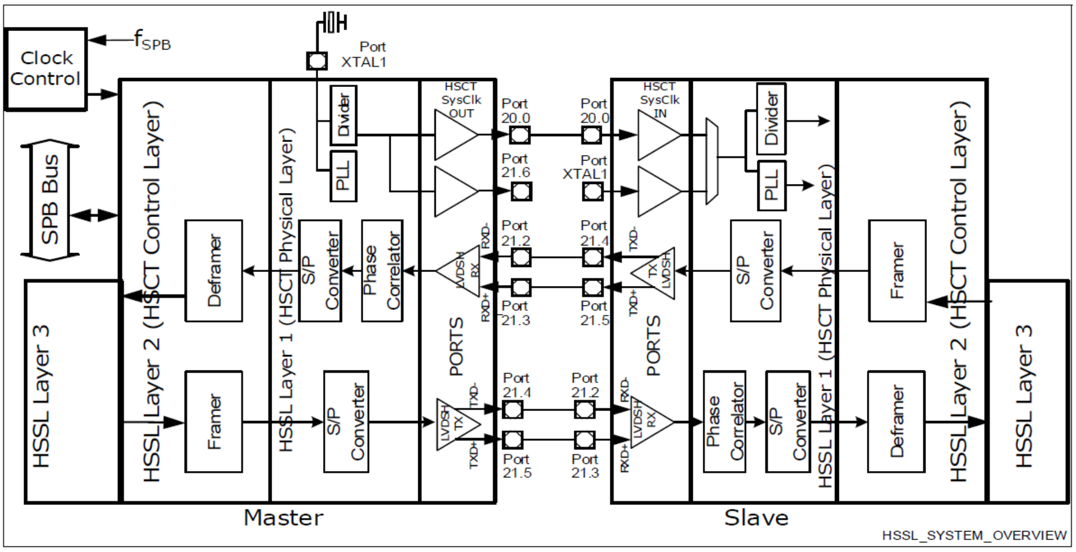
This test step is intended to enable Chip-to-chip (C2C) bus communication.

Tip

At slow speed - before switching to high speed mode with CMD01 - the communication runs at half speed of reference clock. After switching to high speed mode the setting of baud rate at CMD00 becomes valid.

For direct controller to controller communication a C2C bus interface is available at MDG1. Currently only IFX devices 3 and 4 are supported.

System overview for C2C bus at ifx implementation



Test step commands

.

00 - C2C initialization

This command configures basic communication parameters of a C2C bus. At initialization Master and Slave are set to slow communication speed.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	08	00	01	...

Parameters

Parameter set for C2C bus initialization.

5 (1*)	+0 PORTTXP	+1 PINTXP	+2 PORTTXN	+3 PINTXN	+4 PORTRXP	+5 PINRXP

... (1*)	+6 PORTRXN	+7 PINRXN	+8 PORTCLK	+9 PINCLK	+10 MOD	+11 BAUDR

... (1*)	+13 TIMOUT

Details

Parameter set for C2C bus initialization.

Name	Length	Fmt	Description
PORTTXP	**1**	**xx**	**Port for C2C TxD positive line**
PINTXP	**1**	**xx**	**Pin for C2C TxD positive line**
PORTTXN	**1**	**xx**	**Port for C2C TxD negative line**
PINTXN	**1**	**xx**	**Pin for C2C TxD negative line**
PORTRXP	**1**	**xx**	**Port for C2C RxD positive line**
PINRXP	**1**	**xx**	**Pin for C2C RxD positive line**
PORTRXN	**1**	**xx**	**Port for C2C RxD negative line**
PINRXN	**1**	**xx**	**Pin for C2C RxD negative line**
PORTCLK	**1**	**xx**	**Port for C2C clock line**
PINCLK	**1**	**xx**	**Pin for C2C clock line**
MOD	**1**	**xx**	**C2C mode** Definition of C2C operation mode **Bit(s)** / **Description** **7:2** — **reserved - should be set to 0** **1** — **Reference clock** Reference clock of LFAST / HSCT. Please take care to take same setting at master and slave controller. **0_B:** 10Mhz **1_B:** 20Mhz **0** — **MASTER** **0_B:** slave **1_B:** master
BAUDR	**2**	**xxxx**	**Baudrate** C2C baud rate setting. Valid range: 120 to 320 MBaud. Step width depends on reference clock setting (10 or 20MBaud).

Name	Length	Fmt	Description
			Please take care to take same setting at master and slave controller. Format: RVVV where R=Resolution, V=Value (For example: 0x6005 = 5MBit/s)
TIMOUT	**2**	**xxxx**	**Timeout** Timeout for attempt to establish connection to C2C partner. On failure the attempt to establish connection is repeated each 50ms. With this time the number of attempts for repetition is defined. Valid range: up to 30s Please consider for slave mode that a timeout means that in the given time no master action was recognized. Nevertheless even if a timeout happened the connection can be established afterwards successfully by master if slave initialize was performed before. Format: RVVV where R=Resolution, V=Value (For example: 0xF001 = 100ms)

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	08	xx	xx	01

.

01 - C2C set speed

This command changes speed rate of master and slave. This command can only be executed by master!

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	08	01	01	...

Parameters

Parameter set for C2C bus bus speed.

5 (1*)	+0	+1	+2	+3	+4	+5
	PORTTXP	**PINTXP**	**PORTTXN**	**PINTXN**	**PORTRXP**	**PINRXP**

... (1*)	+6	+7	+8	+9	+10
	PORTRXN	**PINRXN**	**PORTCLK**	**PINCLK**	**MOD**

Details

Parameter set for C2C bus bus speed.

Name	Length	Fmt	Description
PORTTXP	**1**	**xx**	**Port for C2C TxD positive line** 0xFF: PINTXP is interpreted as C2C handle ID for faster access
PINTXP	**1**	**xx**	**Pin for C2C TxD positive line** If PORTTXP was 0xFF, this is used as C2C handle ID. These IDs are in the same order the C2C initialization where done using CMD_00, starting with 0. maximum: 2 entries
PORTTXN	**1**	**xx**	**Port for C2C TxD negative line**
PINTXN	**1**	**xx**	**Pin for C2C TxD negative line**
PORTRXP	**1**	**xx**	**Port for C2C RxD positive line**
PINRXP	**1**	**xx**	**Pin for C2C RxD positive line**
PORTRXN	**1**	**xx**	**Port for C2C RxD negative line**
PINRXN	**1**	**xx**	**Pin for C2C RxD negative line**
PORTCLK	**1**	**xx**	**Port for C2C clock line**
PINCLK	**1**	**xx**	**Pin for C2C clock line**
MOD	**1**	**xx**	**C2C mode** Definition of C2C operation mode

MOD bits:

Bit(s)	Description
7:2	**reserved - should be set to 0**
1	**MASTERSPEED** Master Tx Speed **0_B:** slow **1_B:** fast
0	**SLAVESPEED** Slave Tx Speed Medium speed mode according to controller manual is not supported. **0_B:** slow **1_B:** fast

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	08	xx	xx	01

.

10 - C2C single read

This command performs single read operation from connected controllers memory.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	08	10	01	...

Parameters

Parameter set for C2C single read.

5 (1*)	+0	+1	+2	+3	+4	+5
	PORTTXP	**PINTXP**	**PORTTXN**	**PINTXN**	**PORTRXP**	**PINRXP**

... (1*)	+6	+7	+8	+9	+10	+11
	PORTRXN	**PINRXN**	**PORTCLK**	**PINCLK**	**MOD**	**CH**

... (1*)	+12	+16
	ADR	**DSIZE**

Details

Parameter set for C2C single read.

Name	Length	Fmt	Description
PORTTXP	**1**	**xx**	**Port for C2C TxD positive line** 0xFF: PINTXP is interpreted as C2C handle ID for faster access
PINTXP	**1**	**xx**	**Pin for C2C TxD positive line** If PORTTXP was 0xFF, this is used as C2C handle ID. These IDs are in the same order the C2C initialization where done using CMD_00, starting with 0. maximum: 2 entries
PORTTXN	**1**	**xx**	**Port for C2C TxD negative line**
PINTXN	**1**	**xx**	**Pin for C2C TxD negative line**
PORTRXP	**1**	**xx**	**Port for C2C RxD positive line**
PINRXP	**1**	**xx**	**Pin for C2C RxD positive line**
PORTRXN	**1**	**xx**	**Port for C2C RxD negative line**
PINRXN	**1**	**xx**	**Pin for C2C RxD negative line**
PORTCLK	**1**	**xx**	**Port for C2C clock line**
PINCLK	**1**	**xx**	**Pin for C2C clock line**
MOD	**1**	**xx**	**Single read mode** Definition of C2C transmission mode **Bit(s)** / **Description** **7:0** / **reserved - should be set to 0**

Name	Length	Fmt	Description
CH	**1**	**xx**	**Channel** Channel This field selects HSSL-Channel 0..3
ADR	**4**	**xxxxxxxx**	**Slave address** Memory start address of slave to be read from.
DSIZE	**1**	**xx**	**Read size** Number of byte to read.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	08	xx	xx	01	...

Parameters

Reponse single read command

6	+0
(1..n*)	**RXDAT**

Details

Reponse single read command

Name	Length	Fmt	Description
RXDAT	**1**	**xx**	**Received bytes** Return bytes of read operation.

.

20 - C2C single write

This command performs C2C single write operation into connected controllers memory.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	08	20	01	...

Parameters

Parameter set for C2C single write

5	+0	+1	+2	+3	+4	+5
(1*)	**PORTTXP**	**PINTXP**	**PORTTXN**	**PINTXN**	**PORTRXP**	**PINRXP**

... (1*)	+6 PORTRXN	+7 PINRXN	+8 PORTCLK	+9 PINCLK	+10 MOD	+11 CH

... (1*)	+12 ADR

Data for C2C single write.

21 (1..n*)	+0 TXDAT

Details

Parameter set for C2C single write

Name	Length	Fmt	Description
PORTTXP	**1**	**xx**	**Port for C2C TxD positive line** 0xFF: PINTXP is interpreted as C2C handle ID for faster access
PINTXP	**1**	**xx**	**Pin for C2C TxD positive line** If PORTTXP was 0xFF, this is used as C2C handle ID. These IDs are in the same order the C2C initialization where done using CMD_00, starting with 0. maximum: 2 entries
PORTTXN	**1**	**xx**	**Port for C2C TxD negative line**
PINTXN	**1**	**xx**	**Pin for C2C TxD negative line**
PORTRXP	**1**	**xx**	**Port for C2C RxD positive line**
PINRXP	**1**	**xx**	**Pin for C2C RxD positive line**
PORTRXN	**1**	**xx**	**Port for C2C RxD negative line**
PINRXN	**1**	**xx**	**Pin for C2C RxD negative line**
PORTCLK	**1**	**xx**	**Port for C2C clock line**
PINCLK	**1**	**xx**	**Pin for C2C clock line**
MOD	**1**	**xx**	**Single read mode** Definition of C2C transmission mode **Bit(s)** / **Description** **7:0** / **reserved - should be set to 0**
CH	**1**	**xx**	**Channel** Channel This field selects HSSL-Channel 0..3
ADR	**4**	**xxxxxxxx**	**Slave address** Start address for write operation.

Data for C2C single write.

Name	Length	Fmt	Description
TXDAT	**1**	**xx**	**Data for C2C write transmission**

Name	Length	Fmt	Description
			Data byte to send via C2C. Repeatable up to totally 4 bytes.

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	08	xx	xx	01

.

30 - C2C read slave uC ID

This command reads identifier register from connected controller.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	08	30	01	...

Parameters

Parameter set for C2C slave uC ID read.

5	+0	+1	+2	+3	+4	+5
(1*)	**PORTTXP**	**PINTXP**	**PORTTXN**	**PINTXN**	**PORTRXP**	**PINRXP**

...	+6	+7	+8	+9	+10	+11
(1*)	**PORTRXN**	**PINRXN**	**PORTCLK**	**PINCLK**	**MOD**	**CH**

Details

Parameter set for C2C slave uC ID read.

Name	Length	Fmt	Description
PORTTXP	**1**	**xx**	**Port for C2C TxD positive line** 0xFF: PINTXP is interpreted as C2C handle ID for faster access
PINTXP	**1**	**xx**	**Pin for C2C TxD positive line** If PORTTXP was 0xFF, this is used as C2C handle ID. These IDs are in the same order the C2C initialization where done using CMD_00, starting with 0. maximum: 2 entries
PORTTXN	**1**	**xx**	**Port for C2C TxD negative line**
PINTXN	**1**	**xx**	**Pin for C2C TxD negative line**
PORTRXP	**1**	**xx**	**Port for C2C RxD positive line**
PINRXP	**1**	**xx**	**Pin for C2C RxD positive line**
PORTRXN	**1**	**xx**	**Port for C2C RxD negative line**

Name	Length	Fmt	Description
PINRXN	**1**	**xx**	**Pin for C2C RxD negative line**
PORTCLK	**1**	**xx**	**Port for C2C clock line**
PINCLK	**1**	**xx**	**Pin for C2C clock line**
MOD	**1**	**xx**	**Single read mode** Definition of C2C transmission mode **Bit(s)** | **Description** **7:0** | **reserved - should be set to 0**
CH	**1**	**xx**	**Channel** Channel This field selects HSSL-Channel 0..3

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	08	xx	xx	01	...

Parameters

Reponse on C2C slave uC ID read command

6	+0
(1*)	**SLID**

Details

Reponse on C2C slave uC ID read command

Name	Length	Fmt	Description
SLID	**4**	**xxxxxxxx**	**Slave ID register** Slave controller ID register content.

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**

Code	Description
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**Data size invalid**
0x02	**Mode invalid**
0x03	**Invalid timeout time**
0x04	**Invalid baudrate**
0xFF	**HAL busy**
0xFE	**Generic HAL error**
0xFD	**HAL not compatible to current CPU**
0xFC	**Invalid params**
0xFB	**HAL module not implemented**
0xFA	**HAL function not implemented**
0xF9	**HAL timeout**
0xAC	**Various transmission error happened**
0xAB	**uC Module Initialization failed**
0xAA	**Requested action only allowed for master**
0xA9	**Baudrate together with reference clock frequency invalid**

Code	Description
0xA8	**Acknowledge wrong**
0xA7	**Transaction ID Error**
0xA6	**CRC Error**
0xA5	**Speed change of Master TX / Slave RX failed**
0xA4	**Speed change of Master RX / Slave TX failed**
0xA3	**Setup of C2C link to partner failed**
0xA2	**Size invalid**
0xA1	**Address alignment invalid**
0xA0	**Channel invalid**

TSW_09 - Manual setup of PWM output

Test step description

This test step is used to configure GTM output timer cells and output multiplexers for ports and MSC manually.

Note 1: All used controller pins have to be initialized with TSW_04.	
Note 2: Neither ressource management is done within this test step nor any information is shared with TSW_05. So ALL resource usage must be planned and supervised by user!	

The following output timer resources (GTM) are available:

Table 1. Timer resources

Value for Timer module	IFX/JDP
0x00	TOM0
0x01	TOM1
0x02	TOM2
0x0x	TOMx
0x10	ATOM0
0x11	ATOM1
0x12	ATOM2
0x1x	ATOMx

Note

Existing timer resources are depending on used device!

Test step commands

.

01 - Setup timer cells

This command allows to configure and modify the output timer cells.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	09	01	01	...

Parameters

Parameter set for PWM output timer configuration request

5	+0	+2	+4
(1..n*)	**PERIOD**	**DUTY**	**TIMRES**

Details

Parameter set for PWM output timer configuration request

Name	Length	Fmt	Description
PERIOD	**2**	**xxxx**	**Period and resolution PWM output** Format: RVVV where R=Resolution, V=Value of period.
DUTY	**2**	**xxxx**	**Duty cycle of PWM output** Resolution of duty cycle is 0.01%. The output is active "low" (i.e. DutyCycle = 10% means 10% low and 90% high). Setting to 0% or 100% will force the pin to a static low/high level. Example: 0x2710 means 100% duty cycle.
TIMRES	**2**	**xxxx**	**Output timer resource** Timer resource for generating output signal. Example 1: TIMRES = 0x0105: --> Select TOM1_5. Example 2: TIMRES = 0x1805: --> Select ATOM8_5. **Bit(s)** / **Description** **15:8** — **Timer module x ((A)TOMx_y) (see table 'Timer resources')** **7:0** — **Timer channel y (TOMx_y=0..15, ATOMx_y=0..7)**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	09	xx	xx	01	...

Parameters

Response set for PWM output timer configuration request

6	+0
(1..n*)	**TIMRES**

Details

Response set for PWM output timer configuration request

Name	Length	Fmt	Description
TIMRES	**2**	**xxxx**	**Used output timer resource.** Timer module that generates the signal. **Bit(s)** / **Description** **15:8** — **Timer module (see table 'Timer resources')**

Name	Length	Fmt	Description
			Bit(s) / **Description** **7:0** / **Timer channel (TOMx_0..15, ATOMx_0..7)**

.

02 - Setup GTM to port multiplexer

This command allows to configure and modify the GTM to port multiplexer.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	09	02	01	...

Parameters

Parameter set for PWM output port multiplexer configuration request

5	+0	+2
(1..n*)	**PORTNUM**	**PORTVAL**

Details

Parameter set for PWM output port multiplexer configuration request

Name	Length	Fmt	Description
PORTNUM	**2**	**xxxx**	**GTM to port output number** Relating number n of output TOUTn (relating to register TOUTSELx[SELy]). Example: PORTNUM = 0x27: --> select TOUT39. Please refer to controller specific manual, chapter "GTM to Port Control Registers" for details!
PORTVAL	**1**	**xx**	**GTM to port register value** Relating value v to be written into register TOUTSELx[SELy] (relating to TOUTn). Example: PORTVAL = 0x2: --> results in: TOUTSELx[SELy] = 2. Please refer to controller specific manual, chapter "GTM to Port Control Registers" for details and valid values!

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	09	xx	xx	01

.

03 - Setup GTM to MSC multiplexer

This command allows to configure and modify the GTM to MSC multiplexer.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	09	03	01	...

Parameters

Parameter set for PWM output MSC multiplexer configuration request

5	+0	+2	+3	+5	+6
(1..n*)	**MSCNUMO**	**MSCVALO**	**MSCNUMI**	**MSCSELI**	**MSCVALI**

Details

Parameter set for PWM output MSC multiplexer configuration request

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>MSCNUMO</td><td>2</td><td>xxxx</td><td>GTM to MSC output number

Relating number n,m of output MSCSETnCONx[SELm] (relating SETn SIGNALm out).

Note
Numbering of SET multiplexers in docu (1..n) differs from register (0..(n-1)). Here numbering from (0..(n-1)) must be used!

Example: MSCNUMO = 0x0105: --> select SET2 SIGNAL5 out.

Please refer to controller specific manual, chapter "GTM to MSC Control Registers" for details and valid values!

<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>15:8</td><td>number (n-1) of multiplexer SETn</td></tr>
<tr><td>7:0</td><td>number m of signal SIGNALm of SETn</td></tr>
</table>
</td></tr>
<tr><td>MSCVALO</td><td>1</td><td>xx</td><td>GTM to MSC register value

Relating value v to be written into register MSCSETnCONx[SELm] (relating SETn SIGNALm out).

Example: MSCVALO = 0x2: --> results in: MSCSETnCONx[SELm] = 2.</td></tr>
</table>

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>Please refer to controller specific manual, chapter "GTM to MSC Control Registers" for details and valid values!</td></tr>
<tr><td>MSCNUMI</td><td>2</td><td>xxxx</td><td>MSC input control number

Relating number n,m of input MSCnINmCON (relating MSCn INPUTm connection).

Example: MSCNUMI = 0x0101: --> select MSC1 INPUT HIGH connection.

Please refer to controller specific manual, chapter "GTM to MSC Control Registers" for details and valid values!

<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>15:8</td><td>MSC number n of multiplexer MSCnINmCON</td></tr>
<tr><td>7:2</td><td>Not used. Set to 0.</td></tr>
<tr><td>1:0</td><td>3: MSC HIGHextended input MSCnINHECON.</td></tr>
<tr><td>1:0</td><td>2: MSC LOWextended input MSCnINLECON.</td></tr>
<tr><td>1:0</td><td>1: MSC HIGH input MSCnINHCON.</td></tr>
<tr><td>1:0</td><td>0: MSC LOW input MSCnINLCON.</td></tr>
</table></td></tr>
<tr><td>MSCSELI</td><td>1</td><td>xx</td><td>MSC input selection number

Relating number s of input MSCnINmCON[SELs].

Example: MSCSELI = 0x5: --> select MSCnINmCON[SEL5] connection.

Please refer to controller specific manual, chapter "GTM to MSC Control Registers" for details and valid values!</td></tr>
<tr><td>MSCVALI</td><td>1</td><td>xx</td><td>MSC input control value

Relating value v to be written into register MSCnINmCON[SELs]) (relating MSCn INm INPUTs connection).

Example: MSCVALI = 0x2: --> results in: MSCSETnCONm[SELs] = 2.

Please refer to controller specific manual, chapter "GTM to MSC Control Registers" for details and valid values!</td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	09	xx	xx	01

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**

TSW_0A - Clock System

Test step description

Date: 06.12.2012

The TSW-0A provides interfacing commandos to configure the clock system of the CPU.

Test step commands

.

00 - Configure EXTCLOCK signal

This command configures the EXTCLOCK signal, thus it configures the prescaler to provide the desired frequency. The appropriate pin has to be configured by TSW-04. It is possible to activate more than one EXTCLOCK pin.

Notes for IFX

1) The frequencies for EXTCLK0 and EXTCLK1 pins are generated using different mechanisms:

a) EXTCLK0 pins use a fractional divider with the following formula:

```
f_out = f_FPI * n / 2048
```

(with f_FPI = 100MHz, n = divider value). This means that only frequencies with multiples of (1/2048) * 100MHz = ca. 48,8 kHz can be set exactly. All other frequencies are rounded to the next 48 kHz.

b) EXTCLK1 pins use a normal divider with the following formula:

```
f_out = f_SPB / (DIV1 + 1)
```

(with f_SPB = 100MHz). This, too, means that not all frequencies can be generated exactly. The actually achieved frequency is returned in parameter RCLK.

2) All pins that belong to an EXTCLK group (EXTCLK0 or EXTCLK1) share the same frequency.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	0A	00	01	...

Parameters

5	+0	+1	+2
(1..n*)	**PORT**	**PIN**	**FCLK**

Details

Name	Length	Fmt	Description
PORT	**1**	**xx**	**Clock output port**
PIN	**1**	**xx**	**Clock output pin**
FCLK	**2**	**xxxx**	**Clock frequency** Output frequency [Hz] of the external clock output pin. Format: RVVV where R=Resolution, V=Value

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	0A	xx	xx	01	...

Parameters

Response data

6	+0
(1..n*)	**RCLK**

Details

Response data

Name	Length	Fmt	Description
RCLK	**4**	**xxxxxxxx**	**Real clock frequency [Hz]**

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)

Code	Description
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0xFF	**HAL busy**
0xFE	**Generic HAL error**
0xFD	**HAL not compatible to current CPU**
0xFC	**Invalid params**
0xFB	**HAL module not implemented**
0xFA	**HAL function not implemented**

TSW_10 - Read/write CPU registers directly by address

Test step description

Date: 14.09.2012

This TSW module provides to the CPU peripherals' registers by directly reading or writing a from/to a specific address. No read/writeaccess is granted to flash areas.

Test step commands

.

00 - Read from address

This command provides a direct read access to the peripheral registers by giving an address. The value which is read from the address is then binary-masked (AND) by the parameter **MASK** and writen to the transmit PDU.

The access is limited to peripheral address space. No read access to flash is possible.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	10	00	01	...

Parameters

Parameter for command 0x00

5	+0	+1	+5
(1..n*)	**MOD**	**ADR**	**MASK**

Details

Parameter for command 0x00

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>MOD</td><td>1</td><td>xx</td><td>Access Mode for Read
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:4</td><td>Protection:
0_H: No Protection
1_H: EndInit Protection
2_H: SaftyEndInit Protection</td></tr>
<tr><td>3:0</td><td>Width:
1_H: 8Bit</td></tr>
</table>
</td></tr>
</table>

Name	Length	Fmt	Description	
			Bit(s)	**Description**
				2_H: 16Bit **4_H:** 32Bit
ADR	**4**	**xxxxxxxx**	**Read Address**	
MASK	**4**	**xxxxxxxx**	**Read Mask**	

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	10	xx	xx	01	...

Parameters

Response data

6	+0
(1..n*)	**VAL**

Details

Response data

Name	Length	Fmt	Description
VAL	**4**	**xxxxxxxx**	**Read value**

.

80 - Write to address

This command provides direct write access to the (endinit) peripheral registers by giving an address. The value to be written is binary-masked (AND) with the parameter **MASK**. So only bits of the write value where the corresponding bitposition in the MASK contains a '1', are modified in the register to be written.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	10	80	01	...

Parameters

Parameter for command 0x80

5	+0	+1	+5	+9
(1..n*)	**MOD**	**ADR**	**MASK**	**VAL**

Details

Parameter for command 0x80

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>MOD</td><td>1</td><td>xx</td><td>Access Mode for Write
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:4</td><td>Protection:
0_H: No Protection
1_H: EndInit Protection
2_H: SaftyEndInit Protection</td></tr>
<tr><td>3:0</td><td>Width:
1_H: 8Bit
2_H: 16Bit
4_H: 32Bit</td></tr>
</table></td></tr>
<tr><td>ADR</td><td>4</td><td>xxxxxxxx</td><td>Write Address</td></tr>
<tr><td>MASK</td><td>4</td><td>xxxxxxxx</td><td>Write Mask</td></tr>
<tr><td>VAL</td><td>4</td><td>xxxxxxxx</td><td>Write Value</td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	10	xx	xx	01	...

Parameters

Response data

6	+0
(1..n*)	**VAL**

Details

Response data

Name	Length	Fmt	Description
VAL	**4**	**xxxxxxxx**	**Read value after write**

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)

Code	Description
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**TSW: Address is not allowed to read or write.**
0x02	**TSW: Protection is not supported.**
0x03	**TSW: Address fits not to given width type.**
0x04	**TSW: Width type not supported.**

TSW_18 - CAN-Communication

Test step description

This test step is used to setup the CAN controllers on the CPU, transmit and receive data and check for CAN errors.

Table 1. Explaination of node configuration for all commands using parameter "NODE" (availability of modules depending from device).

Parameter "NODE"	CAN Module	CAN Node
0x00..0x03	CAN Module 0	Node 0..3
0x10..0x13	CAN Module 1	Node 0..3
0x20..0x23	CAN Module 2	Node 0..3
0x30..0x33	CAN Module 3	Node 0..3
0x40..0x41	CANR Module 0	Node 0..1
0x50..0x51	CANR Module 1	Node 0..1
0x60..0x61	CANR Module 2	Node 0..1
0x70..0x71	CANR Module 3	Node 0..1

Note for CAN FD operation: Use first command 0x1 "Initialize CAN controller". After this switch node to CAN FD with command 0x20 "Change CAN (FD) node". Use then command 0x2 "Set Baudrate" for setting arbitration baud rate. After this set data baud rate with command 0x21 "Set CAN FD Data Baud Rate". Use then command 0x3 "Setup Message Buffer", command 0x4 "Transmit CAN (FD) Data" and command 0x5 "Receive CAN (FD) Data" as for normal CAN operation.

Test step commands

.

01 - Initialize CAN controller

This command has to be used only once to initially enable the CAN modules on the CPU.

Note: Receive input configuration (RXSEL in node port control register NPCRx) as required for IFX controllers must be configured with command 0x10 "Change receive input" for values different then 0.

Note: All nodes are initalized as CAN nodes. To switch nodes to CAN FD or back use command 0x20 "Change CAN (FD) node" after command 0x1.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	18	01	01

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	18	xx	xx	01

.

02 - Set Baudrate

This command is used to configure the CAN module to a specific baudrate.

Note: With CAN FD the baud rate set with this command is for arbitration phase only. Use command 0x21 "Set CAN FD Data Baud Rate" after command 0x2 to set data baud rate for CAN FD.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	18	02	01	...

Parameters

Parameter set for baudrate setting

5	+0	+1
(1..n*)	**NODE**	**BR**

Details

Parameter set for baudrate setting

Name	Length	Fmt	Description
NODE	**1**	**xx**	**CAN node number** Index of the CAN node for which the baudrate shall be set (0 ... N-1)
BR	**2**	**xxxx**	**Baudrate** CAN Baudrate setting Format: RVVV where R=Resolution, V=Value (R=3: kBit/s. For example: 0x31F4 = 500kBit/s, 0x33E8 = 1 MBit/s) IFX Dev5: Only 125,250,500kBit/s and 1MBit/s supported!

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	18	xx	xx	01

.

03 - Setup Message Buffer

This command is used to configure message buffers for transmission or reception of data.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	18	03	01	...

Parameters

Parameter set for SetupMessageBuffer request

5	+0	+1	+2	+3
(1..n*)	**NODE**	**MSGBUF**	**CFG**	**MSGID**

Details

Parameter set for SetupMessageBuffer request

Name	Length	Fmt	Description
NODE	**1**	**xx**	**CAN node number** Index of the CAN node to prepare the message buffer for (0 ... N-1)
MSGBUF	**1**	**xx**	**Message Buffer** CAN Message Buffer number (0,1,...) that shall be configured **Note:** Max. allowed buffers: 23 buffers (for each RX and TX) for device 5. **Note:** Max. allowed buffers: 85 buffers for CAN module and 42 buffers for CANR module for all other devices.
CFG	**1**	**xx**	**Message Buffer Configuration** Configuration of the Message Buffer **0_H:** buffer for CAN reception **1_H:** buffer for CAN transmission
MSGID	**4**	**xxxxxxxx**	**CAN Message ID (standard or extended format)** This parameter contains the message ID for the transmit message to be sent or the receive filter ID. The uppermost bit selects the length of the ID, the lower bits have to be set to the ID value. 0x8XXX XXXX: ID is 29 bit long (extended CAN ID) 0x0XXX XXXX: ID is 11 bit long (standard CAN ID)

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	18	xx	xx	01

.

04 - Transmit CAN (FD) Data

This command is used to write data into a message buffer (which has to be previously configured to TX direction by subcommand "SetupMessageBuffer") and trigger the transmission.By setting the parameter TXTO (transmit timeout) to a value != 0, the TSW checks whether the message has been transmitted sucessfully within this time. Otherwise a timeout is reported and transmission will be aborted.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	18	04	01	...

Parameters

Parameter set for CAN Transmit Data

5	+0	+1	+2	+4
(1..n*)	**NODE**	**MSGBUF**	**TXTO**	**TXDAT**

Details

Parameter set for CAN Transmit Data

Name	Length	Fmt	Description
NODE	**1**	**xx**	**CAN node index** Index of the CAN node (0 ... N-1)
MSGBUF	**1**	**xx**	**Message Buffer** CAN Message Buffer number (0 ... Number of message buffers -1) to be written and transmitted. Must be previously configured by command "Setup Message Buffer".
TXTO	**2**	**xxxx**	**Transmit Timeout** = 0: TSW will not check for transmission success. The CAN controller stops transmission after the TX errorcounter reaches a threshold value of 0x80.> 0: TSW will check for transmission success within the time set by TXTO and reports an error if the message failed to transmit within the given time.Format: RVVV where R=Resolution, V=Value (e.g. R=0: Seconds, R=D: Milliseconds)Maximum allowed timeout value is 40 seconds
TXDAT	**64**	**xx....**	**Transmit Data** Number of data may be from 0 up to a maximum of 8 (64 with CAN FD) bytes. Others are ignored

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	18	xx	xx	01	...

Parameters

Response set of CAN Transmit Data request

6 (1..n*)	+0 **TXSTAT1**	+1 **TXSTAT2**	+2 **TECNT**	+3 **RECNT**

Details

Response set of CAN Transmit Data request

Name	Length	Fmt	Description
TXSTAT1	1	xx	**Transmit Status 1** **Bit(s)** / **Description** **7** — **Error flag** **0_B:** No error occured **1_B:** An error occured during transmission attempt **6** — **Previous Message Status** **0_B:** No message is pending **1_B:** A message was pending and overwritten by new value **5** — **Bus-Off Status** **0_B:** No bus-off occured **1_B:** Last transmission resulted in a bus-off state **4:0** — **not used**
TXSTAT2	1	xx	**Transmit Status 2 (IFX Dev3/IFX Dev4 only, IFX Dev5/JDP/FSL: not used)** **Bit(s)** / **Description** **7:2** — **not used** **1** — **Last Error Increment Flag** **0_B:** The last error caused error counter to increment by 1 **1_B:** The last error caused error counter to increment by 8 **0** — **Last Error Transfer Direction Flag** **0_B:** Last error occured while the CAN node was receiver, RX-errorcounter has been incremented **1_B:** Last error occured while the CAN node was transmitter, TX-errorcounter has been incremented

Name	Length	Fmt	Description
TECNT	**1**	**xx**	**Transmit Error Counter**
RECNT	**1**	**xx**	**Receive Error Counter**

.

05 - Receive CAN (FD) Data

This command is used to read data from a message buffer.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	18	05	01	...

Parameters

Parameter set for CAN Receive Data

5	+0	+1	+2
(1..n*)	**NODE**	**MSGBUF**	**RXTO**

Details

Parameter set for CAN Receive Data

Name	Length	Fmt	Description
NODE	**1**	**xx**	**CAN node index** Index of the CAN node (0 .. NumberOfCanNodes - 1)
MSGBUF	**1**	**xx**	**Message Buffer** Index of the buffer to read data from. The buffer itself has to be configured previously by using command 0x04.
RXTO	**2**	**xxxx**	**Receive Timeout** = 0: TSW checks if the message buffer contains new data. An error will be reported when no data is present in the message buffer.> 0: TSW checks if message has been received within the time set by RXTO. An error will be reported if no message has been received within the given time.Format: RVVV where R=Resolution, V=Value (e.g. R=0: Seconds, R=D: Milliseconds)Maximum allowed timeout value is 40 seconds

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	18	xx	xx	01	...

Parameters

Response set of CAN Receive Data request

6	+0	+1	+2	+3	+4
(1..n*)	RXSTAT1	RXSTAT2	TECNT	RECNT	RXDAT

Details

Response set of CAN Receive Data request

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>RXSTAT1</td><td>1</td><td>xx</td><td>Receive Status 1
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7</td><td>Message Valid Flag
0_B: RX data is valid
1_B: No data received</td></tr>
<tr><td>6</td><td>Message Overwritten Flag
0_B: No overwriten occured
1_B: Receive message buffer has been overwritten with a new message</td></tr>
<tr><td>5:0</td><td>not used</td></tr>
</table></td></tr>
<tr><td>RXSTAT2</td><td>1</td><td>xx</td><td>Receive Status 2 (IFX Dev3/IFX Dev4 only, IFX Dev5/JDP/FSL: not used)
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:2</td><td>not used</td></tr>
<tr><td>1</td><td>Last Error Increment Flag
0_B: The last error caused error counter to increment by 1
1_B: The last error caused error counter to increment by 8</td></tr>
<tr><td>0</td><td>Last Error Transfer Direction Flag
0_B: Last error occured while the CAN node was receiver, RX-errorcounter has been incremented
1_B: Last error occured while the CAN node was transmitter, TX-errorcounter has been incremented</td></tr>
</table></td></tr>
<tr><td>TECNT</td><td>1</td><td>xx</td><td>Transmit Error Counter</td></tr>
<tr><td>RECNT</td><td>1</td><td>xx</td><td>Receive Error Counter</td></tr>
<tr><td>RXDAT</td><td>64</td><td>xx....</td><td>Received data bytes</td></tr>
</table>

Name	Length	Fmt	Description
			These bytes contain the received message data. The number of bytes depends on the real message length and may be from 0 up to 8 (64 with CAN FD) bytes.

.

10 - Change receive input

This command is used to change receive input register (RXSEL in node port control register NPCRx) for a given node.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	18	10	01	...

Parameters

Parameter set for change receive input register request

5	+0	+1
(1..n*)	**NODE**	**RXSEL**

Details

Parameter set for change receive input register request

Name	Length	Fmt	Description
NODE	**1**	**xx**	**CAN node number** Index of the CAN node (0 ... N-1) to change receive input register
RXSEL	**1**	**xx**	**RXSEL value** Receive select value (RXSEL)

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	18	xx	xx	01

.

20 - Change CAN (FD) node

This command is used to change a node between CAN and CAN FD operation.

Note: Command 0x1 "Initialize CAN controller" must be called at least once before.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	18	20	01	...

Parameters

Parameter set for change CAN (FD) node request

5	+0	+1
(1..n*)	**NODE**	**MODE**

Details

Parameter set for change CAN (FD) node request

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>NODE</td><td>1</td><td>xx</td><td>CAN (FD) node number
Index of the CAN(FD) node (0 ... N-1) to change node to CAN or CAN FD</td></tr>
<tr><td>MODE</td><td>1</td><td>xx</td><td>CAN (FD) mode
<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>7:1</td><td>not used, should be set to 0</td></tr><tr><td>0</td><td>CAN (FD) mode
0_B: CAN mode
1_B: CAN FD mode</td></tr></table></td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	18	xx	xx	01

.

21 - Set CAN FD Data Baud Rate

This command is used to configure the CAN FD node to a specific data baud rate.

Note: Command 0x2 "Set Baudrate" must be called before to set baud rate for arbitration phase.

Note: The baud rate configured for the CAN FD data phase here must be higher or equal to the baud rate configured for the arbitration phase via command 0x2 "Set Baudrate".

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	18	21	01	...

Parameters

Parameter set for CAN Fd data baud rate setting

5	+0	+1	+3
(1..n*)	**NODE**	**FDBR**	**FDTDC**

Details

Parameter set for CAN Fd data baud rate setting

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>NODE</td><td>1</td><td>xx</td><td>CAN FD node number
Index of the CAN FD node for which the baud rate shall be set (0 ... N-1)</td></tr>
<tr><td>FDBR</td><td>2</td><td>xxxx</td><td>CAN FD baud rate for data phase
CAN FD baud rate setting for data phase

Format: RVVV where R=Resolution, V=Value (R=3: kBit/s. For example: 0x31F4 = 500kBit/s, 0x33E8 = 1 MBit/s)

Device 1, 2: Only 1, 2, 3.333, 4 and 5 MBd supported!

Device 3, 4, 5: Only 1, 2, 3.333, 4, 5, 6.667, 8 and 10 MBd supported!</td></tr>
<tr><td>FDTDC</td><td>2</td><td>xxxx</td><td>CAN FD transceiver delay compensation
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>15</td><td>Source of transceiver delay compensation data for IFX Device 5

0_B: Fixed transceiver delay compensation data will be used.
1_B: Transceiver delay compensation given by following data is used.</td></tr>
<tr><td>14</td><td>switch on/ off transceiver delay compensation.

0_B: Transceiver delay compensation off
1_B: Transceiver delay compensation on</td></tr>
<tr><td>13:7</td><td>not used, should be 0</td></tr>
<tr><td>6:0</td><td>CAN FD transceiver delay compensation value. Only used if Bit 15 and 14 are set.
Transceiver delay compensation value is used to adjust the secondary sample point inside the bit time. Reason: Avoid errors during comparison of transmitted data with the received data from the local CAN transceiver. See CAN FD specification and controller manuals for details!

Format: CAN clock periods.

Device 5: 80 MHz</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>15</td><td>Source of transceiver delay compensation data for IFX all other devices

0_B: Fixed transceiver delay compensation data will be used.
1_B: Transceiver delay compensation given by following data is used.</td></tr>
<tr><td>14</td><td>switch on/ off transceiver delay compensation.

0_B: Transceiver delay compensation off
1_B: Transceiver delay compensation on</td></tr>
<tr><td>13:4</td><td>not used, should be 0</td></tr>
<tr><td>3:0</td><td>CAN FD transceiver delay compensation value. Only used if Bit 15 and 14 are set.
Transceiver delay compensation value is used to adjust the secondary sample point inside the bit time. Reason: Avoid errors during comparison of transmitted data with the received data from the local CAN transceiver. See CAN FD specification and controller manuals for details!

Format: CAN clock periods.

Device 1, 2: 20 MHz

Device 3, 4: 40 MHz</td></tr>
</table>
</td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	18	xx	xx	01

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**

Code	Description
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**TSW: Command not supported**
0xFF	**HAL busy**
0xFE	**Generic HAL error**
0xFD	**HAL not compatible to current CPU**
0xFC	**Invalid params**
0xFB	**HAL module not implemented**
0xFA	**HAL function not implemented**
0xF9	**HAL timeout**
0xEF	**HAL-CAN: Baudrate not supported**

Code	Description
0xEE	**HAL-CAN: Requested CAN module not found**
0xED	**HAL-CAN: Requested subfunction not supported**
0xEC	**HAL-CAN: RX/TX buffer full, communication busy**
0xEB	**HAL-CAN: Unsupported CAN input clock**

TSW_1A - FlexRay bus- and echotest

Test step description

This test step is used to test the FlexRay busdriver, FlexRay controller and to run a communication test.

Test step commands

.

01 - Flexray porttest

Note

Previously configure each used flexray TXD, TXEN and RXD pin as General Purpose I/O pin using TSW_04 BEFORE running porttest! Port test for channel A and B must be performed separately.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	1A	01	01	...

Parameters

Parameter set for Flexray port test request

5	+0	+1	+2	+3	+4	+5
(1*)	**RXDPO**	**RXDPI**	**TXDPO**	**TXDPI**	**TXENPO**	**TXENPI**

Details

Parameter set for Flexray port test request

Name	Length	Fmt	Description
RXDPO	**1**	**xx**	**Port of signal RXD**
RXDPI	**1**	**xx**	**Pin of signal RXD**
TXDPO	**1**	**xx**	**Port of signal TXD**
TXDPI	**1**	**xx**	**Pin of signal TXD**
TXENPO	**1**	**xx**	**Port of signal TXEN**
TXENPI	**1**	**xx**	**Pin of signal TXEN**

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	1A	xx	xx	01

.

02 - Get transceiver state

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	1A	02	01	...

Parameters

Parameter set for get transceiver state request

5 (1*)	+0	+1	+2	+3	+4	+5
	ENPO	**ENPI**	**STBNPO**	**STBNPI**	**ERRNPO**	**ERRNPI**

... (1*)	+6
	TCTYPE

Details

Parameter set for get transceiver state request

Name	Length	Fmt	Description
ENPO	**1**	**xx**	**Port of signal EN** connected to transceiver
ENPI	**1**	**xx**	**Pin of signal EN** connected to transceiver
STBNPO	**1**	**xx**	**Port of signal STBN** connected to transceiver
STBNPI	**1**	**xx**	**Pin of signal STBN** connected to transceiver
ERRNPO	**1**	**xx**	**Port of signal ERRN** connected to transceiver
ERRNPI	**1**	**xx**	**Pin of signal ERRN** connected to transceiver
TCTYPE	**1**	**xx**	**Transceiver type** **Bit(s)** – **Description** **0** – **1: Transceiver is TJA1080A, TJA1081A** **1** – **1: Transceiver is TJA1081B**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	1A	xx	xx	01	...

Parameters

Response set for Flexray get transceiver state request

6	+0
(1*)	**TCSTAT**

Details

Response set for Flexray get transceiver state request

Name	Length	Fmt	Description
TCSTAT	**2**	**xxxx**	**Transceiver state** of status register TJA1080/81. For a detailed description refer to manual of transceiver.

Bit(s)	Description
15:13	**not used, always set to 0**
12	**TRXD collision, TJA1081B: not used, always reset**
11	**Star locked mode, TJA1081B: BGE feedback**
10	**Undervoltage on VIO pin**
9	**Undervoltage on VCC pin**
8	**Undervoltage on VBAT pin**
7	**TXEN-BGE clamped**
6	**Medium temperature**
5	**High temperature**
4	**Bus error**
3	**Power-on status**
2	**Node config, TJA1081B: not used, always set**
1	**Remote wakeup**
0	**Local wakeup**

.

03 - Echo test

The purpose of the echo test is to startup a flexray cluster together with a 2nd ECU which is running in echo server mode.

The bus is setup to have 2 static and 2 dynamic slots which contains the transmission of testpattern and one dynamic slot which contains the remote datablocks 0–2 (RDAT0–2) and will be sent at the end of the test. All local Eray register which are responded by this command, are read after receiption of frame ID5. The following tables shows details about the frame usage and the initialisation values of the Eray/CC controller.

Table 1. TSW_1A Dataframes for echotest

Frame ID	Type	Length [Byte]	Sent by	Message buffer (MBS) in local ECU
1	static	254	echo server	1
2	static	254	ECU	0
3	dynamic	254	echo server	3

Frame ID	Type	Length [Byte]	Sent by	Message buffer (MBS) in local ECU
4	dynamic	254	ECU	2
5	dynamic	12	echo server	4

Table 2. TSW_1A Eray initialisation values (IFX)

Register	Value		Register	Value
PRTC1	0x7C1F 011A		GTUC6	0x001f 00cd
PRTC2	0x0612 0505		GTUC7	0x0002 00c8
MRC	0x0004 8002		GTUC8	0x0064 0004
MHDC	0x0064 007F		GTUC9	0x0000 0201
GTUC1	0x0001 3880		GTUC10	0x03e8 03e8
GTUC2	0x0003 03e8		GTUC11	0x0000 0000
GTUC3	0x0202 1717		SUCC2	0x0F02 713E
GTUC4	0x033e 031f		SUCC3	0x0000 0011
GTUC5	0x3801 0101		NEMC	0x0000 000C

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	1A	03	01	...

Parameters

Parameter set for Flexray Echo test request

5	+0	+1	+2
(1*)	**ENCHAN**	**RISA**	**RISB**

Details

Parameter set for Flexray Echo test request

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>ENCHAN</td><td>1</td><td>xx</td><td>Enable Flexray module/channel
Only one channel (channel A or channel B) is supported at a time!
<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>7:4</td><td>Flexray module number.</td></tr><tr><td>3:2</td><td>not used, should be set to 0</td></tr><tr><td>1</td><td>1: Enable Flexray channel B</td></tr><tr><td>0</td><td>1: Enable Flexray channel A</td></tr></table></td></tr>
<tr><td>RISA</td><td>1</td><td>xx</td><td>Flexray RX pin register RISA configuration

IFX: Configuration as required on IFX controller. Please refer to "Customer Registers" -> CUST1 -> RISA/RISB in chapter "E-Ray Kernel Registers" of the manual for allowed values.</td></tr>
</table>

Name	Length	Fmt	Description
RISB	**1**	**xx**	**Flexray RX pin register RISB configuration** IFX: Configuration as required on IFX controller. Please refer to "Customer Registers" -> CUST1 -> RISA/RISB in chapter "E-Ray Kernel Registers" of the manual for allowed values.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	1A	xx	xx	01	...

Parameters

Response set for Flexray echo test request

6 (1*)	+0	+4	+8	+12	+14	+18
	LDAT0	**LDAT1**	**LDAT2**	**LDAT3**	**LDAT4**	**LDAT5**

... (1*)	+22	+26	+30	+34	+38	+42
	LDAT6	**LDAT7**	**LDAT8**	**LDAT9**	**LDAT10**	**LDAT11**

... (1*)	+46	+50	+54	+58	+62	+66
	LDAT12	**LDAT13**	**LDAT14**	**RDAT0**	**RDAT1**	**RDAT2**

... (1*)	+70
	LDAT15

Details

Response set for Flexray echo test request

Name	Length	Fmt	Description
LDAT0	**4**	**xxxxxxxx**	**Register EIR (IFX) or CHIERFR (Bit 15:0) (JDP)** (local flexray register)
LDAT1	**4**	**xxxxxxxx**	**Register SIR (IFX) or PIFR0 (Bit 15:0) (JDP)** (local flexray register)
LDAT2	**4**	**xxxxxxxx**	**Register ACS (IFX) or PIFR1 (Bit 15:0) (JDP)** (local flexray register)
LDAT3	**2**	**xxxx**	**Register CHIPID (IFX) or MVR (JDP)** (local flexray register)
LDAT4	**4**	**xxxxxxxx**	**Register CLC (IFX) or MCR (Bit 15:0) (JDP)** (local flexray register)
LDAT5	**4**	**xxxxxxxx**	**Maximum rate correction value RCV (IFX) or RTCORVR (Bit 15:0)(JDP)** (local flexray register)
LDAT6	**4**	**xxxxxxxx**	**Offset correction value OCV (IFX) or OFCORVR (Bit 15:0) (JDP)** (local flexray register)
LDAT7	**4**	**xxxxxxxx**	**Register SFS (IFX) or CBSERCR (Bit 31:16) and CASERCR (Bit 15:0)) (JDP)** (local flexray register)

Name	Length	Fmt	Description
LDAT8	**4**	**xxxxxxxx**	**Register CCSV (IFX) or PSR0 (Bit 15:0) (JDP)** (local flexray register)
LDAT9	**4**	**xxxxxxxx**	**Register CCEV (IFX) or PSR1 (Bit 15:0) (JDP)** (local flexray register)
LDAT10	**4**	**xxxxxxxx**	**Register SWNIT (IFX) or PSR2 (Bit 15:0) (JDP)** (local flexray register)
LDAT11	**4**	**xxxxxxxx**	**Message buffer 0 status (IFX) or PSR3 (Bit 15:0) (JDP)** (local flexray register)
LDAT12	**4**	**xxxxxxxx**	**Message buffer 1 status (IFX) or FrameId 2 status (odd-cycle (Bit 31:16), even-cycle (Bit 15:0)) (JDP)** (local flexray register)
LDAT13	**4**	**xxxxxxxx**	**Message buffer 2 status (IFX) or FrameId 3 status (odd-cycle (Bit 31:16), even-cycle (Bit 15:0)) (JDP)** (local flexray register)
LDAT14	**4**	**xxxxxxxx**	**Message buffer 3 status (IFX) or FrameId 4 status (odd-cycle (Bit 31:16), even-cycle (Bit 15:0)) (JDP)** (local flexray register)
RDAT0	**4**	**xxxxxxxx**	**Remote datablock 0** from echo server NOTE: Value is 0xFFFFFFFF if no connection to the echo server could be established

Bit(s)	Description
31	**not used, always set to 0**
30:26	**CCSV.RCA Remaining coldstart attempts**
25:20	**CCSV.POCS Protocol operation control status**
19	**SFS.RCLR Rate correction limit reached**
18	**SFS.MRCS Missing rate correction signal**
17	**SFS.OCLR Offset correction limit reached**
16	**SFS.MOCS Missing offset correction signal**
15:12	**SFS.VSBO Valid sync frames on channel B; odd cycles**
11:8	**SFS.VSBE Valid sync frames on channel B; even cycles**
7:4	**SFS.VSAO Valid sync frames on channel A; odd cycles**
3:0	**SFS.VSAE Valid sync frames on channel A; even cycles**

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>RDAT1</td><td>4</td><td>xxxxxxxx</td><td>Remote datablock 1
from echo server

NOTE: Value is 0xFFFFFFFF if no connection to the echo server could be established

<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>31:17</td><td>not used, always set to 0</td></tr>
<tr><td>16</td><td>ACS.SBVB Slot boundary violation on channel B</td></tr>
<tr><td>15</td><td>ACS.CIB Communication indicator channel B</td></tr>
<tr><td>14</td><td>ACS.CEDB Content error detected on channel B</td></tr>
<tr><td>13</td><td>ACS.SEDB Syntax error detected on channel B</td></tr>
<tr><td>12</td><td>ACS.VFRB Valid frames received on channel B</td></tr>
<tr><td>11:9</td><td>not used, always set to 0</td></tr>
<tr><td>8</td><td>ACS.SBVA Slot boundary violation on channel A</td></tr>
<tr><td>7</td><td>ACS.CIA Communication indicator channel A</td></tr>
<tr><td>6</td><td>ACS.CEDA Content error detected on channel A</td></tr>
<tr><td>5</td><td>ACS.SEDA Syntax error detected on channel A</td></tr>
<tr><td>4</td><td>ACS.VFRA Valid frames received on channel A</td></tr>
<tr><td>3:0</td><td>CCEV.CCFC Clock correction failed counter</td></tr>
</table></td></tr>
<tr><td>RDAT2</td><td>4</td><td>xxxxxxxx</td><td>Remote datablock 2
from echo server

NOTE: Value is 0xFFFFFFFF if no connection to the echo server could be established

<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>31:12</td><td>Offset correction value</td></tr>
<tr><td>11:0</td><td>Maximum rate correction value</td></tr>
</table></td></tr>
<tr><td>LDAT15</td><td>4</td><td>xxxxxxxx</td><td>Message buffer 4 status (IFX) or FrameId 5 status (odd-cycle (Bit 31:16), even-cycle (Bit 15:0)) (JDP)
Value reflects the status of the frame ID5 immediatly after its reception.</td></tr>
</table>

.

04 - Start echo server (only IFX)

With this command the echo server is started, which is the counterpart for the echo test. After starting this function, the software does not return and therefore there will be not response from ECU if no errors occur during communication initialization

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	1A	04	01	...

Parameters

Parameter set for Flexray echo server request

5	+0	+1	+2
(1*)	**ENCHAN**	**RISA**	**RISB**

Details

Parameter set for Flexray echo server request

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>ENCHAN</td><td>1</td><td>xx</td><td>Enable Flexray channel
Only one channel (channel A or channel B) is supported at a time!
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>0</td><td>1: Enable Flexray channel A</td></tr>
<tr><td>1</td><td>1: Enable Flexray channel B</td></tr>
</table></td></tr>
<tr><td>RISA</td><td>1</td><td>xx</td><td>Flexray RX pin register RISA configuration
Pin configuration as required on IFX controller. Please refer to "Customer Registers" -> CUST1 -> RISA/RISB in chapter "E-Ray Kernel Registers" of the manual for allowed values.</td></tr>
<tr><td>RISB</td><td>1</td><td>xx</td><td>Flexray RX pin register RISB configuration
Pin configuration as required on IFX controller. Please refer to "Customer Registers" -> CUST1 -> RISA/RISB in chapter "E-Ray Kernel Registers" of the manual for allowed values.</td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	1A	xx	xx	01

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0xEF	**Lock of ERAY_PLL failed**
0xEE	**General error of testmode initialization**
0xED	**Port test failed**
0xEA	**Testmode not supported**
0xE9	**Port not found**
0xE8	**Pin not found**
0xE7	**Transceiver type unknown**

Code	Description
0xE6	Failed to set CONFIG mode
0xE5	Failed to set READY mode
0xE4	Failed to set ACTIVATE_COLDSTART mode
0xE3	Failed to set RUN mode
0xE2	Timeout during cluster startup
0xE1	Failed to set back to CONFIG mode
0xE0	Flexray channel not supported

TSW_1B - Ethernet bus test

Test step description

This test step is used to test the Ethernet PHY, Ethernet MAC and to run a communication test.

Note: All used pins have to be initialized with TSW_04.

Note: All PHY relating configuration must be done with command "PHY register write" before start of communication test. No PHY configuration will be done from test step except remote loopback configuration for command "Communication echo partner initialization".

Test step commands

.

01 - Ethernet controller initialization

This function allows to configure the Ethernet controller

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	1B	01	01	...

Parameters

Parameter set for Ethernet controller initialization request

5 (1*)	+0	+1	+2
	CTRLIDX	**MIIMODE**	**GPCTL**

Details

Parameter set for Ethernet controller initialization request

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>CTRLIDX</td><td>1</td><td>xx</td><td>Ethernet controller index
Number of Ethernet controller (MAC)</td></tr>
<tr><td>MIIMODE</td><td>1</td><td>xx</td><td>MII/ RMII
Type of media independent interface: MII or RMII
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:1</td><td>not used, should be set to 0</td></tr>
<tr><td>0</td><td>1: Media independent interface is RMII</td></tr>
<tr><td>0</td><td>0: Media independent interface is MII</td></tr>
</table></td></tr>
<tr><td>GPCTL</td><td>4</td><td>xxxxxxxx</td><td>Ethernet input pin register GPCTL configuration</td></tr>
</table>

Name	Length	Fmt	Description
			IFX: Configuration of register ETH_GPCTL as required on IFX controllers. Please refer to "Additional Register" -> ETH_GPCTL -> ALTIy in chapter "Ethernet MAC Module-Implementation Related Registers" of the manual for details!

Bit(s)	Description
31:22	**not used, always set to 0**
21:20	**ALTI10: Alternate Input Select 10**
19:18	**ALTI9: Alternate Input Select 9**
17:16	**ALTI8: Alternate Input Select 8**
15:14	**ALTI7: Alternate Input Select 7**
13:12	**ALTI6: Alternate Input Select 6**
11:10	**ALTI5: Alternate Input Select 5**
9:8	**ALTI4: Alternate Input Select 4**
7:6	**ALTI3: Alternate Input Select 3**
5:4	**ALTI2: Alternate Input Select 2**
3:2	**ALTI1: Alternate Input Select 1**
1:0	**ALTI0: Alternate Input Select 0**

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	1B	xx	xx	01

.

02 - PHY register read

Note: Required previously successful command execution: Ethernet controller initialization

This function allows to read registers from PHY

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	1B	02	01	...

Parameters

Parameter set for Ethernet read PHY register request

5	+0	+1
(1..n*)	**PHYADR**	**PHYREGN**

Details

Parameter set for Ethernet read PHY register request

Name	Length	Fmt	Description
PHYADR	1	xx	PHY address Bit(s) / Description 7:5 — not used, always set to 0 4:0 — PHY address
PHYREGN	1	xx	PHY register number

Response

1	2	3	4	5	6
SVC	TSW	ESVC	ETSW	VER	...
C0	1B	xx	xx	01	...

Parameters

Response set for Ethernet read PHY register request

6 (1..n*)	+0
	PHYREGV

Details

Response set for Ethernet read PHY register request

Name	Length	Fmt	Description
PHYREGV	2	xxxx	PHY register value

.

03 - PHY register write

Note: Required previously successful command execution: Ethernet controller initialization

This function allows to write registers to PHY

Command

1	2	3	4	5
SVC	TSW	CMD	VER	...
80	1B	03	01	...

Parameters

Parameter set for Ethernet write PHY register request

5 (1..n*)	+0	+1	+2	+4
	PHYADR	PHYREGN	PHYREGV	PHYREGM

Details

Parameter set for Ethernet write PHY register request

Name	Length	Fmt	Description
PHYADR	1	xx	PHY address

Name	Length	Fmt	Description
			Bit(s) / **Description** **7:5** — **not used, always set to 0** **4:0** — **PHY address**
PHYREGN	**1**	**xx**	**PHY register number**
PHYREGV	**2**	**xxxx**	**PHY register value**
PHYREGM	**2**	**xxxx**	**PHY register mask** Set bit position to 1, if bit should be changed.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	1B	xx	xx	01	...

Parameters

Response set for Ethernet write PHY register request

6 (1..n*)	+0
	PHYREGV

Details

Response set for Ethernet write PHY register request

Name	Length	Fmt	Description
PHYREGV	**2**	**xxxx**	**PHY register value**

.

10 - Communication master initialization

Note: Required previously successful command execution: Ethernet controller initialization

This function allows to configure communication master

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	1B	10	01	...

Parameters

Parameter set for Ethernet configure communication master request

5 (1*)	+0	+1	+2	+3	+4	+5
	PHYTYPE	**PHYADR**	**MACADR5**	**MACADR4**	**MACADR3**	**MACADR2**

... (1*)	+6	+7	+8	+9	+10
	MACADR1	**MACADR0**	**MIIMODE**	**LOOPBCK**	**DATA**

Details

Parameter set for Ethernet configure communication master request

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>PHYTYPE</td><td>1</td><td>xx</td><td>Ethernet PHY type
Type of Ethernet PHY:
<table>
<tr><th>Legal values</th><th>Media type</th><th>PHY type</th></tr>
<tr><td>0x00</td><td>100Base-T1</td><td>TJA1100</td></tr>
<tr><td>0x10</td><td>100Base-TX</td><td>DP83822</td></tr>
</table></td></tr>
<tr><td>PHYADR</td><td>1</td><td>xx</td><td>PHY address
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:5</td><td>not used, always set to 0</td></tr>
<tr><td>4:0</td><td>PHY address</td></tr>
</table></td></tr>
<tr><td>MACADR5</td><td>1</td><td>xx</td><td>MAC address byte 5
Byte 5 of physical address of ethernet controller (MAC address). Will be used for source MAC address in ethernet frame.</td></tr>
<tr><td>MACADR4</td><td>1</td><td>xx</td><td>MAC address byte 4
Byte 4 of physical address of ethernet controller (MAC address). Will be used for source MAC address in ethernet frame.</td></tr>
<tr><td>MACADR3</td><td>1</td><td>xx</td><td>MAC address byte 3
Byte 3 of physical address of ethernet controller (MAC address). Will be used for source MAC address in ethernet frame.</td></tr>
<tr><td>MACADR2</td><td>1</td><td>xx</td><td>MAC address byte 2
Byte 2 of physical address of ethernet controller (MAC address). Will be used for source MAC address in ethernet frame.</td></tr>
<tr><td>MACADR1</td><td>1</td><td>xx</td><td>MAC address byte 1
Byte 1 of physical address of ethernet controller (MAC address). Will be used for source MAC address in ethernet frame.</td></tr>
<tr><td>MACADR0</td><td>1</td><td>xx</td><td>MAC address byte 0
Byte 0 of physical address of ethernet controller (MAC address). Will be used for source MAC address in ethernet frame.</td></tr>
<tr><td>MIIMODE</td><td>1</td><td>xx</td><td>MII/ RMII
Type of media independent interface: MII or RMII
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:1</td><td>not used, should be set to 0</td></tr>
<tr><td>0</td><td>1: Media independent interface is RMII</td></tr>
<tr><td>0</td><td>0: Media independent interface is MII</td></tr>
</table></td></tr>
<tr><td>LOOPBCK</td><td>1</td><td>xx</td><td>PHY or MAC loopback mode</td></tr>
</table>

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>Type of loopback mode: normal operation, MAC internal loopback or PHY loopback.

Note: For normal operation configure PHY accordingly using command "PHY register write" first and choose "Normal operation" here.

Note: For PHY loopback configure PHY accordingly using command "PHY register write" first and choose "PHY: Local PHY loopback" here.

<table><tr><th>Legal values</th><th>Description</th></tr><tr><td>0x00</td><td>Normal operation.</td></tr><tr><td>0x10</td><td>MAC: Internal MAC loopback.</td></tr><tr><td>0x20</td><td>PHY: Local PHY loopback.</td></tr></table></td></tr>
<tr><td>DATA</td><td>4</td><td>xxxxxxxx</td><td>Transmit data
This data will be used to create two ethernet frames with 256 Bytes data each.

First frame: All data will be sent in given byte order.

Second frame: All data will be sent in inverse byte order.</td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	1B	xx	xx	01

.

11 - Communication master start and compare

Note: Required previously successful command execution: Communication master initalization

This function allows to start communication master and compare data

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	1B	11	01	...

Parameters

Parameter set for Ethernet start and compare communication master request

5	+0
(1*)	**NLOOP**

Details

Parameter set for Ethernet start and compare communication master request

Name	Length	Fmt	Description
NLOOP	**2**	**xxxx**	**Number of execution loops**

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	1B	xx	xx	01

.

12 - Communication master get number of compare errors

Note: Required previously successful command execution: Communication master start and compare

This function allows to get number of compare errors from master

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	1B	12	01

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	1B	xx	xx	01	...

Parameters

Response set for Ethernet get number of compare errors request

6	+0
(1..n*)	**CMPERR**

Details

Response set for Ethernet get number of compare errors request

Name	Length	Fmt	Description
CMPERR	**1**	**xx**	**Compare errors**

.

18 - Ethernet controller stop

Note: Required previously successful command execution: Communication master initalization

This function allows to stop the ethernet controller

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	1B	18	01

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	1B	xx	xx	01

.

20 - Communication echo partner initialization

Note: Required previously successful command execution: Ethernet controller initialization

This function allows to configure communication echo partner

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	1B	20	01	...

Parameters

Parameter set for Ethernet configure and start communication echo partner request

5	+0	+1
(1*)	**PHYTYPE**	**PHYADR**

Details

Parameter set for Ethernet configure and start communication echo partner request

Name	Length	Fmt	Description
PHYTYPE	**1**	**xx**	**Ethernet PHY type** Type of Ethernet PHY: <table><tr><th>Legal values</th><th>Media type</th><th>PHY type</th></tr><tr><td>0x00</td><td>100Base-T1</td><td>TJA1100</td></tr></table>

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td><table><tr><th>Legal values</th><th>Media type</th><th>PHY type</th></tr><tr><td>0x10</td><td>100Base-TX</td><td>DP83822</td></tr></table></td></tr>
<tr><td>PHYADR</td><td>1</td><td>xx</td><td>PHY address<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>7:5</td><td>not used, always set to 0</td></tr><tr><td>4:0</td><td>PHY address</td></tr></table></td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	1B	xx	xx	01

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)

Code	Description
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0xEF	**HAL: Ethernet Controller not supported**
0xEE	**HAL: Type of PHY not supported**
0xED	**HAL: Link timeout**
0xEC	**HAL: Communication timeout**
0xEB	**HAL: Compare failed**
0xEA	**HAL: Reset MAC timeout**

TSW_1E - LM71 Temperature meas device

Test step description

Purpose: This TSW provides testing of LM71 temperature measuring device.

Important: before using, initializing of some other hardware resources is required.

Required test steps

tsw_04: configuration of port/pin signals for serial communication to LM71.

This test step also allows continuous monitoring of board temperature.

Test step commands

.

00 - LM71 operating/communication with device

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	1E	00	01	...

Parameters

Port configuration and mode

5 (1*)	+0	+1	+2	+3	+4	+5
	PORTCS	**PINCS**	**PORTIO**	**PINIO**	**PORTCK**	**PINCK**

... (1*)	+6
	MODE

Details

Port configuration and mode
Port and Pin parameters to serial communication lines where LM71 is connected

Name	Length	Fmt	Description
PORTCS	**1**	**xx**	**Port number of CS signal** Port number of CS (Chip select) signal
PINCS	**1**	**xx**	**Pin number of CS signal** Pin number within PORTCS of CS (Chip select) signal
PORTIO	**1**	**xx**	**Port number of IO signal** Port number of IO (Input/Output data) signal
PINIO	**1**	**xx**	**Pin number of IO signal**

Name	Length	Fmt	Description
			Pin number within PORTIO of IO (Input/Output data) signal
PORTCK	**1**	**xx**	**Port number of CK signal** Port number of CK (Clock) signal
PINCK	**1**	**xx**	**Pin number of CK signal** Pin number within PORTCK of CK (Clock) signal
MODE	**1**	**xx**	**Communication mode of LM71** **Note:** 300ms will have to pass for a conversion to complete before LM71 actually transmits temperature data (mode 02). Bit(s) / Description: 3 – **Start conversion (prerequisite for mode 02)** 2 – **Read actual temperature (conversion has started first, see note)** 1 – **Start conversion and read actual temperature value (measurement takes 300ms)** 0 – **Read ID of LM71 (typ.=800F)**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	1E	xx	xx	01	...

Parameters

Response of requested value

6	+0	+2
(0*)	**RVAL**	**RRAW**

Details

Response of requested value

Name	Length	Fmt	Description
RVAL	**2**	**xxxx**	**Device ID / Temp.val [°C]** If operation was successful, see table below for responded values... Bit(s) / Description: 3 – **return value is void** 2 – **Actual temperature value in [°C]** 1 – **Actual temperature value in [°C]** 0 – **ID of LM71 (typ.=800F)**

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>RRAW</td><td>2</td><td>xxxx</td><td>Temp.raw value [x0.03125°C]
If operation was successful, see table below for responded values...
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>3</td><td>return value is void</td></tr>
<tr><td>2</td><td>Actual raw temperature value in [x0.03125°C]</td></tr>
<tr><td>1</td><td>Actual raw temperature value in [x0.03125°C]</td></tr>
<tr><td>0</td><td>ID of LM71 (typ.=800F)</td></tr>
</table>
</td></tr>
</table>

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)

Code	Description
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**LM71 identification code not valid**
0xFF	**HAL busy**
0xFE	**Generic HAL error**
0xFD	**HAL not compatible to current CPU**
0xFC	**Invalid params**
0xFB	**HAL module not implemented**
0xFA	**HAL function not implemented**
0xEF	**GPIO port not found**
0xEE	**GPIO pin not found**
0xED	**GPIO function not supported**
0xEC	**GPIO port.pin / module not connectable**

TSW_20 - CY32x test

Test step description

This test step is used to test the CY32x.

The TSW_20 supports CY326, CY327 and CY328.

Test step commands

.

01 - RST5 test

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	20	01	01	...

Parameters

Parameter set for CY32x RST5 test request

5	+0	+1	+2	+3	+4	+5
(1*)	**PORTMT**	**PINMT**	**PORTMR**	**PINMR**	**PORTCK**	**PINCK**

...	+6	+7	+8	+9
(1*)	**PORTSL**	**PINSL**	**RST5PO**	**RST5PIN**

Details

Parameter set for CY32x RST5 test request

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal**

Name	Length	Fmt	Description
			Port number of SLSO (Slave Select) signal. If MSB bit is set, emulated SLSO is used with given port (only for IFX and JDP).
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal. If MSB bit is set, emulated SLSO is used with given pin (only for IFX and JDP)
RST5PO	**1**	**xx**	**RST5 input port number** Port number of RST5 input signal.
RST5PIN	**1**	**xx**	**RST5 input pin number** Pin number of RST5 input signal.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	20	xx	xx	01	...

Parameters

Response set for CY32x RST5 test request

6	+0
(1*)	**RST5TIM**

Details

Response set for CY32x RST5 test request

Name	Length	Fmt	Description
RST5TIM	**2**	**xxxx**	**elapsed RST5 active (low) time** Format: RVVV where R=Resolution, V=Value (e.g. R=A: 1 µs). Maximum time is limited to 409.5s.

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**

Code	Description
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**TSW: SPI communication error**
0x02	**TSW: Unknown device, not CY32x**
0x03	**TSW: Port not found**
0x04	**TSW: RST5 test failed => pin already LOW at test start**
0x05	**TSW: RST5 test timeout => pin toggling takes longer than 100ms**

TSW_22 - CJ72x test

Test step description

This test step is used to test the software reset of CJ72x.

The TSW_22 supports CJ721 and CJ722.

Command 01 generates a U-chip internal reset initiated by the uC´s software ("software reset").

As RST5 is an internal signal without a specific pin the uC gets the information of VDD5-undervoltage via SDO pin. As long as VDD5 is in undervoltage state the pin SDO is pulled to '0'.

Test step commands

.

01 - RST5 test

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	22	01	01	...

Parameters

Port/pin parameter set for CJ72x

5	+0	+1	+2	+3	+4	+5
(1..n*)	**PO_FCLP**	**PI_FCLP**	**PO_FCLN**	**PI_FCLN**	**PO_EN**	**PI_EN**

...	+6	+7	+8	+9	+10	+11
(1..n*)	**PO_SOP**	**PI_SOP**	**PO_SON**	**PI_SON**	**PO_SDI**	**PI_SDI**

Parameter set for RST input signal

17	+0	+1
(1..n*)	**PO_RST**	**PI_RST**

Details

Port/pin parameter set for CJ72x

Name	Length	Fmt	Description
PO_FCLP	**1**	**xx**	**Port number of FCLP signal** Port number of FCLP (Clock positiv) signal
PI_FCLP	**1**	**xx**	**Pin number of FCLP signal** Pin number of FCLP (Clock positiv) signal
PO_FCLN	**1**	**xx**	**Port number of FCLN signal**

Name	Length	Fmt	Description
			Port number of FCLN (Clock negativ) signal. Set 0xFF if single ended is used.
PI_FCLN	**1**	**xx**	**Pin number of FCLN signal** Pin number of FCLN (Clock negativ) signal. Set 0xFF if single ended is used.
PO_EN	**1**	**xx**	**Port number of EN signal** Port number of EN (Enable) signal
PI_EN	**1**	**xx**	**Pin number of EN signal** Pin number of EN (Enable) signal
PO_SOP	**1**	**xx**	**Port number of SOP signal** Port number of SOP (Serial Output Positiv) signal.
PI_SOP	**1**	**xx**	**Pin number of SOP signal** Pin number of SOP (Serial Output Positiv) signal.
PO_SON	**1**	**xx**	**Port number of SON signal** Port number of SON (Serial Output Negativ) signal. Set 0xFF is single ended is used.
PI_SON	**1**	**xx**	**Pin number of SON signal** Pin number of SON (Serial Output Negativ) signal. Set 0xFF is single ended is used.
PO_SDI	**1**	**xx**	**Port number of SDI signal** Port number of SDI (Serial Data In) signal
PI_SDI	**1**	**xx**	**Pin number of SDI signal** Pin number of SDI (Serial Data In) signal

Parameter set for RST input signal

Name	Length	Fmt	Description
PO_RST	**1**	**xx**	**Port number of RST signal** Set 0xFF if MSC-SDI signal as RST signal used.
PI_RST	**1**	**xx**	**Pin number of RST signal** Set 0xFF if MSC-SDI signal as RST signal used.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	22	xx	xx	01	...

Parameters

Response for CJ72x SW RST test request

6	+0
(1*)	**RSTTIM**

Details

Response for CJ72x SW RST test request

Name	Length	Fmt	Description
RSTTIM	**2**	**xxxx**	**Elapsed RST active (low) time**

Name	Length	Fmt	Description
			Format: RVVV where R=Resolution, V=Value (e.g. R=A: 1 µs).

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**

Code	Description
0x01	TSW: MSC communication error
0x02	TSW: Unknown device, not CJ721 or CJ722
0x03	TSW: Port not found
0x04	TSW: RST test failed => pin already LOW at test start
0x05	TSW: RST test timeout => pin toggling takes longer than 200µs
0x06	TSW: RST5 test failed => CJ72x not detected an undervoltage reset

TSW_63 - AST

Test step description

Date: 19.05.2014

Created: 18.01.2017 09:12.

This TSW provides functions to start the logical and memory tests of the AST. Single parts of the tests may be activated or disabled by parameter of the TSW command. The implementation and usage of this TSW depends on the ECU family it is used for (see following explanations).

Notes for MDG1 AST

For MDG1 this TSW is part of the AST V3 CPU tests and may only be used as part of the sequence control interpreter (SCI). The file must be included into a combi file which has to be located into flash. Downloading to RAM like other teststeps is not possible.

Test step commands

.

01 - Run AST memory tests

TSW_63 contains several single tests and is designed to be executed in a loop. The teststep responses 0x01 as long as not all test are finished and 0x00 if all tests are finished. Therefore the loop exit condition should be 0x00. In case that the memory tests detect an error, 0xFF is responded followed by the test ID and error specific data.

The TSW is called without parameters to run all flash memory tests. By adding the test ID to the parameter, any test can be deactivated.

Table 1. TSW_63 AST V3 MDG1 Memory tests

ID	AST Int.ID	TEST
0x00	0x00	uC internal flash test involves PFlash and DFlash test.
0x01	0x01	Test of all uC RAM's (eg: program cache, data cache, GTM, CAN, ...).
0x02	0x02	walking 0 test (only if 0x01 is not disabled).
0x03	0x03	walking 1 test (only if 0x01 is not disabled).
0x04	0x04	Test of overlay (ED) RAM.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	63	01	01	...

Parameters

Parameter run memory test

5 (0..n*)	+0
	ID

Details

Parameter run memory test

Name	Length	Fmt	Description
ID	**1**	**xx**	**Disable test ID** Disable test ID Optional parameter to disabled a test.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	63	xx	xx	01	...

Parameters

General response data for errors at AST memory test

6 (1*)	+0
	DATA

Details

General response data for errors at AST memory test

Table 2. Test Response Data Content for MDG1

Index	Content	Description
0	RC	Return Code of AST Test Function (0xFF, 0xFE, 0xFD, 0x01, 0x00)
1-12	ID	ASCII("*Memory Test")
13-240	Result	see AST Specification

Table 3. Recommended caller reaction for MDG1

Return code	Store result	Call test again
0xFF	yes	yes
0xFE	yes	no
0xFD	yes	no
0x01	no	yes
0x00	no	no

Name	Length	Fmt	Description
DATA	**1**	**xx..**	**Test Response Data** MDG1: 241 Bytes Response Data

.

02 - Run AST logic tests

The TSW is called without parameters to run all logic tests. By adding the test ID to the parameter, any test can be deactivated.

Table 4. TSW_63 AST V3 MDG1 Logic tests

ID	AST Int.ID	TEST
0x00	0x00	CAN test: Test of all the CAN nodes in loopback mode at 1MB baud for communication across node.
0x01	0x01	GTM test: Test of GTM sub modules TOM, TIM, ATOM, BRC, PSM, CMP are test for its functionality as per the spec.
0x02	0x02	DMA test: Various available DMA transfers modes are tested (single request, multiple request, modulo feature, shadow mode).
0x03	0x03	SARADC test: Test of all possible GTM trigger for every SAR ADC channel and check for conversion.
0x04	0x04	DSADC test: Test of all possible GTM trigger for every DS ADC channel and check for conversion.
0x05	0x05	ASC test: ASC internal Loopback test.
0x06	0x06	Overlay RAM test: Overlay RAM functionality test.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	63	02	01	...

Parameters

Parameter test run logic test

5	+0
(0..n*)	**ID**

Details

Parameter test run logic test

Name	Length	Fmt	Description
ID	**1**	**xx**	**Disable test ID** Disable test ID Optional parameter to disabled a test.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	63	xx	xx	01	...

Parameters

Command response data

6	+0
(1*)	**DATA**

Details

Command response data
Command response data.

Table 5. Test Response Data Content for MDG1

Index	Content	Description
0	RC	Return Code of AST Test Function (0xFF, 0xFE, 0xFD, 0x01, 0x00)
1-12	ID	ASCII("**Logic Test")
13-240	Result	see AST Specification

Table 6. Recommended caller reaction for MDG1

Return code	Store result	Call test again
0xFF	yes	yes
0xFE	yes	no
0xFD	yes	no
0x01	no	yes
0x00	no	no

Name	Length	Fmt	Description
DATA	**1**	**xx..**	**Test Response Data** MDG1: 241 Bytes Response Data

.

80 - Get additional AST library infos

Function to get additional informations from AST library (only MDG1, not implemented)

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	63	80	01

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	63	xx	xx	01	...

Parameters

Response of Get AST library infos

6	+0
(0..n*)	**INFOS**

Details

Response of Get AST library infos

Name	Length	Fmt	Description
INFOS	*****	**xx....**	**Infos [0...240]**

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**Tests not finished; call again (only E84)**
0x02	**Actual test provided data (only E84)**
0x03	**Invoke Reset in SC (only E84)**
0xFF	**Actual test reported an error (only E84)**

TSW_65 - AST Report Analysis

Test step description

This TSW is used to readout and analyse the log memory contents of AST. The log memory is written into FLASH.

Created 18.01.2017 09:12.
Structure of an Flash Logger Entry

Each entry has a 9 byte header containing a time stamp, an address field that usually contains the address of the currently executed Sequence Control command or alternatively the Error ID, as well as the Log Entry type and the length of the used data. After the header the data is stored; the data field contains 247 Byte. Table 1. shows the flash layout of Flash Log Area and Table 2. gives an overview of the structure of a Flash Logging Module entry.

Table 1. Entry structure

<table>
<tr><th>Byte(s)</th><th>Description</th></tr>
<tr><td>0:3</td><td>Time stamp (big endian)</td></tr>
<tr><td>4:6</td><td>Sequence Control Address (big endian) or Error ID</td></tr>
<tr><td>7</td><td>Entry Type
<table>
<tr><th>Type</th><th>Name</th><th>When is written</th></tr>
<tr><td>0x00</td><td>Startup</td><td>The SCI boots the ECU</td></tr>
<tr><td>0x20</td><td>Regular</td><td>Instead SCI sends a CAN Message</td></tr>
<tr><td>0x40</td><td>Error</td><td>The SCI encounters a General Error</td></tr>
<tr><td>0x60</td><td>End</td><td>The last command in the SC was executed</td></tr>
<tr><td>0x80</td><td>Info</td><td>Like a Regular Entry but not as important</td></tr>
</table>
</td></tr>
<tr><td>8</td><td>Number of used bytes at Data</td></tr>
<tr><td>9:256</td><td>Data</td></tr>
</table>

Following sections describe the content of Data for the different Type´s:

Startup Entry (Type 0x00)

Each time the SCI starts after any cpu reset, the SCI stores information about the ECU. The content of the Startup Entry is listed in the Table 3. In the table the offset is noted relative to the start of the entry.

Table 3. Content of Startup Log Entry

Part	Pos. [HEX]	Size[Byte]	Explanation
Version (VE)	0x9	1	SCI Logger Startup Version version: 11 — remark: MDG1 Version 1
CPU ID	0xA	4	CPU ident register BIG ENDIAN SCU.CHIPID(IFX) / SIUL2_PD.MIDR1(jdp)
SCI Info	0xE	32	Sci Information cs, content:
	0xE	4	Cpu_Manufacturer_u32* — Defined in xx\tsw_common\machine.h 0x01000000 = IFX 0x00000002 = JDP 0x00000003 = FSL
	0x12	16	ASCII SCI build information e.g. "sci_jdp_dev4"
	0x22	12	ASCII Version String in Format: V<x>.<y>.<z>
	0x2E	10	InfoblockPointer_ps->TTNr_u8 of BootCtrl
BootCtrl Info	0x38	35 (33)	ASCII String in Format: CUBAS BOOTCTRL_V<x>.<y>.<z>/<manufacturer> <device> z.B. "CUBAS BOOTCTRL_V02.00.00/JDP DEV4C2"
TSW Combi Version	0x5B	24	ASCII Combi build information e.g. "combi_a00_jdp_dev4"
	0x73	12	ASCII Version string in format: V<x>.<y>.<z>
TSW Combi HAL Version	0x7F	24	ASCII Combi HAL build information e.g. "hal_jdp_dev4"
	0x97	12	ASCII Version string in format: V<x>.<y>.<z>
ESB Mode Byte (EM)	0xA3	1	ESB Mode Byte overtaken once at start of ECU
Config() Setting	0xA4	10	Setting of command config() in scc code made before Logger Start
	0xA4	1	CANMessageSuppression
	0xA5	1	CANErrorSuppression
	0xA6	1	CANWarningSuppression
	0xA7	1	TimedLoopWarningSuppression
	0xA8	1	TimerWarningSuppression
	0xA9	1	WDTStatus
	0xAA	1	RvvvWarningSuppression
	0xAB	1	LoggerMode
	0xAC	1	MultiCoreV1
	0xAD	1	Endmarker 0xAA
Reset Type	0xAE	40	Reset_ActEnv_st*, size depending on uController
Endmarker	0xD6	1	Endmarker 0xAA
	0xD7	215	

* = Little or Big Endian acc. to uC familiy

Table 4. Software Reset Type IDs (Byte pos.: 0xBE)

Value	Description
	ASW RESETS
0x3000	Mcu Module dummy reset type for Mcu_GetResetReason()
0x3001	this is the first configured reset
0x3002	default reset id of reset trigger
0x3003	Reset id created from rba_Reset comp. Used in case of infoblock handling fault.
	SB RESETS
0x1000	PowerOn Reset Simu
0x1001	HW Reset Simu
0x1002	Reprogramming Reset
0x1003	Reprogramming CB Reset
0x1004	CB Cpu Reset
0x1005	Illegal Path in RB Programming Reset
0x1006	TSW Exit Reset
0x1007	TSW Magic Pattern Reset
0x1008	Trap occured in SB
	HW RESETS
0x0000	PowerOn Reset
0x0001	Warm PowerOn Reset
0x0002	PowerOn Monitoring Modul Reset
0x0003	PowerOn Main Relay Reset

Value	Description
0x0004	ESR0 Line Reset
0x0005	ESR1 Line Reset
0x0006	Watchdog Reset
0x0007	SMU Reset
0x0008	Cerberus System Reset
0x0009	OCDS Reset
0x000A	Cerberus System Reset
0x000B	Cerberus System Reset
0x000C	Cerberus System Reset
0x000D	TP Reset
0x000E	CAN wakeup reset
0x000F	PIN1 wakeup reset
0x0010	PIN2 wakeup reset
0x0011	PIN3 wakeup reset
0x0012	StopCounter wakeup reset
	TRAPS CPU0
0x5000	TCN = 0, TIN = 0: MMU: Virtual Address Fill
0x5001	TCN = 0, TIN = 1: MMU: Virtual Address Protection
0x5002	TCN = 0, TIN = >1
0x5003	TCN = 1, TIN = 0: Unknown
0x5004	TCN = 1, TIN = 1: Privileged Instruction
0x5005	TCN = 1, TIN = 2: Memory Protection: Read Access
0x5006	TCN = 1, TIN = 3: Memory Protection: Write Access
0x5007	TCN = 1, TIN = 4: Memory Protection: Execution Access
0x5008	TCN = 1, TIN = 5: Memory Protection: Peripheral Access
0x5009	TCN = 1, TIN = 6: Memory Protection: Null Address
0x500A	TCN = 1, TIN = 7: Global Register Write Protection
0x500B	TCN = 1, TIN = >7
0x500C	TCN = 2, TIN = : Unknown
0x500D	TCN = 2, TIN = 1: Illegal Opcode
0x500E	TCN = 2, TIN = 2: Unimplemented Opcode
0x500F	TCN = 2, TIN = 3: Invalid Operand Specification
0x5010	TCN = 2, TIN = 4: Data Address Alignment Error
0x5011	TCN = 2, TIN = 5: Invalid Local Memory Address
0x5012	TCN = 2, TIN = >5
0x5013	TCN = 3, TIN = 0: Unknown
0x5014	TCN = 3, TIN = 1: Free Context List Depleted
0x5015	TCN = 3, TIN = 2: Call Depth Overflow
0x5016	TCN = 3, TIN = 3: Call Depth Underflow
0x5017	TCN = 3, TIN = 4: Free Context List Underflow
0x5018	TCN = 3, TIN = 5: Context List Underflow

Value	Description
0x5019	TCN = 3, TIN = 6: Context Type Error
0x501A	TCN = 3, TIN = 7: Nesting Error
0x501B	TCN = 3, TIN = >7
0x501C	TCN = 4, TIN = 0: Unknown
0x501D	TCN = 4, TIN = 1: Program Fetch Synchronous Error
0x501E	TCN = 4, TIN = 2: Data Access Synchronous Error
0x501F	TCN = 4, TIN = 3: Data Access Asynchronous Error
0x5020	TCN = 4, TIN = 4: Coprocessor TRAP_ Asynchronous Error
0x5021	TCN = 4, TIN = 5: Program Memory Integrity Error - no bit set in PIETR
0x5022	TCN = 4, TIN = 6: Data Memory Integrity Error - no bit set in DIETR
0x5023	TCN = 4, TIN = >6
0x5024	TCN = 4, TIN = 5: no bit is set in PIETR
0x5025	TCN = 4, TIN = 5: CPU0, DMI NonSafe: Program Integrity Error Detected
0x5026	TCN = 4, TIN = 5: CPU0, DMI NonSafe: Program Integrity Error Tag Memory
0x5027	TCN = 4, TIN = 5: CPU0, DMI NonSafe: Program Integrity Error Cache Memory
0x5028	TCN = 4, TIN = 5: CPU0, DMI NonSafe: Program Integrity Error Scratchpad Memory
0x5029	TCN = 4, TIN = 5: CPU0, DMI NonSafe: Program Integrity Error Bus Interface
0x502A	TCN = 4, TIN = 5: CPU0, DMI NonSafe: Program Integrity Error Dual Error Detected
0x502B	TCN = 4, TIN = 5: CPU0, DMI NonSafe: Program Integrity Error Safety Protection Error Detected
0x502C	TCN = 4, TIN = 5: CPU0, DMI NonSafe: Program Integrity Error Bus Slave Access Indicator
0x502D	TCN = 4, TIN = 5: CPU0, DMI Safe: Program Integrity Error Detected
0x502E	TCN = 4, TIN = 5: CPU0, DMI Safe: Program Integrity Error Tag Memory
0x502F	TCN = 4, TIN = 5: CPU0, DMI Safe: Program Integrity Error Cache Memory
0x5030	TCN = 4, TIN = 5: CPU0, DMI Safe: Program Integrity Error Scratchpad Memory
0x5031	TCN = 4, TIN = 5: CPU0, DMI Safe: Program Integrity Error Bus Interface
0x5032	TCN = 4, TIN = 5: CPU0, DMI Safe: Program Integrity Error Dual Error Detected
0x5033	TCN = 4, TIN = 5: CPU0, DMI Safe: Program Integrity Error Safety Protection Error Detected
0x5034	TCN = 4, TIN = 5: CPU0, DMI Safe: Program Integrity Error Bus Slave Access Indicator
0x5035	TCN = 4, TIN = 5: CPU1, DMI NonSafe: Program Integrity Error Detected
0x5036	TCN = 4, TIN = 5: CPU1, DMI NonSafe: Program Integrity Error Tag Memory
0x5037	TCN = 4, TIN = 5: CPU1, DMI NonSafe: Program Integrity Error Cache Memory
0x5038	TCN = 4, TIN = 5: CPU1, DMI NonSafe: Program Integrity Error Scratchpad Memory
0x5039	TCN = 4, TIN = 5: CPU1, DMI NonSafe: Program Integrity Error Bus Interface
0x503A	TCN = 4, TIN = 5: CPU1, DMI NonSafe: Program Integrity Error Dual Error Detected

Value	Description
0x503B	TCN = 4, TIN = 5: CPU1, DMI NonSafe: Program Integrity Error Safety Protection Error Detected
0x503C	TCN = 4, TIN = 5: CPU1, DMI NonSafe: Program Integrity Error Bus Slave Access Indicator
0x503D	TCN = 4, TIN = 5: CPU1, DMI Safe: Program Integrity Error Detected
0x503E	TCN = 4, TIN = 5: CPU1, DMI Safe: Program Integrity Error Tag Memory
0x503F	TCN = 4, TIN = 5: CPU1, DMI Safe: Program Integrity Error Cache Memory
0x5040	TCN = 4, TIN = 5: CPU1, DMI Safe: Program Integrity Error Scratchpad Memory
0x5041	TCN = 4, TIN = 5: CPU1, DMI Safe: Program Integrity Error Bus Interface
0x5042	TCN = 4, TIN = 5: CPU1, DMI Safe: Program Integrity Error Dual Error Detected
0x5043	TCN = 4, TIN = 5: CPU1, DMI Safe: Program Integrity Error Safety Protection Error Detected
0x5044	TCN = 4, TIN = 5: CPU1, DMI Safe: Program Integrity Error Bus Slave Access Indicator
0x5045	TCN = 4, TIN = 5: CPU2, DMI NonSafe: Program Integrity Error Detected
0x5046	TCN = 4, TIN = 5: CPU2, DMI NonSafe: Program Integrity Error Tag Memory
0x5047	TCN = 4, TIN = 5: CPU2, DMI NonSafe: Program Integrity Error Cache Memory
0x5048	TCN = 4, TIN = 5: CPU2, DMI NonSafe: Program Integrity Error Scratchpad Memory
0x5049	TCN = 4, TIN = 5: CPU2, DMI NonSafe: Program Integrity Error Bus Interface
0x504A	TCN = 4, TIN = 5: CPU2, DMI NonSafe: Program Integrity Error Dual Error Detected
0x504B	TCN = 4, TIN = 5: CPU2, DMI NonSafe: Program Integrity Error Safety Protection Error Detected
0x504C	TCN = 4, TIN = 5: CPU2, DMI NonSafe: Program Integrity Error Bus Slave Access Indicator
0x504D	TCN = 4, TIN = 5: CPU2, DMI Safe: Program Integrity Error Detected
0x504E	TCN = 4, TIN = 5: CPU2, DMI Safe: Program Integrity Error Tag Memory
0x504F	TCN = 4, TIN = 5: CPU2, DMI Safe: Program Integrity Error Cache Memory
0x5050	TCN = 4, TIN = 5: CPU2, DMI Safe: Program Integrity Error Scratchpad Memory
0x5051	TCN = 4, TIN = 5: CPU2, DMI Safe: Program Integrity Error Bus Interface
0x5052	TCN = 4, TIN = 5: CPU2, DMI Safe: Program Integrity Error Dual Error Detected
0x5053	TCN = 4, TIN = 5: CPU2, DMI Safe: Program Integrity Error Safety Protection Error Detected
0x5054	TCN = 4, TIN = 5: CPU2, DMI Safe: Program Integrity Error Bus Slave Access Indicator
0x5055	TCN = 4, TIN = 5: SDMA MoveEngine0: Program Integrity Error Detected
0x5056	TCN = 4, TIN = 5: SDMA MoveEngine0: Program Integrity Error Tag Memory
0x5057	TCN = 4, TIN = 5: SDMA MoveEngine0: Program Integrity Error Cache Memory
0x5058	TCN = 4, TIN = 5: SDMA MoveEngine0: Program Integrity Error Scratchpad Memory
0x5059	TCN = 4, TIN = 5: SDMA MoveEngine0: Program Integrity Error Bus Interface

Value	Description
0x505A	TCN = 4, TIN = 5: SDMA MoveEngine0: Program Integrity Error Dual Error Detected
0x505B	TCN = 4, TIN = 5: SDMA MoveEngine0: Program Integrity Error Safety Protection Error Detected
0x505C	TCN = 4, TIN = 5: SDMA MoveEngine0: Program Integrity Error Bus Slave Access Indicator
0x505D	TCN = 4, TIN = 5: SDMA MoveEngine1: Program Integrity Error Detected
0x505E	TCN = 4, TIN = 5: SDMA MoveEngine1: Program Integrity Error Tag Memory
0x505F	TCN = 4, TIN = 5: SDMA MoveEngine1: Program Integrity Error Cache Memory
0x5060	TCN = 4, TIN = 5: SDMA MoveEngine1: Program Integrity Error Scratchpad Memory
0x5061	TCN = 4, TIN = 5: SDMA MoveEngine1: Program Integrity Error Bus Interface
0x5062	TCN = 4, TIN = 5: SDMA MoveEngine1: Program Integrity Error Dual Error Detected
0x5063	TCN = 4, TIN = 5: SDMA MoveEngine1: Program Integrity Error Safety Protection Error Detected
0x5064	TCN = 4, TIN = 5: SDMA MoveEngine1: Program Integrity Error Bus Slave Access Indicator
0x5065	TCN = 4, TIN = 5: Cerberus: Program Integrity Error Detected
0x5066	TCN = 4, TIN = 5: Cerberus: Program Integrity Error Tag Memory
0x5067	TCN = 4, TIN = 5: Cerberus: Program Integrity Error Cache Memory
0x5068	TCN = 4, TIN = 5: Cerberus: Program Integrity Error Scratchpad Memory
0x5069	TCN = 4, TIN = 5: Cerberus: Program Integrity Error Bus Interface
0x506A	TCN = 4, TIN = 5: Cerberus: Program Integrity Error Dual Error Detected
0x506B	TCN = 4, TIN = 5: Cerberus: Program Integrity Error Safety Protection Error Detected
0x506C	TCN = 4, TIN = 5: Cerberus: Program Integrity Error Bus Slave Access Indicator
0x506D	TCN = 4, TIN = 5: DMA MoveEngine0: Program Integrity Error Detected
0x506E	TCN = 4, TIN = 5: DMA MoveEngine0: Program Integrity Error Tag Memory
0x506F	TCN = 4, TIN = 5: DMA MoveEngine0: Program Integrity Error Cache Memory
0x5070	TCN = 4, TIN = 5: DMA MoveEngine0: Program Integrity Error Scratchpad Memory
0x5071	TCN = 4, TIN = 5: DMA MoveEngine0: Program Integrity Error Bus Interface
0x5072	TCN = 4, TIN = 5: DMA MoveEngine0: Program Integrity Error Dual Error Detected
0x5073	TCN = 4, TIN = 5: DMA MoveEngine0: Program Integrity Error Safety Protection Error Detected
0x5074	TCN = 4, TIN = 5: DMA MoveEngine0: Program Integrity Error Bus Slave Access Indicator
0x5075	TCN = 4, TIN = 5: DMA MoveEngine1: Program Integrity Error Detected
0x5076	TCN = 4, TIN = 5: DMA MoveEngine1: Program Integrity Error Tag Memory
0x5077	TCN = 4, TIN = 5: DMA MoveEngine1: Program Integrity Error Cache Memory
0x5078	TCN = 4, TIN = 5: DMA MoveEngine1: Program Integrity Error Scratchpad Memory

Value	Description
0x5079	TCN = 4, TIN = 5: DMA MoveEngine1: Program Integrity Error Bus Interface
0x507A	TCN = 4, TIN = 5: DMA MoveEngine1: Program Integrity Error Dual Error Detected
0x507B	TCN = 4, TIN = 5: DMA MoveEngine1: Program Integrity Error Safety Protection Error Detected
0x507C	TCN = 4, TIN = 5: DMA MoveEngine1: Program Integrity Error Bus Slave Access Indicator
0x507D	TCN = 4, TIN = 5: HSSL: Program Integrity Error Detected
0x507E	TCN = 4, TIN = 5: HSSL: Program Integrity Error Tag Memory
0x507F	TCN = 4, TIN = 5: HSSL: Program Integrity Error Cache Memory
0x5080	TCN = 4, TIN = 5: HSSL: Program Integrity Error Scratchpad Memory
0x5081	TCN = 4, TIN = 5: HSSL: Program Integrity Error Bus Interface
0x5082	TCN = 4, TIN = 5: HSSL: Program Integrity Error Dual Error Detected
0x5083	TCN = 4, TIN = 5: HSSL: Program Integrity Error Safety Protection Error Detected
0x5084	TCN = 4, TIN = 5: HSSL: Program Integrity Error Bus Slave Access Indicator
0x5085	TCN = 4, TIN = 5: Ethernet: Program Integrity Error Detected
0x5086	TCN = 4, TIN = 5: Ethernet: Program Integrity Error Tag Memory
0x5087	TCN = 4, TIN = 5: Ethernet: Program Integrity Error Cache Memory
0x5088	TCN = 4, TIN = 5: Ethernet: Program Integrity Error Scratchpad Memory
0x5089	TCN = 4, TIN = 5: Ethernet: Program Integrity Error Bus Interface
0x508A	TCN = 4, TIN = 5: Ethernet: Program Integrity Error Dual Error Detected
0x508B	TCN = 4, TIN = 5: Ethernet: Program Integrity Error Safety Protection Error Detected
0x508C	TCN = 4, TIN = 5: Ethernet: Program Integrity Error Bus Slave Access Indicator
0x508D	TCN = 4, TIN = 5: HSM: Program Integrity Error Detected
0x508E	TCN = 4, TIN = 5: HSM: Program Integrity Error Tag Memory
0x508F	TCN = 4, TIN = 5: HSM: Program Integrity Error Cache Memory
0x5090	TCN = 4, TIN = 5: HSM: Program Integrity Error Scratchpad Memory
0x5091	TCN = 4, TIN = 5: HSM: Program Integrity Error Bus Interface
0x5092	TCN = 4, TIN = 5: HSM: Program Integrity Error Dual Error Detected
0x5093	TCN = 4, TIN = 5: HSM: Program Integrity Error Safety Protection Error Detected
0x5094	TCN = 4, TIN = 5: HSM: Program Integrity Error Bus Slave Access Indicator
0x5095	TCN = 4, TIN = 5: CPU0 SRI PMI: Program Integrity Error Detected
0x5096	TCN = 4, TIN = 5: CPU0 SRI PMI: Program Integrity Error Tag Memory
0x5097	TCN = 4, TIN = 5: CPU0 SRI PMI: Program Integrity Error Cache Memory
0x5098	TCN = 4, TIN = 5: CPU0 SRI PMI: Program Integrity Error Scratchpad Memory
0x5099	TCN = 4, TIN = 5: CPU0 SRI PMI: Program Integrity Error Bus Interface
0x509A	TCN = 4, TIN = 5: CPU0 SRI PMI: Program Integrity Error Dual Error Detected
0x509B	TCN = 4, TIN = 5: CPU0 SRI PMI: Program Integrity Error Safety Protection Error Detected

Value	Description
0x509C	TCN = 4, TIN = 5: CPU0 SRI PMI: Program Integrity Error Bus Slave Access Indicator
0x509D	TCN = 4, TIN = 5: CPU1 SRI PMI: Program Integrity Error Detected
0x509E	TCN = 4, TIN = 5: CPU1 SRI PMI: Program Integrity Error Tag Memory
0x509F	TCN = 4, TIN = 5: CPU1 SRI PMI: Program Integrity Error Cache Memory
0x50A0	TCN = 4, TIN = 5: CPU1 SRI PMI: Program Integrity Error Scratchpad Memory
0x50A1	TCN = 4, TIN = 5: CPU1 SRI PMI: Program Integrity Error Bus Interface
0x50A2	TCN = 4, TIN = 5: CPU1 SRI PMI: Program Integrity Error Dual Error Detected
0x50A3	TCN = 4, TIN = 5: CPU1 SRI PMI: Program Integrity Error Safety Protection Error Detected
0x50A4	TCN = 4, TIN = 5: CPU1 SRI PMI: Program Integrity Error Bus Slave Access Indicator
0x50A5	TCN = 4, TIN = 5: CPU2 SRI PMI: Program Integrity Error Detected
0x50A6	TCN = 4, TIN = 5: CPU2 SRI PMI: Program Integrity Error Tag Memory
0x50A7	TCN = 4, TIN = 5: CPU2 SRI PMI: Program Integrity Error Cache Memory
0x50A8	TCN = 4, TIN = 5: CPU2 SRI PMI: Program Integrity Error Scratchpad Memory
0x50A9	TCN = 4, TIN = 5: CPU2 SRI PMI: Program Integrity Error Bus Interface
0x50AA	TCN = 4, TIN = 5: CPU2 SRI PMI: Program Integrity Error Dual Error Detected
0x50AB	TCN = 4, TIN = 5: CPU2 SRI PMI: Program Integrity Error Safety Protection Error Detected
0x50AC	TCN = 4, TIN = 5: CPU2 SRI PMI: Program Integrity Error Bus Slave Access Indicator
0x50AD	TCN = 4, TIN = 5: SRI DAM: Program Integrity Error Detected
0x50AE	TCN = 4, TIN = 5: SRI DAM: Program Integrity Error Tag Memory
0x50AF	TCN = 4, TIN = 5: SRI DAM: Program Integrity Error Cache Memory
0x50B0	TCN = 4, TIN = 5: SRI DAM: Program Integrity Error Scratchpad Memory
0x50B1	TCN = 4, TIN = 5: SRI DAM: Program Integrity Error Bus Interface
0x50B2	TCN = 4, TIN = 5: SRI DAM: Program Integrity Error Dual Error Detected
0x50B3	TCN = 4, TIN = 5: SRI DAM: Program Integrity Error Safety Protection Error Detected
0x50B4	TCN = 4, TIN = 5: SRI DAM: Program Integrity Error Bus Slave Access Indicator
0x50B5	TCN = 4, TIN = 5: Cerberus Back Bone Bus ED: Program Integrity Error Detected
0x50B6	TCN = 4, TIN = 5: Cerberus Back Bone Bus ED: Program Integrity Error Tag Memory
0x50B7	TCN = 4, TIN = 5: Cerberus Back Bone Bus ED: Program Integrity Error Cache Memory
0x50B8	TCN = 4, TIN = 5: Cerberus Back Bone Bus ED: Program Integrity Error Scratchpad Memory
0x50B9	TCN = 4, TIN = 5: Cerberus Back Bone Bus ED: Program Integrity Error Bus Interface
0x50BA	TCN = 4, TIN = 5: Cerberus Back Bone Bus ED: Program Integrity Error Dual Error Detected

Value	Description
0x50BB	TCN = 4, TIN = 5: Cerberus Back Bone Bus ED: Program Integrity Error Safety Protection Error Detected
0x50BC	TCN = 4, TIN = 5: Cerberus Back Bone Bus ED: Program Integrity Error Bus Slave Access Indicator
0x50BD	TCN = 4, TIN = 5: CIF Master on Back Bone Bus ED: Program Integrity Error Detected
0x50BE	TCN = 4, TIN = 5: CIF Master on Back Bone Bus ED: Program Integrity Error Tag Memory
0x50BF	TCN = 4, TIN = 5: CIF Master on Back Bone Bus ED: Program Integrity Error Cache Memory
0x50C0	TCN = 4, TIN = 5: CIF Master on Back Bone Bus ED: Program Integrity Error Scratchpad Memory
0x50C1	TCN = 4, TIN = 5: CIF Master on Back Bone Bus ED: Program Integrity Error Bus Interface
0x50C2	TCN = 4, TIN = 5: CIF Master on Back Bone Bus ED: Program Integrity Error Dual Error Detected
0x50C3	TCN = 4, TIN = 5: CIF Master on Back Bone Bus ED: Program Integrity Error Safety Protection Error Detected
0x50C4	TCN = 4, TIN = 5: CIF Master on Back Bone Bus ED: Program Integrity Error Bus Slave Access Indicator
0x50C5	TCN = 4, TIN = 5: LMU Master on Back Bone Bus ED: Program Integrity Error Detected
0x50C6	TCN = 4, TIN = 5: LMU Master on Back Bone Bus ED: Program Integrity Error Tag Memory
0x50C7	TCN = 4, TIN = 5: LMU Master on Back Bone Bus ED: Program Integrity Error Cache Memory
0x50C8	TCN = 4, TIN = 5: LMU Master on Back Bone Bus ED: Program Integrity Error Scratchpad Memory
0x50C9	TCN = 4, TIN = 5: LMU Master on Back Bone Bus ED: Program Integrity Error Bus Interface
0x50CA	TCN = 4, TIN = 5: LMU Master on Back Bone Bus ED: Program Integrity Error Dual Error Detected
0x50CB	TCN = 4, TIN = 5: LMU Master on Back Bone Bus ED: Program Integrity Error Safety Protection Error Detected
0x50CC	TCN = 4, TIN = 5: LMU Master on Back Bone Bus ED: Program Integrity Error Bus Slave Access Indicator
0x50CD	TCN = 4, TIN = 6: no bit is set in DIETR
0x50CE	TCN = 4, TIN = 6: CPU0, DMI NonSafe: Data Integrity Error Detected
0x50CF	TCN = 4, TIN = 6: CPU0, DMI NonSafe: Data Integrity Error Tag Memory
0x50D0	TCN = 4, TIN = 6: CPU0, DMI NonSafe: Data Integrity Error Cache Memory
0x50D1	TCN = 4, TIN = 6: CPU0, DMI NonSafe: Data Integrity Error Scratchpad Memory
0x50D2	TCN = 4, TIN = 6: CPU0, DMI NonSafe: Data Integrity Error Bus Interface
0x50D3	TCN = 4, TIN = 6: CPU0, DMI NonSafe: Data Integrity Error Dual Error Detected
0x50D4	TCN = 4, TIN = 6: CPU0, DMI NonSafe: Data Integrity Error Safety Pro-tection Error Detected

Value	Description
0x50D5	TCN = 4, TIN = 6: CPU0, DMI NonSafe: Data Integrity Error Bus Slave Access Indicator
0x50D6	TCN = 4, TIN = 6: CPU0, DMI Safe: Data Integrity Error Detected
0x50D7	TCN = 4, TIN = 6: CPU0, DMI Safe: Data Integrity Error Tag Memory
0x50D8	TCN = 4, TIN = 6: CPU0, DMI Safe: Data Integrity Error Cache Memory
0x50D9	TCN = 4, TIN = 6: CPU0, DMI Safe: Data Integrity Error Scratchpad Memory
0x50DA	TCN = 4, TIN = 6: CPU0, DMI Safe: Data Integrity Error Bus Interface
0x50DB	TCN = 4, TIN = 6: CPU0, DMI Safe: Data Integrity Error Dual Error Detected
0x50DC	TCN = 4, TIN = 6: CPU0, DMI Safe: Data Integrity Error Safety Protection Error Detected
0x50DD	TCN = 4, TIN = 6: CPU0, DMI Safe: Data Integrity Error Bus Slave Access Indicator
0x50DE	TCN = 4, TIN = 6: CPU1, DMI NonSafe: Data Integrity Error Detected
0x50DF	TCN = 4, TIN = 6: CPU1, DMI NonSafe: Data Integrity Error Tag Memory
0x50E0	TCN = 4, TIN = 6: CPU1, DMI NonSafe: Data Integrity Error Cache Memory
0x50E1	TCN = 4, TIN = 6: CPU1, DMI NonSafe: Data Integrity Error Scratchpad Memory
0x50E2	TCN = 4, TIN = 6: CPU1, DMI NonSafe: Data Integrity Error Bus Interface
0x50E3	TCN = 4, TIN = 6: CPU1, DMI NonSafe: Data Integrity Error Dual Error Detected
0x50E4	TCN = 4, TIN = 6: CPU1, DMI NonSafe: Data Integrity Error Safety Pro-tection Error Detected
0x50E5	TCN = 4, TIN = 6: CPU1, DMI NonSafe: Data Integrity Error Bus Slave Access Indicator
0x50E6	TCN = 4, TIN = 6: CPU1, DMI Safe: Data Integrity Error Detected
0x50E7	TCN = 4, TIN = 6: CPU1, DMI Safe: Data Integrity Error Tag Memory
0x50E8	TCN = 4, TIN = 6: CPU1, DMI Safe: Data Integrity Error Cache Memory
0x50E9	TCN = 4, TIN = 6: CPU1, DMI Safe: Data Integrity Error Scratchpad Memory
0x50EA	TCN = 4, TIN = 6: CPU1, DMI Safe: Data Integrity Error Bus Interface
0x50EB	TCN = 4, TIN = 6: CPU1, DMI Safe: Data Integrity Error Dual Error Detected
0x50EC	TCN = 4, TIN = 6: CPU1, DMI Safe: Data Integrity Error Safety Protection Error Detected
0x50ED	TCN = 4, TIN = 6: CPU1, DMI Safe: Data Integrity Error Bus Slave Access Indicator
0x50EE	TCN = 4, TIN = 6: CPU2, DMI NonSafe: Data Integrity Error Detected
0x50EF	TCN = 4, TIN = 6: CPU2, DMI NonSafe: Data Integrity Error Tag Memory
0x50F0	TCN = 4, TIN = 6: CPU2, DMI NonSafe: Data Integrity Error Cache Memory
0x50F1	TCN = 4, TIN = 6: CPU2, DMI NonSafe: Data Integrity Error Scratchpad Memory
0x50F2	TCN = 4, TIN = 6: CPU2, DMI NonSafe: Data Integrity Error Bus Interface
0x50F3	TCN = 4, TIN = 6: CPU2, DMI NonSafe: Data Integrity Error Dual Error Detected
0x50F4	TCN = 4, TIN = 6: CPU2, DMI NonSafe: Data Integrity Error Safety Pro-tection Error Detected

Value	Description
0x50F5	TCN = 4, TIN = 6: CPU2, DMI NonSafe: Data Integrity Error Bus Slave Access Indicator
0x50F6	TCN = 4, TIN = 6: CPU2, DMI Safe: Data Integrity Error Detected
0x50F7	TCN = 4, TIN = 6: CPU2, DMI Safe: Data Integrity Error Tag Memory
0x50F8	TCN = 4, TIN = 6: CPU2, DMI Safe: Data Integrity Error Cache Memory
0x50F9	TCN = 4, TIN = 6: CPU2, DMI Safe: Data Integrity Error Scratchpad Memory
0x50FA	TCN = 4, TIN = 6: CPU2, DMI Safe: Data Integrity Error Bus Interface
0x50FB	TCN = 4, TIN = 6: CPU2, DMI Safe: Data Integrity Error Dual Error Detected
0x50FC	TCN = 4, TIN = 6: CPU2, DMI Safe: Data Integrity Error Safety Protection Error Detected
0x50FD	TCN = 4, TIN = 6: CPU2, DMI Safe: Data Integrity Error Bus Slave Access Indicator
0x50FE	TCN = 4, TIN = 6: SDMA MoveEngine0: Data Integrity Error Detected
0x50FF	TCN = 4, TIN = 6: SDMA MoveEngine0: Data Integrity Error Tag Memory
0x5100	TCN = 4, TIN = 6: SDMA MoveEngine0: Data Integrity Error Cache Memory
0x5101	TCN = 4, TIN = 6: SDMA MoveEngine0: Data Integrity Error Scratchpad Memory
0x5102	TCN = 4, TIN = 6: SDMA MoveEngine0: Data Integrity Error Bus Interface
0x5103	TCN = 4, TIN = 6: SDMA MoveEngine0: Data Integrity Error Dual Error Detected
0x5104	TCN = 4, TIN = 6: SDMA MoveEngine0: Data Integrity Error Safety Protection Error Detected
0x5105	TCN = 4, TIN = 6: SDMA MoveEngine0: Data Integrity Error Bus Slave Access Indicator
0x5106	TCN = 4, TIN = 6: SDMA MoveEngine1: Data Integrity Error Detected
0x5107	TCN = 4, TIN = 6: SDMA MoveEngine1: Data Integrity Error Tag Memory
0x5108	TCN = 4, TIN = 6: SDMA MoveEngine1: Data Integrity Error Cache Memory
0x5109	TCN = 4, TIN = 6: SDMA MoveEngine1: Data Integrity Error Scratchpad Memory
0x510A	TCN = 4, TIN = 6: SDMA MoveEngine1: Data Integrity Error Bus Interface
0x510B	TCN = 4, TIN = 6: SDMA MoveEngine1: Data Integrity Error Dual Error Detected
0x510C	TCN = 4, TIN = 6: SDMA MoveEngine1: Data Integrity Error Safety Protection Error Detected
0x510D	TCN = 4, TIN = 6: SDMA MoveEngine1: Data Integrity Error Bus Slave Access Indicator
0x510E	TCN = 4, TIN = 6: Cerberus: Data Integrity Error Detected
0x510F	TCN = 4, TIN = 6: Cerberus: Data Integrity Error Tag Memory
0x5110	TCN = 4, TIN = 6: Cerberus: Data Integrity Error Cache Memory
0x5111	TCN = 4, TIN = 6: Cerberus: Data Integrity Error Scratchpad Memory
0x5112	TCN = 4, TIN = 6: Cerberus: Data Integrity Error Bus Interface
0x5113	TCN = 4, TIN = 6: Cerberus: Data Integrity Error Dual Error Detected
0x5114	TCN = 4, TIN = 6: Cerberus: Data Integrity Error Safety Protection Error Detected

Value	Description
0x5115	TCN = 4, TIN = 6: Cerberus: Data Integrity Error Bus Slave Access Indicator
0x5116	TCN = 4, TIN = 6: DMA MoveEngine0: Data Integrity Error Detected
0x5117	TCN = 4, TIN = 6: DMA MoveEngine0: Data Integrity Error Tag Memory
0x5118	TCN = 4, TIN = 6: DMA MoveEngine0: Data Integrity Error Cache Memory
0x5119	TCN = 4, TIN = 6: DMA MoveEngine0: Data Integrity Error Scratchpad Memory
0x511A	TCN = 4, TIN = 6: DMA MoveEngine0: Data Integrity Error Bus Interface
0x511B	TCN = 4, TIN = 6: DMA MoveEngine0: Data Integrity Error Dual Error Detected
0x511C	TCN = 4, TIN = 6: DMA MoveEngine0: Data Integrity Error Safety Protection Error Detected
0x511D	TCN = 4, TIN = 6: DMA MoveEngine0: Data Integrity Error Bus Slave Access Indicator
0x511E	TCN = 4, TIN = 6: DMA MoveEngine1: Data Integrity Error Detected
0x511F	TCN = 4, TIN = 6: DMA MoveEngine1: Data Integrity Error Tag Memory
0x5120	TCN = 4, TIN = 6: DMA MoveEngine1: Data Integrity Error Cache Memory
0x5121	TCN = 4, TIN = 6: DMA MoveEngine1: Data Integrity Error Scratchpad Memory
0x5122	TCN = 4, TIN = 6: DMA MoveEngine1: Data Integrity Error Bus Interface
0x5123	TCN = 4, TIN = 6: DMA MoveEngine1: Data Integrity Error Dual Error Detected
0x5124	TCN = 4, TIN = 6: DMA MoveEngine1: Data Integrity Error Safety Protection Error Detected
0x5125	TCN = 4, TIN = 6: DMA MoveEngine1: Data Integrity Error Bus Slave Access Indicator
0x5126	TCN = 4, TIN = 6: HSSL: Data Integrity Error Detected
0x5127	TCN = 4, TIN = 6: HSSL: Data Integrity Error Tag Memory
0x5128	TCN = 4, TIN = 6: HSSL: Data Integrity Error Cache Memory
0x5129	TCN = 4, TIN = 6: HSSL: Data Integrity Error Scratchpad Memory
0x512A	TCN = 4, TIN = 6: HSSL: Data Integrity Error Bus Interface
0x512B	TCN = 4, TIN = 6: HSSL: Data Integrity Error Dual Error Detected
0x512C	TCN = 4, TIN = 6: HSSL: Data Integrity Error Safety Protection Error Detected
0x512D	TCN = 4, TIN = 6: HSSL: Data Integrity Error Bus Slave Access Indicator
0x512E	TCN = 4, TIN = 6: Ethernet: Data Integrity Error Detected
0x512F	TCN = 4, TIN = 6: Ethernet: Data Integrity Error Tag Memory
0x5130	TCN = 4, TIN = 6: Ethernet: Data Integrity Error Cache Memory
0x5131	TCN = 4, TIN = 6: Ethernet: Data Integrity Error Scratchpad Memory
0x5132	TCN = 4, TIN = 6: Ethernet: Data Integrity Error Bus Interface
0x5133	TCN = 4, TIN = 6: Ethernet: Data Integrity Error Dual Error Detected
0x5134	TCN = 4, TIN = 6: Ethernet: Data Integrity Error Safety Protection Error Detected
0x5135	TCN = 4, TIN = 6: Ethernet: Data Integrity Error Bus Slave Access Indicator
0x5136	TCN = 4, TIN = 6: HSM: Data Integrity Error Detected
0x5137	TCN = 4, TIN = 6: HSM: Data Integrity Error Tag Memory
0x5138	TCN = 4, TIN = 6: HSM: Data Integrity Error Cache Memory
0x5139	TCN = 4, TIN = 6: HSM: Data Integrity Error Scratchpad Memory

Value	Description
0x513A	TCN = 4, TIN = 6: HSM: Data Integrity Error Bus Interface
0x513B	TCN = 4, TIN = 6: HSM: Data Integrity Error Dual Error Detected
0x513C	TCN = 4, TIN = 6: HSM: Data Integrity Error Safety Protection Error Detected
0x513D	TCN = 4, TIN = 6: HSM: Data Integrity Error Bus Slave Access Indicator
0x513E	TCN = 4, TIN = 6: CPU0 SRI PMI: Data Integrity Error Detected
0x513F	TCN = 4, TIN = 6: CPU0 SRI PMI: Data Integrity Error Tag Memory
0x5140	TCN = 4, TIN = 6: CPU0 SRI PMI: Data Integrity Error Cache Memory
0x5141	TCN = 4, TIN = 6: CPU0 SRI PMI: Data Integrity Error Scratchpad Memory
0x5142	TCN = 4, TIN = 6: CPU0 SRI PMI: Data Integrity Error Bus Interface
0x5143	TCN = 4, TIN = 6: CPU0 SRI PMI: Data Integrity Error Dual Error Detected
0x5144	TCN = 4, TIN = 6: CPU0 SRI PMI: Data Integrity Error Safety Protection Error Detected
0x5145	TCN = 4, TIN = 6: CPU0 SRI PMI: Data Integrity Error Bus Slave Access Indicator
0x5146	TCN = 4, TIN = 6: CPU1 SRI PMI: Data Integrity Error Detected
0x5147	TCN = 4, TIN = 6: CPU1 SRI PMI: Data Integrity Error Tag Memory
0x5148	TCN = 4, TIN = 6: CPU1 SRI PMI: Data Integrity Error Cache Memory
0x5149	TCN = 4, TIN = 6: CPU1 SRI PMI: Data Integrity Error Scratchpad Memory
0x514A	TCN = 4, TIN = 6: CPU1 SRI PMI: Data Integrity Error Bus Interface
0x514B	TCN = 4, TIN = 6: CPU1 SRI PMI: Data Integrity Error Dual Error Detected
0x514C	TCN = 4, TIN = 6: CPU1 SRI PMI: Data Integrity Error Safety Protection Error Detected
0x514D	TCN = 4, TIN = 6: CPU1 SRI PMI: Data Integrity Error Bus Slave Access Indicator
0x514E	TCN = 4, TIN = 6: CPU2 SRI PMI: Data Integrity Error Detected
0x514F	TCN = 4, TIN = 6: CPU2 SRI PMI: Data Integrity Error Tag Memory
0x5150	TCN = 4, TIN = 6: CPU2 SRI PMI: Data Integrity Error Cache Memory
0x5151	TCN = 4, TIN = 6: CPU2 SRI PMI: Data Integrity Error Scratchpad Memory
0x5152	TCN = 4, TIN = 6: CPU2 SRI PMI: Data Integrity Error Bus Interface
0x5153	TCN = 4, TIN = 6: CPU2 SRI PMI: Data Integrity Error Dual Error Detected
0x5154	TCN = 4, TIN = 6: CPU2 SRI PMI: Data Integrity Error Safety Protection Error Detected
0x5155	TCN = 4, TIN = 6: CPU2 SRI PMI: Data Integrity Error Bus Slave Access Indicator
0x5156	TCN = 4, TIN = 6: SRI DAM: Data Integrity Error Detected
0x5157	TCN = 4, TIN = 6: SRI DAM: Data Integrity Error Tag Memory
0x5158	TCN = 4, TIN = 6: SRI DAM: Data Integrity Error Cache Memory
0x5159	TCN = 4, TIN = 6: SRI DAM: Data Integrity Error Scratchpad Memory
0x515A	TCN = 4, TIN = 6: SRI DAM: Data Integrity Error Bus Interface
0x515B	TCN = 4, TIN = 6: SRI DAM: Data Integrity Error Dual Error Detected
0x515C	TCN = 4, TIN = 6: SRI DAM: Data Integrity Error Safety Protection Error Detected
0x515D	TCN = 4, TIN = 6: SRI DAM: Data Integrity Error Bus Slave Access Indicator

Value	Description
0x515E	TCN = 4, TIN = 6: Cerberus Back Bone Bus ED: Data Integrity Error Detected
0x515F	TCN = 4, TIN = 6: Cerberus Back Bone Bus ED: Data Integrity Error Tag Memory
0x5160	TCN = 4, TIN = 6: Cerberus Back Bone Bus ED: Data Integrity Error Cache Memory
0x5161	TCN = 4, TIN = 6: Cerberus Back Bone Bus ED: Data Integrity Error Scratchpad Memory
0x5162	TCN = 4, TIN = 6: Cerberus Back Bone Bus ED: Data Integrity Error Bus Interface
0x5163	TCN = 4, TIN = 6: Cerberus Back Bone Bus ED: Data Integrity Error Dual Error Detected
0x5164	TCN = 4, TIN = 6: Cerberus Back Bone Bus ED: Data Integrity Error Safety Protection Error Detected
0x5165	TCN = 4, TIN = 6: Cerberus Back Bone Bus ED: Data Integrity Error Bus Slave Access Indicator
0x5166	TCN = 4, TIN = 6: CIF Master on Back Bone Bus ED: Data Integrity Error Detected
0x5167	TCN = 4, TIN = 6: CIF Master on Back Bone Bus ED: Data Integrity Error Tag Memory
0x5168	TCN = 4, TIN = 6: CIF Master on Back Bone Bus ED: Data Integrity Error Cache Memory
0x5169	TCN = 4, TIN = 6: CIF Master on Back Bone Bus ED: Data Integrity Error Scratchpad Memory
0x516A	TCN = 4, TIN = 6: CIF Master on Back Bone Bus ED: Data Integrity Error Bus Interface
0x516B	TCN = 4, TIN = 6: CIF Master on Back Bone Bus ED: Data Integrity Error Dual Error Detected
0x516C	TCN = 4, TIN = 6: CIF Master on Back Bone Bus ED: Data Integrity Error Safety Protection Error Detected
0x516D	TCN = 4, TIN = 6: CIF Master on Back Bone Bus ED: Data Integrity Error Bus Slave Access Indicator
0x516E	TCN = 4, TIN = 6: LMU Master on Back Bone Bus ED: Data Integrity Error Detected
0x516F	TCN = 4, TIN = 6: LMU Master on Back Bone Bus ED: Data Integrity Error Tag Memory
0x5170	TCN = 4, TIN = 6: LMU Master on Back Bone Bus ED: Data Integrity Error Cache Memory
0x5171	TCN = 4, TIN = 6: LMU Master on Back Bone Bus ED: Data Integrity Error Scratchpad Memory
0x5172	TCN = 4, TIN = 6: LMU Master on Back Bone Bus ED: Data Integrity Error Bus Interface
0x5173	TCN = 4, TIN = 6: LMU Master on Back Bone Bus ED: Data Integrity Error Dual Error Detected
0x5174	TCN = 4, TIN = 6: LMU Master on Back Bone Bus ED: Data Integrity Error Safety Protection Error Detected
0x5175	TCN = 4, TIN = 6: LMU Master on Back Bone Bus ED: Data Integrity Error Bus Slave Access Indicator

Value	Description
0x5176	TCN = 5, TIN = 0: Unknown
0x5177	TCN = 5, TIN = 1: Arithmetic Overflow
0x5178	TCN = 5, TIN = 2: Sticky Arithmetic Overflow
0x5179	TCN = 5, TIN = >2
0x517A	TCN = 6, TIN = 0-255: System Call
0x517B	TCN = 7, TIN = 0: no SMUT, no ESR1, no ESR0 bit
0x517C	TCN = 7, TIN = 0: NMI ESR0
0x517D	TCN = 7, TIN = 0: NMI ESR1
0x517E	TCN = 7, TIN = 0: NMI SMUT additional information under
0x517F	TCN = 7, TIN = 0: NMI SMUT CPU0: MPU error (registers, DSPR, PSPR)
0x5180	TCN = 7, TIN = 0: NMI SMUT CPU1: MPU error (registers, DSPR, PSPR)
0x5181	TCN = 7, TIN = 0: NMI SMUT CPU2: MPU error (registers, DSPR, PSPR)
0x5182	TCN = 7, TIN = 0: NMI SMUT IR: Interrupt Monitor: EDC error
0x5183	TCN = 7, TIN = 0: NMI SMUT CGU WDT: System PLL OSC_WDT: out of range input clock
0x5184	TCN = 7, TIN = 0: NMI SMUT CGU PLL LossOfLock: System PLL VCO Loss-of-Lock Event
0x5185	TCN = 7, TIN = 0: NMI SMUT CGU PLL ERAY LossOfLock: PLL_ERAY VCO Loss-of-Lock Event
0x5186	TCN = 7, TIN = 0: NMI SMUT CGU STM: Clock monitoring: Out of range frequency STM
0x5187	TCN = 7, TIN = 0: NMI SMUT CGU PLLERAY: Clock monitoring: Out of range frequency PLL_ERAY
0x5188	TCN = 7, TIN = 0: NMI SMUT CGU PLL: Clock monitoring: Out of range frequency System PLL
0x5189	TCN = 7, TIN = 0: NMI SMUT CGU SRI: Clock monitoring: Out of range frequency SRI
0x518A	TCN = 7, TIN = 0: NMI SMUT CGU SPB: Clock monitoring: Out of range frequency SPB
0x518B	TCN = 7, TIN = 0: NMI SMUT CGU CPU0: Clock monitoring: Out of range frequency CPU0
0x518C	TCN = 7, TIN = 0: NMI SMUT: EVR 1.2V digital undervoltage
0x518D	TCN = 7, TIN = 0: NMI SMUT: EVR 1.2V digital overvoltage
0x518E	TCN = 7, TIN = 0: NMI SMUT: EVR 3.3V undervoltage
0x518F	TCN = 7, TIN = 0: NMI SMUT: EVR 3.3V overvoltage
0x5190	TCN = 7, TIN = 0: NMI SMUT: EVR 5V supply undervoltage
0x5191	TCN = 7, TIN = 0: NMI SMUT: EVR 5V supply overvoltage
0x5192	TCN = 7, TIN = 0: NMI SMUT: safety watchdog timeout
0x5193	TCN = 7, TIN = 0: NMI SMUT: watchdog0 timeout
0x5194	TCN = 7, TIN = 0: NMI SMUT: watchdog1 timeout
0x5195	TCN = 7, TIN = 0: NMI SMUT: watchdog2 timeout
0x5196	TCN = 7, TIN = >0
	TRAPS CPU1

Value	Description
0x6xxx	xxx refer to CPU0
	TRAPS CPU2
0x7xxx	xxx refer to CPU0

Regular Entry (Type 0x20)

This regular entry is written if the SCI is triggered by the SCC script to store an error. This usually happens if an check on last tsw response fails. The SCI stores the complete TSW response buffer which contains the response of last executed TSW into Log Entries detail data.

Error Entry (Type 0x40)

Each time that the SCI detects an internal error, the information is also stored to the Log Entry. The first 3 bytes data–field contains information about the error class:

0xFFFF00 : SCI internal error

0xFFF800 / 0xFFF900: uC Trap

The fourth byte gives an information about the error. For details please refer to MDG1 Validation SW Documentation.

End Entry (Type 0x60)

This Log Entry is written if SCI has executed the last command of the SCC script. The content of Detail data is the string "End Of SC Reached"

Info Entry (Type 0x80)

This Log Entry is written if the SCI is triggered by the SCC script with command warningil() [APID 43] to store data. The SCI stores the complete TSW response buffer which contains the response of last executed TSW into Log Entries detail data. This entry type is used to put information into logging flash area which should not be counted as error.

Test step commands

.

00 - Analyse AST FLASH.

Read log memory entries and generate summary of faults.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	65	00	01

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	65	xx	xx	01	...

Parameters

Analyse FLASH.

6 (1*)	+0	+2	+3	+4	+5	+6
	ERROR	**PON**	**EOF**	**TRAP**	**SCI_ERR**	**AST_ERR**

... (1*)	+7	+8
	IRI_ERR	**TIME**

Test type and index of a failed AST test.

18 (1..n*)	+0	+1
	TYPE	**IDX**

Details

Analyse FLASH.

Name	Length	Fmt	Description
ERROR	**2**	**xxxx**	**Error summary flags** Summary error flags. **Bit(s)** / **Description**: **0** — **SCI has never finished** **1** — **At least one SCI or script failure has been detected** **2** — **At least one AST failure has been detected** **3** — **At least one IRI failure has been detected** **4** — **At least one CPU trap has occured** **5:14** — **Not used, always set to zero** **15** — **Faultmemory is full**
PON	**1**	**xx**	**Number of power-ons** Number of power-ons. NOTE: Loss of ESBs from TC17 dropped.
EOF	**1**	**xx**	**Finished SCI runs** Number of finished SCI script runs.
TRAP	**1**	**xx**	**Trap resets** Number of trap resets.
SCI_ERR	**1**	**xx**	**SCI internal errors** Number of reported SCI script or SCI internal errors.
AST_ERR	**1**	**xx**	**AST errors** Number of reported AST errors.
IRI_ERR	**1**	**xx**	**IRI errors** Number of reported InitialRunIn errors.
TIME	**4**	**xxxxxxxx**	**SCI script runtime**

Name	Length	Fmt	Description
			Runtime of last finished SCI script. Resolution is 1µs/MSB first. If no log memory entry with error ID = 0x60 can be found, this value is set to 0..

Test type and index of a failed AST test.

Name	Length	Fmt	Description
TYPE	**1**	**xx**	**Test type** AST test type **0_H:** memory **1_H:** logic
IDX	**2**	**xxxx**	**Log entry index** If any AST error has been detected, the index of the log entry for reading with command 0x01 is reported.

.

01 - Read log entry.

Read AST log entry.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	65	01	01	...

Parameters

Index of log entry.

5	+0	+2
(1*)	**IDX**	**PART**

Details

Index of log entry.

Name	Length	Fmt	Description
IDX	**2**	**xxxx**	**Index** Index of log entry: 0x00 – 0xFF. At the moment 256 log records are possible.
PART	**1**	**xx**	**Part** Select log entry part. *Because the size of a KWP frame is limited to 256 byte it's necessary to split reading.* **Bit(s)** – **Description** **7:1** – **Not used** **0** – **Part**

Name	Length	Fmt	Description	
			Bit(s)	**Description**
				0_B: Header
				1_B: Data len and data

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	65	xx	xx	01	...

Parameters

SCI runtime.

6	+0	+4	+5	+6	+7	+8
(1*)	**TIME**	**SCA2**	**SCA1**	**SCA0**	**TYPE**	**BYTES**

log entry data.

16	+0	+2
(1..n*)	**BYTES**	**DATA**

Details

SCI runtime.

Name	Length	Fmt	Description
TIME	**4**	**xxxxxxxx**	**SCI script runtime** Runtime of SCI script that created the log entry. Resolution is 1µs/MSB first.
SCA2	**1**	**xx**	**SC addr byte 2** SCA2-0 are the lower 3 bytes of the current SC address (MSB first).
SCA1	**1**	**xx**	**SC addr byte 1** SC address byte 1.
SCA0	**1**	**xx**	**SC addr byte 0** SC address byte 0.
TYPE	**1**	**xx**	**Entry type** Type of log entry. TBD Table of values.
BYTES	**2**	**xxxx**	**Number of bytes** Number of log entry data bytes. At the moment at most 248.

log entry data.

Name	Length	Fmt	Description
BYTES	**2**	**xxxx**	**Number of bytes** Number of log entry data bytes. At the moment at most 248.
DATA	**1**	**xx..**	**Data byte**

Name	Length	Fmt	Description
			log entry data byte.

.

02 - Read log entry by type.

Read AST log entry by type and pattern.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	65	02	01	...

Parameters

Index of log entry.

5	+0	+1	+3
(1*)	**PAGID**	**IDX**	**PART**

Optional search pattern. Position + 1..20 bytes.

9	+0	+1
(1*)	**PATPOS**	**PATDATA**

Details

Index of log entry.

Name	Length	Fmt	Description
PAGID	**1**	**xx**	**PagId** Log entry type (0x00,20,40,60,80).
IDX	**2**	**xxxx**	**Index** Index of log entry: 0x00 – 0xFF. At the moment 256 log records are possible.
PART	**1**	**xx**	**Part** Select log entry part. *Because the size of a KWP frame is limited to 256 byte it's necessary to split reading.* **Bit(s)** / **Description** **7:1** / **Not used** **0** / **Part** **0_B:** Header **1_B:** Data len and data

Optional search pattern. Position + 1..20 bytes.

Name	Length	Fmt	Description
PATPOS	**1**	**xx**	**PatPos**

Name	Length	Fmt	Description
			Pattern position. First data byte has index 0.
PATDATA	**1**	**xx**	**PatData** Pattern data. Up to 20 Bytes.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	65	xx	xx	01	...

Parameters

SCI runtime.

6	+0	+4	+5	+6	+7	+8
(1*)	**TIME**	**SCA2**	**SCA1**	**SCA0**	**TYPE**	**BYTES**

log entry data.

16	+0	+2
(1..n*)	**BYTES**	**DATA**

Details

SCI runtime.

Name	Length	Fmt	Description
TIME	**4**	**xxxxxxxx**	**SCI script runtime** Runtime of SCI script that created the log entry. Resolution is 1µs/MSB first.
SCA2	**1**	**xx**	**SC addr byte 2** SCA2-0 are the lower 3 bytes of the current SC address (MSB first).
SCA1	**1**	**xx**	**SC addr byte 1** SC address byte 1.
SCA0	**1**	**xx**	**SC addr byte 0** SC address byte 0.
TYPE	**1**	**xx**	**Entry type** Type of log entry. TBD Table of values.
BYTES	**2**	**xxxx**	**Number of bytes** Number of log entry data bytes. At the moment at most 248.

log entry data.

Name	Length	Fmt	Description
BYTES	**2**	**xxxx**	**Number of bytes** Number of log entry data bytes. At the moment at most 248.
DATA	**1**	**xx..**	**Data byte**

Name	Length	Fmt	Description
			log entry data byte.

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**

Code	Description
0x01	**TSW CPU manufacturer not supported.**
0x02	**TSW CPU device not supported.**
0x03	**TSW Invalid command.**
0x04	**TSW Invalid log entry index.**
0x05	**TSW Pattern for type not found.**
0xFF	**HAL busy**
0xFE	**Generic HAL error**
0xFD	**HAL not compatible to current CPU**
0xFC	**Invalid params**
0xFB	**HAL module not implemented**
0xFA	**HAL function not implemented**
0xF9	**HAL timeout**

TSW_74 - Gateway

Test step description

This test step is used to wrap communication data in different other protocols. Actual available KWP2000 on ASC and ISO 15765-2 on CAN

Test step commands

.

01 - Uart init

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	74	01	01	...

Parameters

Parameter set for initialization ASC module

5 (1*)	+0	+1	+2	+3	+4	+5
	TXPORT	**TXPIN**	**RXPORT**	**RXPIN**	**ALTIN**	**BDR**

... (1*)	+9
	MODE

Details

Parameter set for initialization ASC module

Name	Length	Fmt	Description
TXPORT	**1**	**xx**	**Tx Port for UART communication** Tx Port for UART communication
TXPIN	**1**	**xx**	**Tx Pin for UART communication** Tx Pin for UART communication
RXPORT	**1**	**xx**	**Rx Port for UART communication** Tx Port for UART communication
RXPIN	**1**	**xx**	**Rx Pin for UART communication** Tx Pin for UART communication
ALTIN	**1**	**xx**	**Alternative Input** ASC alternative input line. Selection of input line [0..7] specific to selected ASC module. Please refer to corresponding controller documentation for specific pin assignment. (Only for IFX)
BDR	**4**	**xxxxxxxx**	**Baudrate for UART communication** Baudrate in Baud. For 1 MBaud use 1 000 000.
MODE	**1**	**xx**	**Mode of communication**

Name	Length	Fmt	Description
			Table 1. Mode of Communication Value / Mode 0 / UART 1 / LIN(not implemented)

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	74	xx	xx	01	...

Parameters

Command response data

6	+0	+1
(1*)	**MODULE**	**BDR**

Details

Command response data
Command response data.

Name	Length	Fmt	Description
MODULE	**1**	**xx**	**Choosen module instance** Module id according to given port/pin pair.
BDR	**4**	**xxxxxxxx**	**Calculated baudrate** Calculated baudrate in Baud.

.

02 - Uart transmit

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	74	02	01	...

Parameters

Parameter set for transmiting UART

5	+0	+1	+2	+3	+4	+5
(1*)	**TXPORT**	**TXPIN**	**RXPORT**	**RXPIN**	**RESP**	**WAIT**

...	+7	+9	+11
(1*)	**TO**	**P4**	**DATA**

Details

Parameter set for transmiting UART

Name	Length	Fmt	Description
TXPORT	**1**	**xx**	**Tx Port for UART communication** Tx Port for UART communication
TXPIN	**1**	**xx**	**Tx Pin for UART communication** Tx Pin for UART communication
RXPORT	**1**	**xx**	**Rx Port for UART communication** Tx Port for UART communication
RXPIN	**1**	**xx**	**Rx Pin for UART communication** Tx Pin for UART communication
RESP	**1**	**xx**	**Response is expected** In case of an expected response set this to "true".
WAIT	**2**	**xxxx**	**Time till enable Rx** Time to wait before enabling Rx path in ms. Only used if Response is "true".
TO	**2**	**xxxx**	**Reception timeout** Time to wait for an response in ms. Only used if Response is "true".
P4	**2**	**xxxx**	**Inter byte time** Inter byte time for tester request in µs. (according P4 time). Time is used for reception as well. (according P1 time).
DATA	**1**	**xx..**	**Transmit Data** Data to be transmitted. Max length: 261 Byte Hint: maximum length may be restricted by communication to your ECU

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	74	xx	xx	01	...

Parameters

Command response data

6	+0	+1
(1*)	**MODULE**	**DATA**

Details

Command response data
Command response data.

Name	Length	Fmt	Description
MODULE	**1**	**xx**	**Choosen module instance**

Name	Length	Fmt	Description
			Module id according to given port/pin pair.
DATA	**1**	**xx..**	**Receive Data** Received Data sent by target device.

.

03 - Uart transmit KWP

Request:

Source > Gateway

Header	SVC	TSW	CMD	VER	...	Data	CS
FMT	80	74	03	01	...	....	...

Gateway > Target

Header	Data	CS
FMT	....	..

Response:

Target > Gateway

Header	Data	CS
FMT	....	..

Gateway > Source

Header	SVC	TSW	ESVC	ETSW	VER	Data	CS
FMT	C0	74	..	..	01	....	...

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	74	03	01	...

Parameters

Parameter set for transmiting UART

5	+0	+1	+2	+3	+4	+5
(1*)	**TXPORT**	**TXPIN**	**RXPORT**	**RXPIN**	**RESP**	**WAIT**

...	+7	+9	+11
(1*)	**TO**	**P4**	**DATA**

Details

Parameter set for transmiting UART

Put PDU in a KWP frame

Name	Length	Fmt	Description
TXPORT	**1**	**xx**	**Tx Port for UART communication** Tx Port for UART communication
TXPIN	**1**	**xx**	**Tx Pin for UART communication** Tx Pin for UART communication
RXPORT	**1**	**xx**	**Rx Port for UART communication** Tx Port for UART communication
RXPIN	**1**	**xx**	**Rx Pin for UART communication** Tx Pin for UART communication
RESP	**1**	**xx**	**Response is expected** In case of an expected response set this to "true".
WAIT	**2**	**xxxx**	**Time till enable Rx** Time to wait before enabling Rx path in ms. Only used if Response is "true".
TO	**2**	**xxxx**	**Reception timeout** Time to wait for an response in ms. Only used if Response is "true".
P4	**2**	**xxxx**	**Inter byte time** Inter byte time for tester request in µs. (according P4 time). Time is used for reception as well. (according P1 time).
DATA	**1**	**xx..**	**Transmit Data** Data to be transmitted. Max length: 256 Byte Hint: maximum length may be restricted by communication to your ECU

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	74	xx	xx	01	...

Parameters

Command response data

6	+0	+1
(1*)	**MODULE**	**DATA**

Details

Command response data
Command response data.

Name	Length	Fmt	Description
MODULE	**1**	**xx**	**Choosen module instance** Module id according to given port/pin pair.
DATA	**1**	**xx..**	**Receive Data** Received Data sent by target device.

.

10 - CAN Tp Request

Transmit your data via CAN according ISO 15765-2 transport protocol.

Note

Only normal addressing is supported.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	74	10	01	...

Parameters

Parameter set for transmiting data

5	+0	+1	+2	+3
(1*)	**NODE**	**TXMSGB**	**RXMSGB**	**DATA**

Details

Parameter set for transmiting data

Transmit data according ISO 15765-2 transport protocol.

Note

This teststep doesn't configure CAN. Please use TSW_18 to do the initialisation of CAN used for this communication.

Name	Length	Fmt	Description
NODE	**1**	**xx**	**CAN node index** Index of the CAN node (0 ... N-1)
TXMSGB	**1**	**xx**	**Tx Message Buffer** Tx CAN Message Buffer number. Must be previously configured by TSW_18.
RXMSGB	**1**	**xx**	**Rx Message Buffer** Rx CAN Message Buffer number. Must be previously configured by TSW_18.
DATA	**1**	**xx..**	**Transmit Data** Data to be transmitted. Max length: 250 Byte **Note** Maximum length may be restricted by communication to your ECU

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	74	xx	xx	01	...

Parameters

Command response data

6	+0
(1*)	**DATA**

Details

Command response data
Command response data.

Name	Length	Fmt	Description
DATA	**1**	**xx..**	**Receive Data** Received Data sent by target device. **Note** Maximum length may be restricted by communication to your ECU

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)

Code	Description
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

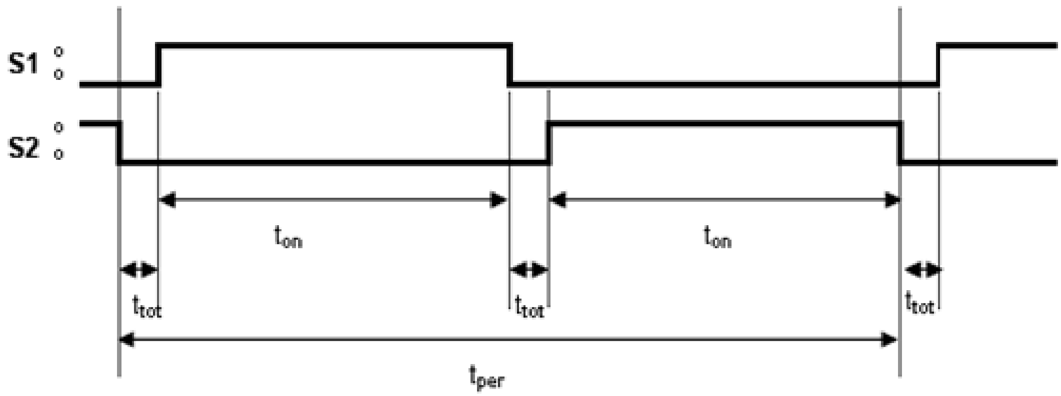
Code	Description
0x00	**TSW execution was successful**
0x01	**TSW - KWP: UART channel not supported**
0x02	**TSW - KWP: Max message size exceeded**
0x03	**TSW - KWP: Invalid / not supported KWP Header received.**
0x04	**TSW - KWP: Invalid Checksum received.**
0x05	**TSW - KWP: Max response message size exceeded**
0x10	**TSW - TP: Tx error**
0x11	**TSW - TP: No Response after complete Msg sent (P2 timeout)**
0x12	**TSW - TP: No Flow Control received (N_Bs)**
0x13	**TSW - TP: No Consequtive Frame received (N_Cr)**
0x15	**TSW - TP: invalid STmin value received in FC**
0x16	**TSW - TP: invalid Sequence number in CF**
0x17	**TSW - TP: FC expected but not received**
0x18	**TSW - TP: Not valid TP message received**
0x1A	**TSW - TP: Receiver reports Overflow**
0x1B	**TSW - TP: Receiver send to many WAIT FC**
0x1E	**TSW - TP: Max message size exceeded**
0x1F	**TSW - TP: Rx Buffer size exceeded**
0xEF	**HAL - Invalid Parameter**
0xEE	**HAL - No message received in specified time**
0xED	**HAL - BootCtrl version not supported**

TSW_75 - Forward converter test

Test step description

This test step is used to test the forward converter (FWC).

Figure 1: Structure of FWC signals



NOTE: Please take into account, that all port and timer ressource management has to be done by the user. NO automatic ressource handling will be done with e.g. TSW_05!

Test step commands

.

01 - FWC signal pair channel configuration

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	75	01	01	...

Parameters

Parameter set for configuaration channel FWC request

5	+0	+1	+2	+3	+4	+5
(1*)	**CID**	**PORTS1**	**PINS1**	**PORTS2**	**PINS2**	**TMRPER**

...	+7	+9	+11	+13
(1*)	**TMRESS1**	**TMRESS2**	**HALFPER**	**T_TOT**

Details

Parameter set for configuaration channel FWC request

Name	Length	Fmt	Description
CID	**1**	**xx**	**Signal pair channel ID (0..1)** ID of signal pair channel to be configured
PORTS1	**1**	**xx**	**FWC S1 signal port**

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>Port number for S1 signal of FWC</td></tr>
<tr><td>PINS1</td><td>1</td><td>xx</td><td>FWC S1 signal pin
Pin number for S1 signal of FWC</td></tr>
<tr><td>PORTS2</td><td>1</td><td>xx</td><td>FWC S2 signal port
Port number for S2 signal of FWC</td></tr>
<tr><td>PINS2</td><td>1</td><td>xx</td><td>FWC S2 signal pin
Pin number for S2 signal of FWC</td></tr>
<tr><td>TMRPER</td><td>2</td><td>xxxx</td><td>Desired timer resource for signal period of FWC
Timer resource for generation of signal period of FWC.

Note: As only ATOMs are suitable here, parameter timer module must be like 0x1x.

Example: TIMRES = 0x1105: --> Select ATOM1_5.

<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>15:8</td><td>Timer module (0x1x: ATOM0..x)</td></tr><tr><td>7:0</td><td>Timer channel (0-7)</td></tr></table></td></tr>
<tr><td>TMRESS1</td><td>2</td><td>xxxx</td><td>Desired timer resource for S1 signal of FWC
Timer resource for generation of FWC S1 signal.

Note 1: As only ATOMs are suitable here, parameter timer module must be like 0x1x.

Note 2: Timer channel used here must be a consecutive successor of the timer channel used for TMRPER. Recommendation: It should also be within the same timer module, or in timer module direct after TMRPER.

Note 3: Timer must be connectable to port/pin PORTS1/PINS1" => See HW resources!

Example: TIMRES = 0x1106: --> Select ATOM1_6.

<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>15:8</td><td>Timer module (0x1x: ATOM0..x)</td></tr><tr><td>7:0</td><td>Timer channel (0-7)</td></tr></table></td></tr>
<tr><td>TMRESS2</td><td>2</td><td>xxxx</td><td>Desired timer resource for S2 signal of FWC
Timer resource for generation of FWC S2 signal.

Note: As only ATOMs are suitable here, parameter timer module must be like 0x1x.

Note 2: Timer channel used here must be a consecutive successor of the timer channel used for TMRPER. Recommendation: It should also be within the same timer module, or in timer module direct after TMRPER.</td></tr>
</table>

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>Note 3: Timer channel used here must be within the same timer module used for TMRESS1.

Note 4: Timer must be connectable to port/pin PORTS2/PINS2 => See HW resources!

Example: TIMRES = 0x1107: --> Select ATOM1_7.

<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>15:8</td><td>Timer module (0x1x: ATOM0..x)</td></tr><tr><td>7:0</td><td>Timer channel (0-7)</td></tr></table></td></tr>
<tr><td>HALFPER</td><td>2</td><td>xxxx</td><td>half period (t(per)/2) of FWC signal

Half period (t(per)/2) of the S1 and S2 signal

Valid: 750ns - 655µs

Resolution: Dev1: 25ns, others: 12.5ns

Note: Larger or smaller values will be limited to these borders.

Format: RVVV where R=Resolution, V=Value (e.g. R=E: 10 ms, R=D: 1 ms).</td></tr>
<tr><td>T_TOT</td><td>2</td><td>xxxx</td><td>deadtime t(tot) of FWC signal

Deadtime t(tot) between S1 and S2 signal

Valid: 750ns - t(per)/2

Resolution: Dev1: 25ns, others: 12.5ns

Note 1: Larger or smaller values will be limited to these borders.

Note 2: Values >= HALFPER are not possible and leads to an error.

ATTENTION: Always take care of the limititations of HW! Too small values for t(tot) can cause irreversible damage to the used HW!

Format: RVVV where R=Resolution, V=Value (e.g. R=E: 10 ms, R=D: 1 ms).</td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	75	xx	xx	01

.

02 - Stop FWC channel

This command stops the FWC signals for the specified signal pair channel.

"Command 0x1: FWC signal pair channel configuration" must be called before for specified signal pair channel.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	75	02	01	...

Parameters

Parameter set for stop FWC signal pair channel request.

5	+0
(1*)	**CID**

Details

Parameter set for stop FWC signal pair channel request.

Name	Length	Fmt	Description
CID	**1**	**xx**	**Signal pair channel ID (0..1)** ID of signal pair channel to be stopped

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	75	xx	xx	01

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)

Code	Description
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**TSW: Combination of Port/Pin and Timerresource unplausible**
0x02	**TSW: Normal Outputs (TOM) are not supported**
0x03	**TSW: Given channel unplausible**
0x04	**TSW: Given value for t(tot) is >= t(per)/2**

TSW_76 - Measurement of current characteristics for current sensor (I-VGL)

Test step description

Date: 08.01.2016

This test step is used for measurement of current characteristics for current sensor (I-VGL)

Note: The used pins have to be initialized with TSW_04. Used ADCs have to be initialized with TSW_01 Command 0x1 and 0x2.

Test step commands

.

01 - IVGL start

This command configures and starts the IVGL measurement procedure.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	76	01	01	...

Parameters

Parameter set for IVGL start request

5	+0	+1	+2	+4	+6
(1*)	**PORT**	**PIN**	**TPREON**	**TSAMPLE**	**NSA**

Parameter set for ADC configuration. Up to 6 ADCs [0..5] are possible.

12	+0	+1
(1..n*)	**ADCCHA**	**ADCMUX**

Details

Parameter set for IVGL start request

Name	Length	Fmt	Description
PORT	**1**	**xx**	**IVGL digital port** Port number for digital signal of IVGL
PIN	**1**	**xx**	**IVGL digital pin** Pin number for digital signal of IVGL
TPREON	**2**	**xxxx**	**On time of digital port before start of ADC measurement**

Name	Length	Fmt	Description
			Valid: 10µs - 1500µs Resolution: 1µs Note: Larger or smaller values will be limited to these borders. Format: RVVV where R=Resolution, V=Value (e.g. R=E: 10 ms, R=D: 1 ms).
TSAMPLE	**2**	**xxxx**	**Time between samples** Valid: 8µs - 15µs Resolution: 1µs Note: Larger or smaller values will be limited to these borders. Format: RVVV where R=Resolution, V=Value (e.g. R=E: 10 ms, R=D: 1 ms).
NSA	**1**	**xx**	**Number of samples per ADC** Max. Number of samples is 122 (per ADC)

Parameter set for ADC configuration. Up to 6 ADCs [0..5] are possible.

Name	Length	Fmt	Description
ADCCHA	**1**	**xx**	**ADC : Index of the channel** Description of parameter see command 0x10 "start conversion and read result" of TSW_01. Note: For analog channel mapping greater then AN47, please take a look to TSW_01 (Table 1)!!!
ADCMUX	**1**	**xx**	**ADC: Channel of the external multiplexer** Description of parameter see command 0x10 "start conversion and read result" of TSW_01.

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	76	xx	xx	01

.

02 - IVGL get measurement ADC

This command gets IVGL measurement results from ADCs.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	76	02	01	...

Parameters

Parameter set for IVGL ADC measurement request

5	+0
(1*)	**ADCNUM**

Details

Parameter set for IVGL ADC measurement request

Name	Length	Fmt	Description
ADCNUM	**1**	**xx**	**Number of ADC which should be read (0..5)**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	76	xx	xx	01	...

Parameters

Response set for IVGL ADC measurement request

6	+0
(1..n*)	**ADCRES**

Details

Response set for IVGL ADC measurement request

Name	Length	Fmt	Description
ADCRES	**2**	**xxxx**	**ADC: Measurement results** ADC measurement result values. Number of measurements as specified by command 0x1 "NSA".

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**

Code	Description
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**TSW: Too many ADC channels given**
0x02	**TSW: Given TSAMPLE is not sufficient (ADC conversion takes longer than TSAMPLE; this may caused by unsuccessful HSM jump to RAM because of broken infoblock chain)**

TSW_77 - FlexIn L9966 sequencer

Test step description

This test step is used to configure and run the FlexIn L9966 sequencer.

This test step is only feasible for L9966 Bx-Step.

Note 1: All used controller pins have to be initialized with TSW_04.
Note 2: All SPI initializations for L9966 must be done with TSW_02 command 0x1 and 0x2.

Test step commands

.

01 - FlexIn sequencer SPI initialization

This function allows to configure the FlexIn sequencer SPI.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	77	01	01	...

Parameters

Parameter set for FlexIn sequencer SPI initialization request

5	+0	+1
(1*)	**DEVID**	**SPIADR**

Parameter set for FlexIn sequencer SPI port initialization request

7	+0	+1	+2	+3	+4	+5
(1*)	**PORTMT**	**PINMT**	**PORTMR**	**PINMR**	**PORTCK**	**PINCK**

...	+6	+7
(1*)	**PORTSL**	**PINSL**

Details

Parameter set for FlexIn sequencer SPI initialization request

Name	Length	Fmt	Description
DEVID	**1**	**xx**	**Virtual device ID (0..2)** Virtual ID of device to be configured.
SPIADR	**1**	**xx**	**SPI address multiplexing** SPI address multiplexing index for FlexIn. **Bit(s)** | **Description** **7:1** | **Not used. Should be 0.**

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td><table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>0</td><td>SPI address multiplexing: Level of the CTRL_CFG pin.</td></tr></table></td></tr>
</table>

Parameter set for FlexIn sequencer SPI port initialization request

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal.
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number within PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal. If MSB bit is set, emulated SLSO is used with given port (only for IFX and JDP).
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal. If MSB bit is set, emulated SLSO is used with given pin (only for IFX and JDP).

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	77	xx	xx	01

.

02 - FlexIn register read

Note: Required previously successful command execution: FlexIn sequencer SPI initialization.

This function allows to read registers from FlexIn

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	77	02	01	...

Parameters

Parameter set for FlexIn register read request

5	+0	+1
(1..n*)	**DEVID**	**REGADR**

Details

Parameter set for FlexIn register read request

Name	Length	Fmt	Description
DEVID	**1**	**xx**	**Virtual device ID (0..2)** Virtual ID of device to read register from.
REGADR	**1**	**xx**	**FlexIn register address**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	77	xx	xx	01	...

Parameters

Response set for FlexIn register read request

6	+0
(1..n*)	**REGVAL**

Details

Response set for FlexIn register read request

Name	Length	Fmt	Description
REGVAL	**2**	**xxxx**	**FlexIn register value**

.

03 - FlexIn register write

Note: Required previously successful command execution: FlexIn sequencer SPI initialization.

This function allows to write registers of FlexIn.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	77	03	01	...

Parameters

Parameter set for FlexIn register write request

5	+0	+1	+2	+4
(1..n*)	**DEVID**	**REGADR**	**REGVAL**	**REGMASK**

Details

Parameter set for FlexIn register write request

Name	Length	Fmt	Description
DEVID	**1**	**xx**	**Virtual device ID (0..2)** Virtual ID of device to write register.
REGADR	**1**	**xx**	**FlexIn register address**
REGVAL	**2**	**xxxx**	**FlexIn register value**
REGMASK	**2**	**xxxx**	**FlexIn register mask** Set bit position to 1, if bit should be changed.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	77	xx	xx	01	...

Parameters

Response set for FlexIn register write request

6	+0
(1..n*)	**REGVAL**

Details

Response set for FlexIn register write request

Name	Length	Fmt	Description
REGVAL	**2**	**xxxx**	**FlexIn register value**

.

05 - FlexIn sequencer initialization

Note: Required previously successful command execution: FlexIn sequencer SPI initialization.

This function allows to configure the FlexIn sequencer.

All registers of the sequencer (which should be configured) must be configured with one call of this command. Therefore repeat parameter SEQREG and SEQVAL as often as required!

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	77	05	01	...

Parameters

Parameter set for FlexIn sequencer initialization request

5	+0
(1*)	**DEVID**

Parameter set for FlexIn sequencer command register initialization request

6	+0	+1
(1..n*)	**SEQREG**	**SEQVAL**

Details

Parameter set for FlexIn sequencer initialization request

Name	Length	Fmt	Description
DEVID	**1**	**xx**	**Virtual device ID (0..2)** Virtual ID of device to be configured.

Parameter set for FlexIn sequencer command register initialization request

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>SEQREG</td><td>1</td><td>xx</td><td>Number of sequencer command register x (x =1..15).
Number of sequencer command register x which shall be initialised with user data.</td></tr>
<tr><td>SEQVAL</td><td>2</td><td>xxxx</td><td>SQNCR_CMD_x: Sequencer command register x.
Value for sequencer control register SQNCR_CMD_x.

See documentation of L9966 for details!

Table 1. Meaning of PUP_DIV values:
<table>
<tr><th>PUP_DIV</th><th>Resistance measeurement</th><th>Voltage measurement</th></tr>
<tr><td>00b</td><td>no pullup</td><td>5V full range</td></tr>
<tr><td>01b</td><td>pullup RR1</td><td>20V full range</td></tr>
<tr><td>10b</td><td>pullup RR2</td><td>40V full range</td></tr>
<tr><td>11b</td><td>pullup RR3</td><td>1.25V full range</td></tr>
</table>
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>15:7</td><td>Not used. Should be 0.</td></tr>
<tr><td>6:3</td><td>NXT_PC</td></tr>
<tr><td>2:1</td><td>PUP_DIV
See description table "Meaning of PUP_DIV values" above.</td></tr>
<tr><td>0</td><td>R_VOLT_MEAS_SELECT
1: Voltage measurement, 0: Resistance measurement</td></tr>
</table>
</td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	77	xx	xx	01

.

10 - FlexIn sequencer control

Note: Required previously successful command execution: FlexIn sequencer initialization.

This function allows to control the FlexIn sequencer.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	77	10	01	...

Parameters

Parameter set for FlexIn sequencer control register request

5	+0	+1
(1*)	**DEVID**	**SEQCTRL**

Details

Parameter set for FlexIn sequencer control register request

Name	Length	Fmt	Description
DEVID	**1**	**xx**	**Virtual device ID (0..2)** Virtual ID of device to be configured.
SEQCTRL	**2**	**xxxx**	**SQNCR_CTRL: Sequencer control register.** Sequencer control register SQNCR_CTRL. See documentation of L9966 for details!

Bit(s)	Description
15	**Not used. Should be 0.**
14	**EU2_SYNC_EN**
13	**Not used. Should be 0.**
12:9	**INIT_PC_EU2**
8	**EU2_EN**
7	**SYNC_COPY_CMD_EN**
6	**EU1_SYNC_EN**
5	**Not used. Should be 0.**
4:1	**INIT_PC_EU1**
0	**EU1_EN**

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	77	xx	xx	01

.

20 - FlexIn sequencer read result

Note: Required previously successful command execution: FlexIn sequencer control.

This function allows to read the results for the FlexIn sequencer.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	77	20	01	...

Parameters

Parameter set for FlexIn sequencer read result request

5	+0
(1*)	**DEVID**

Details

Parameter set for FlexIn sequencer read result request

Name	Length	Fmt	Description
DEVID	**1**	**xx**	**Virtual device ID (0..2)** Virtual ID of device to be configured.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	77	xx	xx	01	...

Parameters

Response set for FlexIn sequencer read result request

6	+0
(1..n*)	**SEQRSLT**

Details

Response set for FlexIn sequencer read result request

Name	Length	Fmt	Description
SEQRSLT	**2**	**xxxx**	**SQNCR_RESULT: Sequencer result register.** Sequencer result register SQNCR_RESULT.

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
Note: Due to SPI burst read, number of result values may differ from configured sequencer channels. Also not configured channels which are between DIG_IN_STAT and maximum configured ADC channels will be given in the result.
Channel order is DIG_IN_STAT, ADC channel 1, ADC channel 2, ..., ADC channel 12, UBSW(13), VI5V(14), VIX (15).
If EU2_SYNC_EN or EU1_SYNC_EN and SYNC_COPY_CMD_EN is set with command "FlexIn sequencer control", copy of shadow buffer to result register is done automatically by L9966. Otherwise register SQNCR_RSLT_COPY_CMD is read by command "FlexIn sequencer read result" to ensure copy shadow register to result register.
If SQNCR_CMD_x is switched to "Resistence measurement": ADC_RESULT: Rpulldown value divided by Rpullup value, Bit[14:11] integer part, Bit[10:0] fractional part. To get result as rational value the complete register content [14:0] in decimal value can be divided by 2048.
See documentation of L9966 for details!
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>15</td><td>Not used. Will be 0.</td></tr>
<tr><td>14</td><td>Channel 15 source level
For DIG_IN_STAT result.</td></tr>
<tr><td>14</td><td>ADC_RESULT Bit 14
If SQNCR_CMD_x is switched to "Resistence measurement"</td></tr>
<tr><td>14</td><td>NEW_RESULT_FLAG
If SQNCR_CMD_x is switched to "Voltage measurement"</td></tr>
<tr><td>13:12</td><td>Channel 14:13 source level
For DIG_IN_STAT result.</td></tr>
<tr><td>13:12</td><td>ADC_RESULT Bit 13:12
If SQNCR_CMD_x is switched to "Resistence measurement"</td></tr>
<tr><td>13:12</td><td>Not used. Will be 0.
If SQNCR_CMD_x is switched to "Voltage measurement"</td></tr>
<tr><td>11:0</td><td>Channel 12:1 source level
For DIG_IN_STAT result.</td></tr>
<tr><td>11:0</td><td>ADC_RESULT Bit 11:0</td></tr>
</table>
</td></tr>
</table>

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**TSW: SPI communication error.**

TSW_90 - CY32x - Monitoring Service

Test step description

The TSW_90 enables or disables the monitoring service of CY32x and thus enables or disables the WDA signal.

The TSW_90 supports CY326, CY327 and CY328.

Before using TSW_90 the SPI connection must be configured using TSW_02 and TSW_04.

The normal interrupt rate is approximately one interrupt per milliseconds. Because of flash test, there is a different behaviour in AST environment. The interrupt rate for AST is approximately one interrupt per 100 milliseconds.

Date: 04.03.2016

Created 18.01.2017 09:12.

Test step commands

.

00 - Initiate Service

The command 0x00 starts the service of the monitoring module in CY32x

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	90	00	01	...

Parameters

Parameters to enable the monitoring service handler

5 (1*)	+0	+1	+2	+3	+4	+5
	PORTMT	**PINMT**	**PORTMR**	**PINMR**	**PORTCK**	**PINCK**

... (1*)	+6	+7
	PORTSL	**PINSL**

Details

Parameters to enable the monitoring service handler

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal

Name	Length	Fmt	Description
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	90	xx	xx	01	...

Parameters

Response of command 0x00

6	+0	+3
(1*)	**WIN**	**ID**

Details

Response of command 0x00

Name	Length	Fmt	Description
WIN	**3**	**xxxxxx**	**Time window length** This value shows the measured length of the short monitoring time window. That is the duration of 2 errorcounter ticks while the monitoring response window period is set to the smallest interval. The nominal value is 1600µs + 12800µs = 14400µs for this value.
ID	**2**	**xxxx**	**Chip Identification/Revision** The upper byte of the value show the Chip-ID, the lower byte the revision number.

.

01 - Shutdown Service

The command 0x01 immediately shuts down the WDA service, thus disabling the WDA signal.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	90	01	01

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	90	xx	xx	01

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)

Code	Description
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**TSW: Invalid SPI RX frame.**
0x02	**TSW: Sync timeout.**
0x03	**TSW: Unknown chip ID.**
0xFF	**HAL busy**
0xFE	**Generic HAL error**
0xFD	**HAL not compatible to current CPU**
0xFC	**Invalid params**
0xFB	**HAL module not implemented**
0xFA	**HAL function not implemented**
0xF9	**HAL timeout**

TSW_91 - CJ721 - Monitoring Service

Test step description

TSW 91 enables or disables the monitoring service of CJ721(Connected via MSC) and thus enables or disables the WDA signal.

Before using TSW_91 the MSC connection must be configured using TSW_03 and TSW_04.

The normal interrupt rate is approximately one interrupt per milliseconds. Because of flash test, there is a different behaviour in AST environment. The interrupt rate for AST is approximately one interrupt per 100 milliseconds.

Created 18.01.2017 09:12.

Test step commands

.

00 - Initiate Service

The command 0x00 starts the service of the monitoring module in CJ721

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	91	00	01	...

Parameters

Parameters to enable the monitoring service handler

5	+0	+1	+2	+3	+4	+5
(1*)	**POFCLP**	**PIFCLP**	**POFCLN**	**PIFCLN**	**POEN**	**PIEN**

...	+6	+7	+8	+9	+10	+11
(1*)	**POSOP**	**PISOP**	**POSON**	**PISON**	**POSDI**	**PISDI**

Details

Parameters to enable the monitoring service handler

Name	Length	Fmt	Description
POFCLP	**1**	**xx**	**Port number of FCLP signal** Port number of FCLP (Clock positiv) signal
PIFCLP	**1**	**xx**	**Pin number of FCLP signal** Pin number within POFCLP of FCLP (Clock positiv) signal
POFCLN	**1**	**xx**	**Port number of FCLN signal** Port number of FCLN (Clock negativ) signal. Set 0xFF if single ended is used.
PIFCLN	**1**	**xx**	**Pin number of FCLN signal**

Name	Length	Fmt	Description
			Pin number within POFCLN of FCLN (Clock negativ) signal. Set 0xFF if single ended is used.
POEN	**1**	**xx**	**Port number of EN signal** Port number of EN (Enable) signal
PIEN	**1**	**xx**	**Pin number of EN signal** Pin number within POEN of EN (Enable) signal
POSOP	**1**	**xx**	**Port number of SOP signal** Port number of SOP (Serial Output Positiv) signal.
PISOP	**1**	**xx**	**Pin number of SOP signal** Pin number within POSOP of SOP (Serial Output Positiv) signal.
POSON	**1**	**xx**	**Port number of SON signal** Port number of SON (Serial Output Negativ) signal. Set 0xFF is single ended is used.
PISON	**1**	**xx**	**Pin number of SON signal** Pin number within POSON of SON (Serial Output Negativ) signal. Set 0xFF is single ended is used.
POSDI	**1**	**xx**	**Port number of SDI signal** Port number of SDI (Serial Data In) signal
PISDI	**1**	**xx**	**Pin number of SDI signal** Pin number within POSDI of SDI (Serial Data In) signal

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	91	xx	xx	01	...

Parameters

Response of command 0x00

6	+0	+3
(1*)	**WIN**	**ID**

Details

Response of command 0x00

Name	Length	Fmt	Description
WIN	**3**	**xxxxxx**	**Time window length** This value shows the measured length of the short monitoring time window. That is the duration of 2 errorcounter ticks while the monitoring response window period is set to the smallest interval. The nominal value is 1600µs + 12800µs = 14400µs for this value.
ID	**2**	**xxxx**	**Chip Identification/Revision** The upper byte of the value show the Chip-ID, the lower byte the revision number.

.

01 - Shutdown Service

The command 0x01 immediately shuts down the WDA service, thus disabling the WDA signal.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	91	01	01

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	91	xx	xx	01

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)

Code	Description
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**TSW: Invalid SPI RX frame.**
0x02	**TSW: Sync timeout.**
0xFF	**HAL busy**
0xFE	**Generic HAL error**
0xFD	**HAL not compatible to current CPU**
0xFC	**Invalid params**
0xFB	**HAL module not implemented**
0xFA	**HAL function not implemented**
0xF9	**HAL timeout**

TSW_92 - L9779 - Monitoring Service

Test step description

TSW 92 enables or disables the monitoring service of L9779 (Connected via MSC) and thus enables or disables the WDA signal. Similar to CJ721.

Before using TSW_92 the MSC connection must be configured using TSW_03 and TSW_04.

The normal interrupt rate is approximately one interrupt per milliseconds. Because of flash test, there is a different behaviour in AST environment. The interrupt rate for AST is approximately one interrupt per 100 milliseconds.

Created 18.01.2017 09:12.

Test step commands

.

00 - Initiate Service

The command 0x00 starts the service of the monitoring module in L9779

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	92	00	01	...

Parameters

Parameters to enable the monitoring service handler

5	+0	+1	+2	+3	+4	+5
(1*)	**POFCLP**	**PIFCLP**	**POFCLN**	**PIFCLN**	**POEN**	**PIEN**

...	+6	+7	+8	+9	+10	+11
(1*)	**POSOP**	**PISOP**	**POSON**	**PISON**	**POSDI**	**PISDI**

Details

Parameters to enable the monitoring service handler

Name	Length	Fmt	Description
POFCLP	**1**	**xx**	**Port number of FCLP signal** Port number of FCLP (Clock positiv) signal
PIFCLP	**1**	**xx**	**Pin number of FCLP signal** Pin number within POFCLP of FCLP (Clock positiv) signal
POFCLN	**1**	**xx**	**Port number of FCLN signal** Port number of FCLN (Clock negativ) signal. Set 0xFF if single ended is used.
PIFCLN	**1**	**xx**	**Pin number of FCLN signal**

Name	Length	Fmt	Description
			Pin number within POFCLN of FCLN (Clock negativ) signal. Set 0xFF if single ended is used.
POEN	**1**	**xx**	**Port number of EN signal** Port number of EN (Enable) signal
PIEN	**1**	**xx**	**Pin number of EN signal** Pin number within POEN of EN (Enable) signal
POSOP	**1**	**xx**	**Port number of SOP signal** Port number of SOP (Serial Output Positiv) signal.
PISOP	**1**	**xx**	**Pin number of SOP signal** Pin number within POSOP of SOP (Serial Output Positiv) signal.
POSON	**1**	**xx**	**Port number of SON signal** Port number of SON (Serial Output Negativ) signal. Set 0xFF is single ended is used.
PISON	**1**	**xx**	**Pin number of SON signal** Pin number within POSON of SON (Serial Output Negativ) signal. Set 0xFF is single ended is used.
POSDI	**1**	**xx**	**Port number of SDI signal** Port number of SDI (Serial Data In) signal
PISDI	**1**	**xx**	**Pin number of SDI signal** Pin number within POSDI of SDI (Serial Data In) signal

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	92	xx	xx	01	...

Parameters

Response of command 0x00

6	+0	+3
(1*)	**WIN**	**ID**

Details

Response of command 0x00

Name	Length	Fmt	Description
WIN	**3**	**xxxxxx**	**Time window length** This value shows the measured length of the short monitoring time window. That is the duration of 2 errorcounter ticks while the monitoring response window period is set to the smallest interval. The nominal value is 1600µs + 12800µs = 14400µs for this value.
ID	**2**	**xxxx**	**Chip Identification/Revision** The upper byte of the value show the Chip-ID, the lower byte the revision number.

.

01 - Shutdown Service

The command 0x01 immediately shuts down the WDA service, thus disabling the WDA signal.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	92	01	01

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	92	xx	xx	01

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)

Code	Description
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**TSW Question/answer watchdog sync failed.**
0x02	**TSW Question/answer watchdog sync timeout.**
0xFF	**HAL busy**
0xFE	**Generic HAL error**
0xFD	**HAL not compatible to current CPU**
0xFC	**Invalid params**
0xFB	**HAL module not implemented**
0xFA	**HAL function not implemented**
0xF9	**HAL timeout**

TSW_94 - Periodic toggling GPIO port/pin

Test step description

This TSW module provides period toggling GPIO port/pin. This concerns a CS in case of a chip on a SPI bus needs toggling on the CS also while no SPI frame is sent.

Created 18.01.2017 09:12.

Test step commands

.

00 - Stop periodic toggling

Stop periodic toggling for CS pins of a SPI bus. The complete pin set must be specified.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	94	00	01	...

Parameters

Parameter set for periodic toggling port/pin supported by cyclic interrupt

5	+0	+1	+2	+3	+4	+5
(1..n*)	**PORT_MT**	**PIN_MT**	**PORT_MR**	**PIN_MR**	**PORT_CK**	**PIN_CK**

...	+6	+7	+8	+9
(1..n*)	**PORT_SL**	**PIN_SL**	**STATE_T**	**TIME_T**

Details

Parameter set for periodic toggling port/pin supported by cyclic interrupt

Name	Length	Fmt	Description
PORT_MT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PIN_MT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORT_MR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PIN_MR	**1**	**xx**	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORT_CK	**1**	**xx**	**Port number of SCK signal**

Name	Length	Fmt	Description
			Port number of SCK (Serial Clock) signal
PIN_CK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORT_SL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal. If MSB bit is set, emulated SLSO is used with given port (only for IFX and JDP). Signal is toggling periodicly
PIN_SL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal. If MSB bit is set, emulated SLSO is used with given pin (only for IFX and JDP). Signal is toggling periodicly
STATE_T	**1**	**xx**	**State of toggle signal inactive** Level of toggle signal during state inactive (e.g. in case of SPI communication) **Bit(s)** – **Description** **7:1** – **-** **0** – **State of toggle signal inactive** **0_B:** pin low **1_B:** pin high
TIME_T	**2**	**xxxx**	**Toggle time** Time after toggle signal has to change state periodicly Format: RVVV where R=Resolution, V=Value

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	94	xx	xx	01	...

Parameters

Number of toggling events responded

6	+0
(1*)	**NUMTEV**

Details

Number of toggling events responded

Name	Length	Fmt	Description
NUMTEV	**4**	**xxxxxxxx**	**Number of toggle events**

.

01 - Start periodic toggling

Set periodic toggling for CS pins of a SPI bus. The complete pin set must be specified. Timing parameters must be the same.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	94	01	01	...

Parameters

Parameter set for periodic toggling port/pin supported by cyclic interrupt

5	+0	+1	+2	+3	+4	+5
(1..n*)	**PORT_MT**	**PIN_MT**	**PORT_MR**	**PIN_MR**	**PORT_CK**	**PIN_CK**

...	+6	+7	+8	+9
(1..n*)	**PORT_SL**	**PIN_SL**	**STATE_T**	**TIME_T**

Details

Parameter set for periodic toggling port/pin supported by cyclic interrupt

Name	Length	Fmt	Description
PORT_MT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PIN_MT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORT_MR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PIN_MR	**1**	**xx**	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORT_CK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PIN_CK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORT_SL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal. If MSB bit is set, emulated SLSO is used with given port (only for IFX and JDP). Signal is toggling periodicly
PIN_SL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal. If MSB bit is set, emulated SLSO is used with given pin (only for IFX and JDP). Signal is toggling periodicly
STATE_T	**1**	**xx**	**State of toggle signal inactive** Level of toggle signal during state inactive (e.g. in case of SPI communication)

Name	Length	Fmt	Description
			Bit(s) / **Description** **7:1** / **-** **0** / **State of toggle signal inactive** **0_B:** pin low **1_B:** pin high
TIME_T	**2**	**xxxx**	**Toggle time** Time after toggle signal has to change state periodicly Format: RVVV where R=Resolution, V=Value

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	94	xx	xx	01	...

Parameters

Number of toggling events responded

6 (1*)	+0
	NUMTEV

Details

Number of toggling events responded

Name	Length	Fmt	Description
NUMTEV	**4**	**xxxxxxxx**	**Number of toggle events**

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)

Code	Description
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0xFF	**HAL busy**
0xFE	**Generic HAL error**
0xFD	**HAL not compatible to current CPU**
0xFC	**Invalid params**
0xFB	**HAL module not implemented**
0xFA	**HAL function not implemented**
0xF9	**HAL timeout**
0xEF	**No SPI controller on given ports found**
0xEE	**SPI function or option not supported**
0xED	**SPI baudrate not adjustable**
0xEC	**SPI HAL parameter error**
0xEB	**SPI Too many toggling SLSOs**

TSW_97 - Analog/Digital dynamic test

Test step description

Date: 24.08.2016

This test step is used to measure analog/digital dynamic response to a transient on a specific processor pin, thus allowing to check for proper circuit board population as well as for resistor and capacitor value tolerances.

The user has to specify a combination of a stimulus port (PS_PIN) which will toggle it's state during measurment process and an input port (ME_PIN) which will be used to measure the overall circuit response to the created transient.

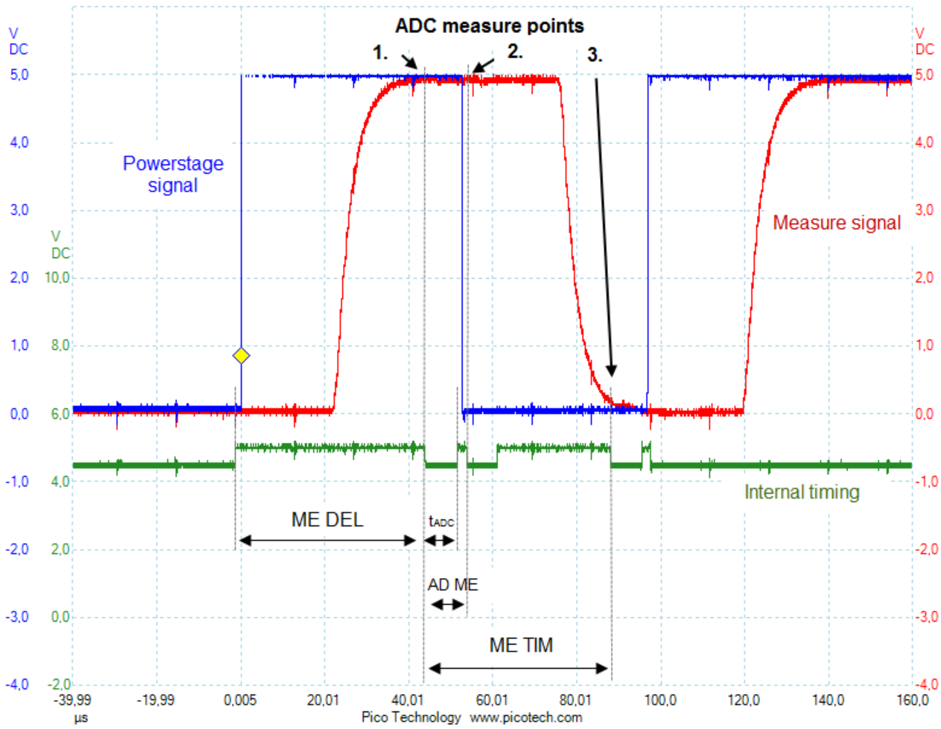
Powerstage pin (PS_PIN) must always be conneted to measure pin (ME_PIN).

Test step commands

.

01 - Analog dynamic test

This command is used to perform a analog dynamic measurement. Essentially this test allows to measure 3 voltage points of the charging (or discharging) e-curve thus allowing tolerance checks.



tADC: Time of conversion (JDP: 8us, IFX: 2,5us). This is the minimum time, which can be used for the first measurement point.

Note: The measurement points show only the beginning of ADC measurement on the image, not the actual measuring point. It will take only a few us later, depends on the used CPU and Device.

Because of the software runtime the measure points, which are defined with the given timing parameters get internally an offset (till to ~2,5us). These Offsets depends on the used controller and has to be considered.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	97	01	01	...

Parameters

Parameter set for selecting a MSC or SPI controller by port/pin numbers (PO=PORT, PI=PIN). Interpretation of the parameters (MSC or SPI) depends on the value of PS_PRT parameter.

5	+0	+1	+2	+3	+4	+5
(1..n*)	**PO_FCLP**	**PI_FCLP**	**PO_FCLN**	**PI_FCLN**	**PO_EN**	**PI_EN**

...	+6	+7	+8	+9	+10	+11
(1..n*)	**PO_SOP**	**PI_SOP**	**PO_SON**	**PI_SON**	**PO_SDI**	**PI_SDI**

Parameter set for analog dynamic test

17	+0	+2	+4	+5	+6	+7
(1..n*)	**ME_DEL**	**ME_TIM**	**PS_PRT**	**PS_PIN**	**CONF**	**ME_PIN**

...	+8	+9	+10
(1..n*)	**MUX**	**AD_ME**	**AD_SAM**

Details

Parameter set for selecting a MSC or SPI controller by port/pin numbers (PO=PORT, PI=PIN). Interpretation of the parameters (MSC or SPI) depends on the value of PS_PRT parameter.

If PS_PRT is set to a value between 0x00 and 0x9F, these parameters may be set to arbitrary values.

If PS_PRT is set to a value between 0xA0 and 0xAF, these parameters must be set to a valid MSC configuration. PS_PIN values higher than 15 will be send out in the high frame.

If PS_PRT is set to a value between 0xB0 and 0xBF, these parameters must be set to a valid SPI configuration.

Table 1. Examples

Command	MSC pins	Times	PS pin	Conf
Digital pin			P00.8	AN3
80970101	0000 0000 0000 0000 0000 0000	A005 D028	0008	00 0300 1F 0014
MSC bit			MSC0 bit 0	AN5
80970101	0D01 0D00 0B0B 0D03 0D02 0A02	A005 D028	A000	00 0500 1F 0032

Name	Length	Fmt	Description
PO_FCLP	**1**	**xx**	**Port number of FCLP signal** Port number of FCLP (Clock Positive) signal when using MSCPort number of MTSR (Master Transmit) signal when using SPI
PI_FCLP	**1**	**xx**	**Pin number of FCLP signal**

Name	Length	Fmt	Description
			Pin number within POFCLP of FCLP (Clock Positive) signal when using MSCPin number within PORTMT of MTSR (Master Transmit) signal when using SPI
PO_FCLN	**1**	**xx**	**Port number of FCLN signal** Port number of FCLN (Clock Negative) signal when using MSC. Set 0xFF if single ended is used.Port number of MRST (Master Receive) signal when using SPI
PI_FCLN	**1**	**xx**	**Pin number of FCLN signal** Pin number within POFCLN of FCLN (Clock Negative) signal when using MSC. Set 0xFF if single ended is used.Pin number withing PORTMR of MRST (Master Receive) signal when using SPI
PO_EN	**1**	**xx**	**Port number of EN signal** Port number of EN (Enable) signal when using MSCPort number of SCK (Serial Clock) signal when using SPI
PI_EN	**1**	**xx**	**Pin number of EN signal** Pin number within POEN of EN (Enable) signal when using MSCPin number within PORTCK of SCK (Serial Clock) signal when using SPI
PO_SOP	**1**	**xx**	**Port number of SOP signal** Port number of SOP (Serial Output Positive) signal when using MSC.Port number of SLSO (Slave Select) signal when using SPI
PI_SOP	**1**	**xx**	**Pin number of SOP signal** Pin number within POSOP of SOP (Serial Output Positive) signal when using MSC.Pin number within PORTSL of SLSO (Slave Select) signal when using SPI
PO_SON	**1**	**xx**	**Port number of SON signal** Port number of SON (Serial Output Negative) signal when using MSC. Set 0xFF if single ended is used.
PI_SON	**1**	**xx**	**Pin number of SON signal** Pin number within POSON of SON (Serial Output Negative) signal when using MSC. Set 0xFF if single ended is used.
PO_SDI	**1**	**xx**	**Port number of SDI signal** Port number of SDI (Serial Data In) signal when using MSC
PI_SDI	**1**	**xx**	**Pin number of SDI signal** Pin number within POSDI of SDI (Serial Data In) signal when using MSC

Parameter set for analog dynamic test

Name	Length	Fmt	Description
ME_DEL	**2**	**xxxx**	**Measure delay** Delay time before measurement starts. Used to get a static level on the measure pin.

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>Format: RVVV where R=Resolution, V=Value (e.g. R=0: Seconds, R=D: Milliseconds)</td></tr>
<tr><td>ME_TIM</td><td>2</td><td>xxxx</td><td>Measure timeout
Sets the end of measurement. This time defines the 3rd measure point.

Format: RVVV where R=Resolution, V=Value (e.g. R=0: Seconds, R=D: Milliseconds)</td></tr>
<tr><td>PS_PRT</td><td>1</td><td>xx</td><td>Powerstage Port
Powerstage port to switch the pin to be measured.

0x00 ... 0x9F: Digital port

Bigger 0x9F: MSC or SPI port

Note: The MSC must be initialized using TSW_03 routines. The SPI must be initialized using TSW_02 routines.

<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:4</td><td>Powerstage Type

A_H: MSC Powerstage
B_H: SPI Powerstage</td></tr>
<tr><td>3:0</td><td>Powerstage ASIC Type

0_H: All MSC ASICs
1_H: ATIC150
2_H: ATIC152
3_H: CJ960
4_H: TLE6232
5_H: TLE7244</td></tr>
</table>
</td></tr>
<tr><td>PS_PIN</td><td>1</td><td>xx</td><td>Powerstage Pin
0x00 ... 0x0F: GPIO Pin number of Powerstage port

If you use the MSC, the Powerstage Pin parameter describes the bit position (0 - 31) in the MSC register.

If you use the SPI with ATIC150, the Powerstage Pin parameter describes the pin position (0 - 15) of the ATIC.

If you use the SPI with ATIC152, the Powerstage Pin parameter describes the pin position (0 - 7 for P-P/LSD and 8 - 13 for LSD only) of the ATIC.

If you use the SPI with CJ960, the Powerstage Pin parameter describes the pin position (0 - 7) of the CJ960.</td></tr>
</table>

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>If you use the SPI with TLE6232, the Powerstage Pin parameter describes the pin position (0 - 6) of the TLE6232.

If you use the SPI with TLE7244, the Powerstage Pin parameter describes the pin position (0 - 7) of the TLE7244.</td></tr>
<tr><td>CONF</td><td>1</td><td>xx</td><td>Configuration
During measurement the power stage pin will be modified. Instead of reverting automatically to the state at the beginning of the test it is necessary to give the state that should be set after the measurement.
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7</td><td>Capacitor treatment

0_B: Capacitor is charged during measurement (PS_PIN switches ON)
1_B: Capacitor is discharged during measurement (PS_PIN switches OFF)</td></tr>
<tr><td>6</td><td>Reference voltage selection

0_B: reference taken from VREF-pin
1_B: reference taken from ADC channel 0</td></tr>
<tr><td>5:2</td><td>not used, must be set to 0</td></tr>
<tr><td>1</td><td>Powerstage is inverting

0_B: no
1_B: yes</td></tr>
<tr><td>0</td><td>Powerstage level after measurement

0_B: low
1_B: high</td></tr>
</table></td></tr>
<tr><td>ME_PIN</td><td>1</td><td>xx</td><td>ADC measurement pin
Analog input number (ANx).

Note: For analog channel mapping greater then AN47, please take a look to TSW_01 (Table 1)!!!</td></tr>
<tr><td>MUX</td><td>1</td><td>xx</td><td>External mux settings

IFX: Multiplexer channel (0-7).
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:4</td><td>Not used</td></tr>
</table></td></tr>
</table>

Name	Length	Fmt	Description
			Bit(s) / **Description** **3:0** / **Selection of Multiplexer input (AN0-AN7)**
AD_ME	**1**	**xx**	**ADC measurement point** Selects the 2nd intermediate measurement point of ADC with a % value. Resolution 0,5%/ LSB. The 2nd measure point is calculated: ADC measurement[%] * MeasureTimeout (= AD_ME * ME_TIM / 200). Note: A AD Conversion needs a little time. So it is not possible to do the intermediate measurement at 0%. It is done directly after the first conversion. 100 % will move the last conversion backward. Approximate value for the conversion time is 10 µs.
AD_SAM	**2**	**xxxx**	**ADC Sample time** Defines the sample time, resolution 10ns/LSB.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	97	xx	xx	01	...

Parameters

Response set of analog dynamic test request

6	+0	+2	+4
(1..n*)	**RES1**	**RES2**	**RES3**

Details

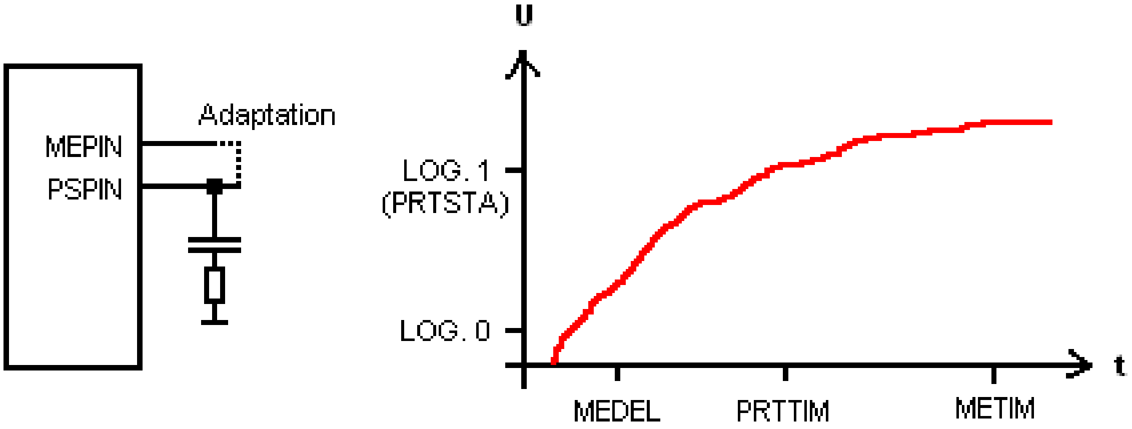
Response set of analog dynamic test request

Name	Length	Fmt	Description
RES1	**2**	**xxxx**	**Result 1** ADC conversion result at end of measure delay
RES2	**2**	**xxxx**	**Result 2** ADC conversion result at measure point (Measure timeout * ADC Measurement)
RES3	**2**	**xxxx**	**Result 3** ADC conversion result at measure timeout

.

02 - Digital dynamic test

This command is used to perform a digital dynamic measurement. Essentially this test measures the time delay between switching (toggling) the power stage (PSPIN) state and the following logic level change as seen on the input channel pin (MEPIN).



Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	97	02	01	...

Parameters

Parameter set for selecting a MSC or SPI controller by port/pin numbers (PO=PORT, PI=PIN). Interpretation of the parameters (MSC or SPI) depends on the value of PS_PRT parameter.

5	+0	+1	+2	+3	+4	+5
(1..n*)	**PO_FCLP**	**PI_FCLP**	**PO_FCLN**	**PI_FCLN**	**PO_EN**	**PI_EN**

...	+6	+7	+8	+9	+10	+11
(1..n*)	**PO_SOP**	**PI_SOP**	**PO_SON**	**PI_SON**	**PO_SDI**	**PI_SDI**

Parameter set for digital dynamic test

17	+0	+2	+4	+5	+6	+7
(1..n*)	**ME_DEL**	**ME_TIM**	**PS_PRT**	**PSPIN**	**CONF**	**ME_PRT**

...	+8
(1..n*)	**ME_PIN**

Details

Parameter set for selecting a MSC or SPI controller by port/pin numbers (PO=PORT, PI=PIN). Interpretation of the parameters (MSC or SPI) depends on the value of PS_PRT parameter.
If PS_PRT is set to a value between 0x00 and 0x9F, these parameters may be set to arbitrary values.

If PS_PRT is set to a value between 0xA0 and 0xAF, these parameters must be set to a valid MSC configuration. PS_PIN values higher than 15 will be send out in the high frame.

If PS_PRT is set to a value between 0xB0 and 0xBF, these parameters must be set to a valid SPI configuration.

Name	Length	Fmt	Description
PO_FCLP	**1**	**xx**	**Port number of FCLP signal** Port number of FCLP (Clock Positive) signal when using MSC Port number of MTSR (Master Transmit) signal when using SPI
PI_FCLP	**1**	**xx**	**Pin number of FCLP signal** Pin number within POFCLP of FCLP (Clock Positive) signal when using MSC Pin number within PORTMT of MTSR (Master Transmit) signal when using SPI
PO_FCLN	**1**	**xx**	**Port number of FCLN signal** Port number of FCLN (Clock Negative) signal when using MSC. Set 0xFF if single ended is used. Port number of MRST (Master Receive) signal when using SPI
PI_FCLN	**1**	**xx**	**Pin number of FCLN signal** Pin number within POFCLN of FCLN (Clock Negative) signal when using MSC. Set 0xFF if single ended is used. Pin number withing PORTMR of MRST (Master Receive) signal when using SPI
PO_EN	**1**	**xx**	**Port number of EN signal** Port number of EN (Enable) signal when using MSC Port number of SCK (Serial Clock) signal when using SPI
PI_EN	**1**	**xx**	**Pin number of EN signal** Pin number within POEN of EN (Enable) signal when using MSC Pin number within PORTCK of SCK (Serial Clock) signal when using SPI
PO_SOP	**1**	**xx**	**Port number of SOP signal** Port number of SOP (Serial Output Positive) signal when using MSC. Port number of SLSO (Slave Select) signal when using SPI
PI_SOP	**1**	**xx**	**Pin number of SOP signal** Pin number within POSOP of SOP (Serial Output Positive) signal when using MSC. Pin number within PORTSL of SLSO (Slave Select) signal when using SPI
PO_SON	**1**	**xx**	**Port number of SON signal**

Name	Length	Fmt	Description
			Port number of SON (Serial Output Negative) signal when using MSC. Set 0xFF if single ended is used.
PI_SON	**1**	**xx**	**Pin number of SON signal** Pin number within POSON of SON (Serial Output Negative) signal when using MSC. Set 0xFF if single ended is used.
PO_SDI	**1**	**xx**	**Port number of SDI signal** Port number of SDI (Serial Data In) signal when using MSC
PI_SDI	**1**	**xx**	**Pin number of SDI signal** Pin number within POSDI of SDI (Serial Data In) signal when using MSC

Parameter set for digital dynamic test

Name	Length	Fmt	Description
ME_DEL	**2**	**xxxx**	**Measure delay** Delay time before measurement starts. Used to get a static level on the measure pin.Format: RVVV where R=Resolution, V=Value (e.g. R=0: Seconds, R=D: Milliseconds)
ME_TIM	**2**	**xxxx**	**Measure timeout** Sets the end of measurement. This time defines the 3rd measure point.Format: RVVV where R=Resolution, V=Value (e.g. R=0: Seconds, R=D: Milliseconds)
PS_PRT	**1**	**xx**	**Powerstage Port** Powerstage port to switch the pin to be measured. 0x00 ... 0x9F: Digital port Bigger 0x9F: MSC or SPI port Note: The MSC must be initialized using TSW_03 routines. The SPI must be initialized using TSW_02 routines. **Bit(s)** – **Description** **7:4** – **Powerstage Type** **A_H:** MSC Powerstage **B_H:** SPI Powerstage **3:0** – **Powerstage ASIC Type** **0_H:** All MSC ASICs **1_H:** ATIC150 **2_H:** ATIC152 **3_H:** CJ960 **4_H:** TLE6232 **5_H:** TLE7244

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>PSPIN</td><td>1</td><td>xx</td><td>Powerstage Pin
0x00 ... 0x0F: GPIO Pin number of Powerstage port

If you use the MSC, the Powerstage Pin parameter describes the bit position (0 - 31) in the MSC register.

If you use the SPI with ATIC150, the Powerstage Pin parameter describes the pin position (0 - 15) of the ATIC.

If you use the SPI with ATIC152, the Powerstage Pin parameter describes the pin position (0 - 7 for P-P/LSD and 8 - 13 for LSD only) of the ATIC.

If you use the SPI with CJ960, the Powerstage Pin parameter describes the pin position (0 - 7) of the CJ960.

If you use the SPI with TLE6232, the Powerstage Pin parameter describes the pin position (0 - 6) of the TLE6232.

If you use the SPI with TLE7244, the Powerstage Pin parameter describes the pin position (0 - 7) of the TLE7244.</td></tr>
<tr><td>CONF</td><td>1</td><td>xx</td><td>Configuration
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7</td><td>Capacitor treatment

0_B: Capacitor is charged during measurement (PS_PIN switches ON)
1_B: Capacitor is discharged during measurement (PS_PIN switches OFF)</td></tr>
<tr><td>6:2</td><td>not used, must be set to 0</td></tr>
<tr><td>1</td><td>Powerstage is inverting

0_B: no
1_B: yes</td></tr>
<tr><td>0</td><td>Powerstage level after measurement

0_B: low
1_B: high</td></tr>
</table></td></tr>
<tr><td>ME_PRT</td><td>1</td><td>xx</td><td>Measurement Port
0x00 ... 0x9F: Digital port</td></tr>
<tr><td>ME_PIN</td><td>1</td><td>xx</td><td>Measurement Pin
Pin number for measurement</td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	97	xx	xx	01	...

Parameters

Response set of digital dynamic test request

6	+0	+2
(1..n*)	**PRT_TIM**	**PRT_STA**

Details

Response set of digital dynamic test request

Name	Length	Fmt	Description
PRT_TIM	**2**	**xxxx**	**Measured time** Point in time where pin state changed. Format: RVVV where R=Resolution, V=Value (e.g. R=0: Seconds, R=D: Milliseconds). The resolution value will be of the same order as given in the field METIM.
PRT_STA	**2**	**xxxx**	**Port state** Actual port state after measurement

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)

Code	Description
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**No SPI or MSC module found for given pins.**
0x02	**Error while writing to response buffer.**
0xA0	**Measurement timeout while waiting for the pin to settle. Short circuit?**
0xA1	**Pin state at start of the measurement not reached. Missing connection?**
0xA2	**Used powerstage channel not supported from the used ASIC.**
0xFF	**HAL busy**
0xFE	**Generic HAL error**
0xFD	**HAL not compatible to current CPU**
0xFC	**Invalid params**
0xFB	**HAL module not implemented**
0xFA	**HAL function not implemented**
0xF9	**HAL timeout**
0xEF	**HAL-GPIO: port not found**
0xEF	**HAL-ADC: channel not supported.**
0xEE	**HAL-GPIO: pin not found**
0xED	**HAL-GPIO: function not supported**
0xEC	**HAL-GPIO: port.pin / module not connectable**

TSW_99 - Data Upload

Test step description

Created: 18.01.2017 09:12.

This TSW allows uploading data from FLASH memory.

Use of CfiTerm's UPLOAD function

To use CfiTerm's UPLOAD function, at least CfiTerm V2.1.1 is required! Example:

UPLOAD

C:\UploadedFile.hex, 08FC0000, 4000, 240, 80990301 ######## %%, C099000001

.

Test step commands

.

01 - Init Module Upload

This command prepares the flash upload of a specified memory module. For PFLASH the overlay area is read. For normal PFLASH area use command 3.

For HSM FLASH addresses are only informational. HSM FLASH in exclusive mode only can be downloaded complete by module. Parts by address aren't allowed because of encryption which only works on complete area.

Table 1. Addressmapping for the IFX Device 3

MODULE	Device 3
- (PFLASH)	0xA0000000 - 0xA03FFFFF
0x00 (PFLASH cached; mapped to PFLASH)	0x80000000 - 0x803FFFFF
0x01 (DFLASH)	0xAF000000 - 0xAF05FFFF
0x04 (HSMPFLASH0; in PFLASH area)	0x80018000 - 0x8001BFFF
0x14 (HSMPFLASH1; in PFLASH area)	0x80060000 - 0x8007FFFF
0x05 (HSMDFLASH)	0xAF110000 - 0xAF11FFFF

Table 2. Addressmapping for the IFX Device 4

MODULE	Device 4
- (PFLASH)	0xA0000000 - 0xA07FFFFF
0x00 (PFLASH cached; mapped to PFLASH)	0x80000000 - 0x807FFFFF
0x01 (DFLASH)	0xAF000000 - 0xAF0BFFFF
0x04 (HSMPFLASH0; in PFLASH area)	0x80018000 - 0x8001BFFF
0x14 (HSMPFLASH1; in PFLASH area)	0x80060000 - 0x8007FFFF
0x05 (HSMDFLASH)	0xAF110000 - 0xAF11FFFF

Table 3. Addressmapping for the IFX Device 5+ (40nm)

MODULE	Device 5+
- (PFLASH)	0xA0000000 - 0xA0FFFFFF
0x00 (PFLASH cached; mapped to PFLASH)	0x80000000 - 0x80FFFFFF
0x01 (DFLASH)	0xAF000000 - 0xAF07FFFF
0x04 (HSMPFLASH0; in PFLASH area)	0x80000000 - 0x8007FFFF
0x14 (HSMPFLASH1; in PFLASH area)	not supported
0x05 (HSMDFLASH)	0xAFC00000 - 0xAFC0FFFF

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	99	01	01	...

Parameters

Parameter set for InitModuleUpload

5	+0
(1*)	**MODULE**

Details

Parameter set for InitModuleUpload

Name	Length	Fmt	Description
MODULE	**1**	**xx**	**Memory module** Number of the memory module (see tables 1 -6) **00_H:** PFLASH (Program Flash) **01_H:** DFLASH (Data Flash) **04_H:** HSMPFLASH0 (HSM Program Flash 0) **14_H:** HSMPFLASH1 (HSM Program Flash 1) **05_H:** HSMDFLASH (HSM Data Flash) **06_H:** BAF (Boot Assist Flash)(JDP only)

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	99	xx	xx	01	...

Parameters

Response of Data Upload By FLASH Id

6	+0	+4
(1..n*)	**ADDR**	**SIZE**

Details

Response of Data Upload By FLASH Id

Name	Length	Fmt	Description
ADDR	**4**	**xxxxxxxx**	**Address of FLASH module.** Address of FLASH area to download.
SIZE	**4**	**xxxxxxxx**	**Size of FLASH module upload.** Size of FLASH area to download. For HSM FLASH areas in exclusive mode the size of additional encrypting overhead (0x300) is added to be used in a following CFI DOWNLOAD command.

.

02 - Upload Module

This command allows uploading a predefined memory module. Command 0x01 ("Init Module Upload") must be executed before this command.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	99	02	01

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	99	xx	xx	01	...

Parameters

Response of Upload Module

6	+0
(1..n*)	**DATA**

Details

Response of Upload Module

Name	Length	Fmt	Description
DATA	**240**	**xx....**	**Uploaded data** Upload the next 240 bytes (or less if the number of bytes to upload or the end of the desired memory module has been reached). The TSW stores the address of the last uploaded byte, so the next call of the UploadModule command uploads the following bytes.

.

03 - Upload By Address

This command allows reading bytes from a specified address. It is checked whether HSM is active and exclusive flags are set. This means that access to these areas is forbidden. In this case you will get an error 0x04 (HSM FLASH exclusive bit set).

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	99	03	01	...

Parameters

Parameter set for Upload By Address

5	+0	+4
(1..n*)	**ADDR**	**LEN**

Details

Parameter set for Upload By Address

Name	Length	Fmt	Description
ADDR	**4**	**xxxxxxxx**	**Address to be read from**
LEN	**1**	**xx**	**Number of bytes to read (0-240)**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	99	xx	xx	01	...

Parameters

Response of Data Upload By Address

6	+0
(1..n*)	**DATA**

Details

Response of Data Upload By Address

Name	Length	Fmt	Description
DATA	**240**	**xx....**	**Read data in ascending order**

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**Offset or Length out of range**
0x02	**No more data**
0x03	**HSM not available or active**
0x04	**HSM FLASH exclusive bit set**
0x05	**HSM uses wrong dump certificate**
0xEF	**HAL_FLS: Locking or unlocking of flash failed**
0xEE	**HAL_FLS: Desired flash area was not erased**

Code	Description
0xED	**HAL_FLS: Programming failed**
0xEC	**HAL_FLS: Erasing failed**
0xEB	**HAL_FLS: Start address not aligned on page size**
0xEA	**HAL_FLS: Desired range is too big for flash**
0xE9	**HAL_FLS: Compare failed**
0xE8	**HAL_FLS: Flash block is not allowed to program/erase**
0xE7	**HAL_FLS: Flash module not found**
0xE6	**HAL_FLS: Sector length too high**
0xE5	**HAL_FLS: Sector length too low**

TSW_A5 - SHE Emulation with HSM

Test step description

Date: 17.11.2014

Created: 18.01.2017 09:12.

This TSW provides functions to access the SHE emulation in HSM.

Table 1. Key Mapping

Key name	Key Id	supported from HSM function
MASTER_ECU_KEY	0x01	yes
KEY_1	0x04	yes
KEY_2	0x05	yes
KEY_3	0x06	yes
KEY_4	0x07	depends on configuration
KEY_5	0x08	depends on configuration
KEY_6	0x09	depends on configuration
KEY_7	0x0A	depends on configuration
KEY_8	0x0B	depends on configuration
KEY_9	0x0C	yes
KEY_10	0x0D	yes

Test step commands

.

00 - Test Key

This command is used to test if a key is already written into the SHE module.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	A5	00	01	...

Parameters

Parameter set for test key

5	+0
(1..n*)	**KEY-ID**

Details

Parameter set for test key

Name	Length	Fmt	Description
KEY-ID	**1**	**xx**	**Key Id, mapping to key name see table 1**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	A5	xx	xx	01	...

Parameters

Response of test key

6	+0
(1..n*)	**RCO**

Details

Response of test key

Name	Length	Fmt	Description
RCO	**1**	**xx**	**Response Code** **Table 2.** <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>0x00</td><td>Key slot empty</td></tr><tr><td>0x01</td><td>Key slot not empty</td></tr><tr><td>0x02</td><td>Error e.g. key not supported</td></tr></table>

.

01 - Write Key

This command is used to write a new key into the SHE module. The reference key ist the previous used key.

Note: If the reference key ist empty, the key value of the reference key is 0.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	A5	01	01	...

Parameters

Parameter set for write key

5	+0	+1	+2	+3
(1*)	**KEY-ID**	**FLAGS**	**OLD_KEY**	**NEW_KEY**

Details

Parameter set for write key

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>KEY-ID</td><td>1</td><td>xx</td><td>Key Id, mapping to key name see table 1</td></tr>
<tr><td>FLAGS</td><td>1</td><td>xx</td><td>SHE-Flags (0x04 means DEBUGGER_DISABLE)
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:5</td><td>reserved</td></tr>
<tr><td>4</td><td>WRITE_PROTECTION</td></tr>
<tr><td>3</td><td>BOOT_ PROTECTION</td></tr>
<tr><td>2</td><td>DEBUGGER_PROTECTION</td></tr>
<tr><td>1</td><td>KEY_USAGE</td></tr>
<tr><td>0</td><td>WILDCARD</td></tr>
</table></td></tr>
<tr><td>OLD_KEY</td><td>1</td><td>xx..</td><td>old 128 bit key</td></tr>
<tr><td>NEW_KEY</td><td>1</td><td>xx..</td><td>new 128 bit key</td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	A5	xx	xx	01	...

Parameters

Response of write key

6	+0
(1*)	**WCN**

Details

Response of write key

Name	Length	Fmt	Description
WCN	**1**	**xx**	**Write Counter** **Note:** Each key has a dedicated counter which is increment on each new key write. This counter is returned in the field WCN.

.

02 - Write Key with Master Key

This command is used to write a new key into the SHE module. The reference key ist the MASTER_ECU_KEY.

Note: If the reference key ist empty, the key value of the reference key is 0.

Note: If the KEY-ID is not 0x01 the MASTER_ECU_KEY should not be empty. In other words, the MASTER_ECU_KEY has to write first.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	A5	02	01	...

Parameters

Parameter set for write key

5	+0	+1	+2	+3
(1*)	**KEY-ID**	**FLAGS**	**MEK**	**NEW_KEY**

Details

Parameter set for write key

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>KEY-ID</td><td>1</td><td>xx</td><td>Key Id, mapping to key name see table 1</td></tr>
<tr><td>FLAGS</td><td>1</td><td>xx</td><td>SHE-Flags (0x04 means DEBUGGER_DISABLE)
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:5</td><td>reserved</td></tr>
<tr><td>4</td><td>WRITE_PROTECTION</td></tr>
<tr><td>3</td><td>BOOT_ PROTECTION</td></tr>
<tr><td>2</td><td>DEBUGGER_PROTECTION</td></tr>
<tr><td>1</td><td>KEY_USAGE</td></tr>
<tr><td>0</td><td>WILDCARD</td></tr>
</table></td></tr>
<tr><td>MEK</td><td>1</td><td>xx..</td><td>128 bit MASTER_ECU_KEY</td></tr>
<tr><td>NEW_KEY</td><td>1</td><td>xx..</td><td>new 128 bit key</td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	A5	xx	xx	01	...

Parameters

Response of write key

6	+0
(1*)	**WCN**

Details

Response of write key

Name	Length	Fmt	Description
WCN	**1**	**xx**	**Write Counter**

Name	Length	Fmt	Description
			Note: Each key has a dedicated counter which is increment on each new key write. This counter is returned in the field WCN.

.

04 - Clear All

This command is used to clear all keys. In other words, the command can be used to reset SHE to the factory defaults.

Note: Clearing of keys is only possible, if no key is write-protected.

Note: Write-protection is possibly set by ASW.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	A5	04	01	...

Parameters

Parameter set for clear all

5	+0
(1*)	**MEK**

Details

Parameter set for clear all

Name	Length	Fmt	Description
MEK	**1**	**xx..**	**128 bit MASTER_ECU_KEY**

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	A5	xx	xx	01

.

E0 - Get last HSM function return code

This command is used to get the last HSM function return code. It is only used for debugging purpose.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	A5	E0	01

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	A5	xx	xx	01	...

Parameters

Response of get last return code

6	+0
(1*)	**RC**

Details

Response of get last return code

Name	Length	Fmt	Description
RC	**4**	**xxxxxxxx**	**return code**

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)

Code	Description
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**TSW: Wrong Version of BootCtrl (must be >= 2.00.00)**
0x02	**TSW: HSM function error (EcuMode_GetEctInfo pending)**
0x03	**TSW: HSM function error**
0x04	**TSW: HSM function error (EcuMode_GetEctInfo error)**
0x80	**TSW: HSM function not available**

TSW_BA - TDG1 CG270/ALTAI

Test step description

Date: 17.09.2014

This test step does the initialisation of the Altai module (CG270), configuration and control of the output channels.

All physical parameters like current, timing etc. use the RVVV format, where R is the resolution (see following table) and VVV is the value of the parameter. For example 2ms would be 0xD002.

Table 1. Resolution table

Value (hex)	Resolution factor		
0	10^0	1	
1	10^1	10	
2	10^2	100	
3	10^3	1.000	1k
4	10^4	10.000	10k
5	10^5	100.000	100k
6	10^6	1.000.000	1M
7	10^7	10.000.000	10M
8	10^-8	0,00000001	10n
9	10^-7	0,0000001	100n
A	10^-6	0,000001	1u
B	10^-5	0,00001	10u
C	10^-4	0,0001	100u
D	10^-3	0,001	1m
E	10^-2	0,01	10m
F	10^-1	0,1	100m

Test step commands

.

01 - Initialize Altai module

This command shall be used to initialize the Altai module.

The serial SPI interface establishes a communication link with the systems microcontroller. SPI may be configured as standard SPI with single-ended inputs / output or as HSSPI (a high speed serial protocol interface) with LVDS (low voltage differential signal) pins. The maximum baud rate is 20MBaud for HSPI and 4MBaud SPI.

NOTE! All SPI parameters must be set with TSW_02 before using Altai commands.

Altai initial configuration is executed with this command too. It is configuring Altai registers for most commonly used values. The initial configuration gives the user a good functional basis for further more detailed settings.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	BA	01	01	...

Parameters

Assign a node index to Altai module

5	+0
(1*)	**NODE**

Parameters for initialize high speed or normal SPI by port/pin numbers (PO=PORT, PI=PIN)

6	+0	+1	+2	+3	+4	+5
(1*)	**PO_MTSR**	**PI_MTSR**	**PO_MRST**	**PI_MRST**	**PO_CLK**	**PI_CLK**

...	+6	+7
(1*)	**PO_SLSO**	**PI_SLSO**

Chopper frequency

14	+0
(1*)	**CFRQ**

Details

Assign a node index to Altai module

Name	Length	Fmt	Description
NODE	**1**	**xx**	**Altai node number** Assign a node index to Altai module.

Parameters for initialize high speed or normal SPI by port/pin numbers (PO=PORT, PI=PIN)

Name	Length	Fmt	Description
PO_MTSR	**1**	**xx**	**Port number of MTSR signal** Port number of SPI out signal. At HSSPI serial output positive signal.
PI_MTSR	**1**	**xx**	**Pin number of MTSR signal** Pin number of SPI out signal. At HSSPI serial output positive signal.
PO_MRST	**1**	**xx**	**Port number of MRST signal**

Name	Length	Fmt	Description
			Port number of SPI out signal. At HSSPI serial output positive signal.
PI_MRST	**1**	**xx**	**Pin number of MRST signal** Pin number of SPI in signal. At HSSPI serial input positive signal.
PO_CLK	**1**	**xx**	**Port number of CLK signal** Port number of SPI clock signal.
PI_CLK	**1**	**xx**	**Pin number of CLK signal** Pin number of SPI clock signal.
PO_SLSO	**1**	**xx**	**Port number of SLSO signal** Port number of slave select signal.
PI_SLSO	**1**	**xx**	**Pin number of SLSO signal** Pin number of slave select signal.

Chopper frequency

Name	Length	Fmt	Description
CFRQ	**2**	**xxxx**	**Chopper frequency** Desired chopper frequency for module. Controlled between 1kHz and 5kHz. Format: RVVV where R=Resolution, V=Value See 'Resolution Table'

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	BA	xx	xx	01

.

02 - Set hardware parameters

With this command the characteristic parameters for the hardware shall be set. These regulator parameters are individually available for each channel.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	BA	02	01	...

Parameters

Parameters for hardware settings

5	+0	+1	+2	+3	+5	+7
(1*)	**NODE**	**CH**	**MODE**	**FWDSLW**	**FWDFST**	**UFWSLW**

...	+9	+11	+13	+15
(1*)	**UFWFST**	**CTFWFST**	**FWDHYS**	**FWCHYS**

Details

Parameters for hardware settings

Name	Length	Fmt	Description
NODE	**1**	**xx**	**Altai node number** Index of the Altai node for which parameters shall be set.
CH	**1**	**xx**	**Channel number** Channel number of output
MODE	**1**	**xx**	**Mode selection** Output operation mode **0_H:** Normal **1_H:** Fast freewheel
FWDSLW	**2**	**xxxx**	**Gate driver configuration during slow mode**
FWDFST	**2**	**xxxx**	**Gate driver configuration during fast mode**
UFWSLW	**2**	**xxxx**	**Freewheel voltage for slow mode**
UFWFST	**2**	**xxxx**	**Freewheel voltage for fast mode**
CTFWFST	**2**	**xxxx**	**Counter for fast freewheel activation**
FWDHYS	**2**	**xxxx**	**Dither hysteresis for fast freewheel on**
FWCHYS	**2**	**xxxx**	**Current hysteresis for fast freewheel on**

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	BA	xx	xx	01

.

03 - Set regulator parameters

With this command the characteristic parameters for the regulator shall be set. These regulator parameters are individually available for each channel.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	BA	03	01	...

Parameters

Parameters for regulator settings

5	+0	+1	+2	+3	+5	+7
(8*)	**NODE**	**CH**	**TYP**	**CFRQ**	**RES**	**NTAU**

...	+9	+11	+13	+15
(8*)	**NCUR**	**HTAU**	**HCUR**	**FSSA**

Details

Parameters for regulator settings

Name	Length	Fmt	Description
NODE	**1**	**xx**	**Altai node number** Index of the Altai node for which parameters shall be set.
CH	**1**	**xx**	**Channel number** Channel number of output
TYP	**1**	**xx**	**Regulator type** Regulator type **Value** / **Regulator type** 0x00 / A 0x01 / B
CFRQ	**2**	**xxxx**	**Chopper frequency** Desired chopper frequency on current output. Controlled between 1kHz and 5kHz. Format: RVVV where R=Resolution, V=Value See 'Resolution Table'
RES	**2**	**xxxx**	**Resistance** Serial resistance of the solenoid. Controlled between 2,25Ohm and 18Ohm. Format: RVVV where R=Resolution, V=Value
NTAU	**2**	**xxxx**	**Nominal time** Time constant based on solenoid type at nominal current. Controlled between 2ms and 200ms. Format: RVVV where R=Resolution, V=Value
NCUR	**2**	**xxxx**	**Nominal current range** Current load at nominal tau was measured. Controlled between 0 and 4A. Format: RVVV where R=Resolution, V=Value
HTAU	**2**	**xxxx**	**High time** Time constant based on solenoid type at maximal control current. Controlled between 2ms and 200ms.

Name	Length	Fmt	Description
			Format: RVVV where R=Resolution, V=Value
HCUR	**2**	**xxxx**	**High current range** Current load at high tau was measured. Controlled between 0 and 4A. Format: RVVV where R=Resolution, V=Value
FSSA	**1**	**xx**	**Factor of shall-spring-answer** Regulatordynamic factor 0x03: CG270 Ax-step, 0x08: CG270 Bx-step

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	BA	xx	xx	01

.

04 - Set current

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	BA	04	01	...

Parameters

Parameters for current settings

5	+0	+1	+2	+3
(1*)	**NODE**	**CH**	**MODE**	**VAL**

Parameter for current compensation function

11	+0
(1*)	**COMP**

Details

Parameters for current settings

Name	Length	Fmt	Description
NODE	**1**	**xx**	**Altai node number** Index of the Altai node for which parameters shall be set.
CH	**1**	**xx**	**Channel number** Channel number of output

Name	Length	Fmt	Description
MODE	**1**	**xx**	**Compensation data search mode** **Bit(s)** / **Description** **7:4** — **Current compensation data** **0_H:** No compensation **1_H:** Automatic data search in EEPROM **2_H:** Data from user defined EEPROM address **3:0** — **reserved**
VAL	**3**	**xxxxxx**	**Current value** Output current Resolution of current value is 0,125mA. Controlled between 0 to 4095,875mA. Format: VVVVTT where V=Value (mA), T=Tenth (n * 0,125mA) Example: 1000.375mA -> V=0x3E8 T=0x03

Parameter for current compensation function

Name	Length	Fmt	Description
COMP	**4**	**xxxxxxxx**	**Parameter for current compensation data in EEPROM** Meaning of bits 32:0 depends on 'MODE' parameter. The first description describes the case, if 'Automatic data search in EEPROM' is selected. The second description describes the case, if 'Data from user defined EEPROM address' is selected. If compensation not used, it shall to be set 0. **Bit(s)** / **Description** **31:23** — **Not used, shall to be set 0** **22:8** — **Block-ID** **7:0** — **Version number** **Bit(s)** / **Description** **31:0** — **EEPROM address of current compensation data**

Response

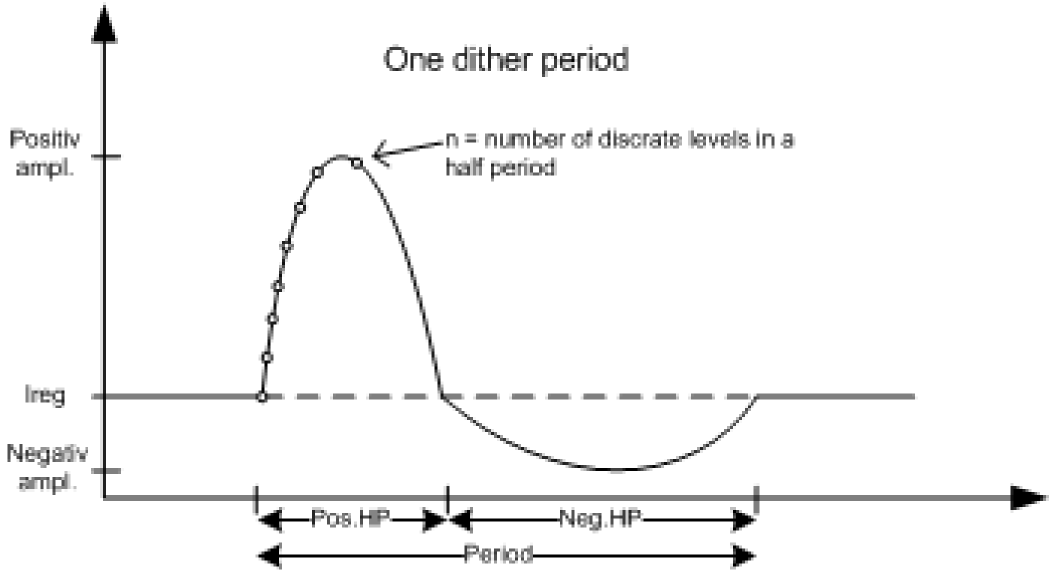
1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	BA	xx	xx	01

.

05 - Set dither parameters

This command shall be used to set the Altai dither parameters.

Dither amplitude is modulated upon the actual output current to get a sensitive reaction on the variable force solenoid (VFS). Therefore the amplitude, frequency and curve shape are individually adjustable for each channel.



Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	BA	05	01	...

Parameters

Parameters for dither settings

5 (1*)	+0	+1	+2	+4	+6	+8
	NODE	**CH**	**AMPPOS**	**AMPNEG**	**PERPOS**	**PERNEG**

... (1*)	+10
	SHAPE

Details

Parameters for dither settings

Name	Length	Fmt	Description
NODE	**1**	**xx**	**Altai node number**

Name	Length	Fmt	Description
			Index of the Altai node for which parameters shall be set.
CH	**1**	**xx**	**Channel number** Channel number of output
AMPPOS	**2**	**xxxx**	**Positive amplitude** Positive dither current amplitude Resolution of current value is 1mA. Controlled between 1mA to 400mA.
AMPNEG	**2**	**xxxx**	**Negative amplitude** Negative dither current amplitude Resolution of current value is 1mA. Controlled between 1mA to 400mA.
PERPOS	**2**	**xxxx**	**Positive half period** Time for positive dither half period Controlled between 20us and 327ms. Format: RVVV where R=Resolution, V=Value See 'Resolution Table'
PERNEG	**2**	**xxxx**	**Negative half period** Time for negative dither half period Controlled between 20us and 328ms. Format: RVVV where R=Resolution, V=Value See 'Resolution Table'
SHAPE	**1**	**xx**	**Curve shape selection** Selection of dither curve **0_H:** Sinus **1_H:** Triangle **2_H:** Square

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	BA	xx	xx	01

.

06 - PWM generator

This command shall be used to control an output with PWM signal.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	BA	06	01	...

Parameters

Parameters for PWM generator

5	+0	+1	+2	+4
(1*)	**NODE**	**CH**	**FRQ**	**DUTY**

Details

Parameters for PWM generator

Name	Length	Fmt	Description
NODE	**1**	**xx**	**Altai node number** Index of the Altai node for which parameters shall be set.
CH	**1**	**xx**	**Channel number** Channel number of output
FRQ	**2**	**xxxx**	**Frequency of PWM signal** PWM frequency Format: RVVV where R=Resolution, V=Value See 'Resolution Table'
DUTY	**2**	**xxxx**	**Duty cycle of PWM signal** Resolution of duty cycle is 0,00305176%. Controlled between 0 and 100% (0x0 - 0x8000).

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	BA	xx	xx	01

.

10 - Measure output current

This command shall be used to measure the actual current value of an output.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	BA	10	01	...

Parameters

Parameters for measurement

5	+0	+1	+2
(1*)	**NODE**	**CH**	**NSA**

Details

Parameters for measurement

Name	Length	Fmt	Description
NODE	**1**	**xx**	**Altai node number** Index of the Altai node for which parameters shall be set.
CH	**1**	**xx**	**Channel number** Channel number of output
NSA	**1**	**xx**	**Number of samples** Number of samples used to calculate avarage, min, and max value.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	BA	xx	xx	01	...

Parameters

Response of current measure

6	+0	+3
(2*)	**AVRC**	**AVRD**

Details

Response of current measure

Name	Length	Fmt	Description
AVRC	**3**	**xxxxxx**	**Average current value** Format: VVVVTT where V=Value (mA), T=Tenth (n * 0,125mA)
AVRD	**3**	**xxxxxx**	**Average current of a dither period** Format: VVVVTT where V=Value (mA), T=Tenth (n * 0,125mA) NOTE! Dither frequency / Chopper frequency = n where n smaller than 448 intiger.

.

20 - Write register

This command may be used to write Altai registers from a specific address. The address increased by offset.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	BA	20	01	...

Parameters

Parameters for write

5	+0	+1	+2	+4	+5
(1*)	**NODE**	**CORE**	**ADR**	**OFS**	**NRR**

Data for write

11	+0
(1..n*)	**VAL**

Details

Parameters for write

Name	Length	Fmt	Description
NODE	**1**	**xx**	**Altai node number** Index of the Altai node for which parameters shall be set.
CORE	**1**	**xx**	**Core ID** Which core (0/1/2) shall be written.
ADR	**2**	**xxxx**	**Register address** First register address to write.
OFS	**1**	**xx**	**Address offset** Offset for the next register address to write.
NRR	**1**	**xx**	**Number of register(s)** How many registers shall be written.

Data for write
Repeatable parameter

Name	Length	Fmt	Description
VAL	**2**	**xxxx**	**Register(s) value**

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	BA	xx	xx	01

.

30 - Read register

This command may be used to read Altai registers from a specific address. The address increased by offset.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	BA	30	01	...

Parameters

Parameters for read

5	+0	+1	+2	+4	+5
(1*)	**NODE**	**CORE**	**ADR**	**OFS**	**NRR**

Details

Parameters for read

Name	Length	Fmt	Description
NODE	**1**	**xx**	**Altai node number** Index of the Altai node for which parameters shall be set.
CORE	**1**	**xx**	**Core ID** Which core (0/1/2) shall be read.
ADR	**2**	**xxxx**	**Register address** First register address to read.
OFS	**1**	**xx**	**Address offset** Offset for the next register address to read.
NRR	**1**	**xx**	**Number of register(s)** How many registers shall be read.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	BA	xx	xx	01	...

Parameters

Register(s) value

6	+0
(1..n*)	**VAL**

Details

Register(s) value

Name	Length	Fmt	Description
VAL	**2**	**xxxx**	**Register(s) value**

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)

Code	Description
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**ASIC not found or not supported**
0x02	**Node not initialized**
0x03	**Invalid current value**
0x04	**Invalid amplitude value**
0x05	**Invalid period value**
0x06	**Invalid Core id**
0x07	**Regulator type not supported**
0x08	**Compension data plausibility error**
0x09	**No compensation data found**
0x0A	**Channel not available**
0x0B	**Invalid EEPROM address**
0x0C	**Invalid compensation value (>25mA)**

Code	Description
0x0D	**Failure during initialization**
0x0E	**Failure during SPI communication**

TSW_BB - TDG1 CG135/TARPAN

Test step description

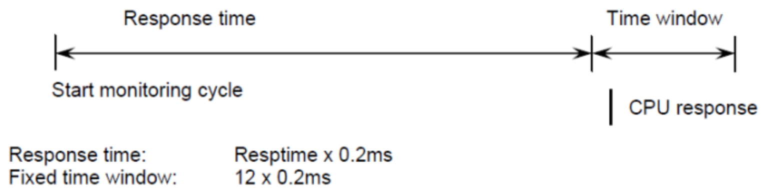
Created 18.01.2017 09:12.

CG135 shall be an ASIC having mainly monitoring and watchdog functionality. CG135 shall make possible an ASIL D capable TCU voltage supply and mC control system together with an SBC that does not comply with ASIL requirements (QM).

The monitoring module starts automatically and works all the time regardless of the controller functions. By default a Question Answer Watchdog (QAWD) is available. For CG135BA in addition there ist a Time Watchdog (TWD).

Question Answer Watchdog

On reset QAWD ist active (For CG135BA this means INIT_CONF.WD_MODE = 0). The watchdog asks different questions that the controller must retrieve and answer correctly within a specified time window. The monitoring module verifies whether the answer appears within a time window and whether the value indicated in the answer is totally correct.



The question is a 4-bit word. This 4-bit word can be retrieved any time by the controller from the REQULO register by a read access. The response time and response time window are not affected by reading of the REQULO register. In addition, the monitoring module uses a combinatorial analysis to generate a set point answer based on the 4-bit word to be compared against the answer from the controller.

The answer is a 32-bit word made up of the 4 bytes RESP-BYTE3, RESP-BYTE2, RESP-BYTE1 and RESP-BYTE0. The 4 bytes are sent over the SPI interface in succession (order: bytes 3-2-1-0) to the monitoring module. All 4 bytes must give a correct answer within the response window.
The error counter can assume values between 0 and 7. When the error count is 7, and there are other wrong responses or responses outside the time window, or a response time = 0, the error count will remain at 7.

The response time window is fixed to 2.4ms. The response time can be set between 0 ms and $(2^6 - 1) * 0.2\text{ms} = 12.6\text{ms}$ by the controller RESP_TIME register.

Time Watchdog

For CG135BA TWD can be activated if INIT_CONF.WD_MODE is set to '1'. The counter is reset by entering INIT mode. The counter shall also be reset and restarted if internal signal ERPN_ST is low and one of the two conditions are fullfiled.

1. SPI register bit TWD_SERV(1-0) changes from "10" to "01" or from "01" to "10"

2. SPI register WD_MODE is set from '0' to '1'

Internal signal ERPN_ST shall be set to low, when

1. Pin ERPN is set to high.

2. SPI bit ERPN_OUT_EN is set to '1' and SPI bit TEST_DN is set to '1' after $t_{ERPN\_DIAG-delay}$. The TIMEWD generates a reset event and set SPI bit TWD_TO to '1' when the counter overflows after time $t_{twd-resp}$ and SPI register bit WD_MODE is set from '0' to '1'. A change of WD_MODE shall take effect after $t_{wd-mode-delay}$.

Test step commands

.

01 - WD and TAR Service

Start/stop watchdog service of CG135.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	BB	01	01	...

Parameters

Parameters for watchdog service.

5	+0	+1	+2	+3	+4	+5
(1*)	**PORTMT**	**PINMT**	**PORTMR**	**PINMR**	**PORTCK**	**PINCK**

...	+6	+7
(1*)	**PORTSL**	**PINSL**

Parameter LVOP/WD/TAR configuration.

13	+0	+1	+2
(1..n*)	**LVOP**	**WD**	**TAR**

Details

Parameters for watchdog service.

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number within PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal

Name	Length	Fmt	Description
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

Parameter LVOP/WD/TAR configuration.

Name	Length	Fmt	Description
LVOP	**1**	**xx**	**Low battery voltage operation.** Enable or disable low battery voltage operation.
WD	**1**	**xx**	**Set watchdog state.** Set watchdog state (Period 9.5 ms) 0: Watchdog 0ff. 1: QAWD Question & answer watchdog. 2: TWD Time watchdog. For Tarpan2 (CG135B) only.
TAR	**1**	**xx**	**Set afterrun state.** Set TAR state (Period 300 ms) 0x00/0x01.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	BB	xx	xx	01	...

Parameters

Response of command 0x00

6	+0	+1	+2	+3	+4
(1*)	**LVOP**	**WD**	**TAR**	**ID**	**REV**

Details

Response of command 0x00

Name	Length	Fmt	Description
LVOP	**1**	**xx**	**Set low battery voltage operation.** Low battery voltage operation enabled or disabled.
WD	**1**	**xx**	**Set watchdog state.** Watchdog state (on/off).
TAR	**1**	**xx**	**Set afterrun state.** TAR state (on/off).
ID	**1**	**xx**	**Chip Id** Chip-ID.
REV	**1**	**xx**	**Chip SW/HW revision.** SW and HW revision.

.

02 - HSD and sensor output.

Configure afterrun.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	BB	02	01	...

Parameters

Parameters to address SPI connection

5	+0	+1	+2	+3	+4	+5
(1*)	**PORTMT**	**PINMT**	**PORTMR**	**PINMR**	**PORTCK**	**PINCK**

...	+6	+7
(1*)	**PORTSL**	**PINSL**

Parameter to set HSD and sensor output.

13	+0	+1
(1*)	**HSD**	**SO**

Details

Parameters to address SPI connection

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number within PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

Parameter to set HSD and sensor output.

Name	Length	Fmt	Description
HSD	**1**	**xx**	**High side driver.** Configure high side drivers on/off: [x,x,x,x,x,HSD2_ON,HSD1_ON,HSD0_ON].
SO	**1**	**xx**	**Sensor output.** Configure sensor output: Cin_UV_SEL = sensor type 1 or 0. [x,x,CI2_UV_SEL,CI2_ON,CI1_UV_SEL,CI1_ON, CI0_UV_SEL,CI0_ON]

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	BB	xx	xx	01	...

Parameters

Status of HSD and sensor output.

6	+0	+1	+2	+3	+4
(1*)	**INIREG0**	**DIA_DN1**	**DIA_CI0**	**DIA_CI1**	**DIA_CI2**

Details

Status of HSD and sensor output.
Registers containing HSD and sensor output.

Name	Length	Fmt	Description
INIREG0	**1**	**xx**	**INIT_REG0** INIREG0
DIA_DN1	**1**	**xx**	**DIA_DN1** DIA_DN1
DIA_CI0	**1**	**xx**	DIA_CI0
DIA_CI1	**1**	**xx**	DIA_CI1
DIA_CI2	**1**	**xx**	DIA_CI2

.

03 - Internal test.

Run a couple of internal tests. Select tests to run using flags: 1: run. Bits 15..6: Flag descriptions.

Table 1. Initialisation Test Steps

N	Name	Function
15	ERPN	External error pin ACTIVE.
14	SUV1	VDD1 slow treshold undervoltage detection OK.
13	SOV1	VDD1 slow treshold overvoltage detection OK.
12	FUV1	VDD1 fast treshold undervoltage detection OK.
11	FOV1	VDD1 fast treshold overvoltage detection OK.

N	Name	Function
10	UV3	VDD3 undervoltage detection OK.
9	OV3	VDD3 overvoltage detection OK.
8	UV5	VDD5 undervoltage detection OK.
7	OV5	VDD5 overvoltage detection OK.
6	OVINT	DN reset simulation OK.
5	VB_SUP	VDD5S and VB OK.
0	DNRST	Initial DN reset detected. Tests skipped.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	BB	03	01	...

Parameters

Parameters address SPI connection

5	+0	+1	+2	+3	+4	+5
(1*)	**PORTMT**	**PINMT**	**PORTMR**	**PINMR**	**PORTCK**	**PINCK**

...	+6	+7
(1*)	**PORTSL**	**PINSL**

Test flags, SMUFSM pin.

13	+0	+2	+3
(1*)	**TESTFLG**	**PORTSMU**	**PINSMUF**

Details

Parameters address SPI connection

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number within PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal

Name	Length	Fmt	Description
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

Test flags, SMUFSM pin.

Name	Length	Fmt	Description
TESTFLG	**2**	**xxxx**	**Test configuration.** Run tests activated by flags: 1: run. See test flag descriptions.
PORTSMU	**1**	**xx**	**SMUFSM port.** Port number of SMUFSM.
PINSMUF	**1**	**xx**	**SMUFSM pin.** Pin number of SMUFSM.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	BB	xx	xx	01	...

Parameters

Result flags of init tests.

6	+0
(1*)	**TESTFLG**

Details

Result flags of init tests.

Name	Length	Fmt	Description
TESTFLG	**2**	**xxxx**	**Test results** Result of tests as flags: 0: OK, 1: error. See test flag descriptions. In case of a failed test additional diagnostic information is returned (DN_DIA0, DN_DIA1).

.

04 - Debug Function.

Control HSD.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	BB	04	01	...

Parameters

Parameters address SPI connection

5	+0	+1	+2	+3	+4	+5
(1*)	**PORTMT**	**PINMT**	**PORTMR**	**PINMR**	**PORTCK**	**PINCK**

...	+6	+7
(1*)	**PORTSL**	**PINSL**

Parameter to set debug mode.

13	+0
(1*)	**STATE**

Details

Parameters address SPI connection

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number within PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

Parameter to set debug mode.

Name	Length	Fmt	Description
STATE	**1**	**xx**	**Debug mode on/off.** Configure debug mode 0x00/0x01.

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	BB	xx	xx	01

.

05 - Read/write registers.

Read and write registers with bit mask.

ATTENTION! Interface changed recently! Reading twice to check correct writing under discussion.

On writing register is read and masked bits are written with value read.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	BB	05	01	...

Parameters

Parameters address SPI connection

5	+0	+1	+2	+3	+4	+5
(1*)	**PORTMT**	**PINMT**	**PORTMR**	**PINMR**	**PORTCK**	**PINCK**

...	+6	+7
(1*)	**PORTSL**	**PINSL**

CG135 commands.

13	+0	+1	+2
(1..n*)	**CMD**	**MASK**	**DATA**

Details

Parameters address SPI connection

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number within PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal**

Name	Length	Fmt	Description
			Port number of SLSO (Slave Select) signal
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

CG135 commands.

Name	Length	Fmt	Description
CMD	**1**	**xx**	**Register read/write command.** Register read/write command with 8 bit mask. Bits not included in mask will be returned as 0 on read.
MASK	**1**	**xx**	**Read/write mask.** Mask for bits to write or read.
DATA	**1**	**xx**	**Write data.** Data to write. 0x00 for read.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	BB	xx	xx	01	...

Parameters

SPI commands.

6	+0	+1
(1..n*)	**CTRL**	**DATA**

Details

SPI commands.

Name	Length	Fmt	Description
CTRL	**1**	**xx**	**Control byte.** Control byte. For CG135 0xAA is ok.
DATA	**1**	**xx**	**Command data.** Written data for write, read data for read. For written data this means data read after write.

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**

Code	Description
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**TSW: Invalid SPI RX frame.**
0x02	**TSW: Sync timeout.**
0x03	**TSW: No command sequence.**
0x04	**TSW: WD not initialised.**
0x05	**TSW: WD type not valid for chip revision.**
0xFF	**HAL busy**
0xFE	**Generic HAL error**
0xFD	**HAL not compatible to current CPU**
0xFC	**Invalid params**
0xFB	**HAL module not implemented**
0xFA	**HAL function not implemented**

Code	Description
0xF9	**HAL timeout**

TSW_BD - Formated plant data programming for TDG1

Test step description

Date: 14.04.2016

This test step collect and program plant data form temporary place to final place in a special form according to block-ID.

Test step commands

.

01 - Programming block-ID 6F14

If one of the following parameters not used, it shall to set 0.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	BD	01	01	...

Parameters

Block parameters for programming block-ID 6F14

5	+0	+4	+8	+12	+16	+20
(1*)	**RT**	**TT**	**HT**	**DEST**	**CH**	**C0**

...	+22	+24	+26	+28	+30	+32
(1*)	**C1**	**C2**	**C3**	**C4**	**T1**	**T2**

...	+34	+36
(1*)	**CAL**	**STEP**

Details

Block parameters for programming block-ID 6F14

Name	Length	Fmt	Description
RT	**4**	**xxxxxxxx**	**Address of teporary data for RT**
TT	**4**	**xxxxxxxx**	**Address of teporary data for TT**
HT	**4**	**xxxxxxxx**	**Address of teporary data for HT**
DEST	**4**	**xxxxxxxx**	**Destination address of 6F14 block**
CH	**4**	**xxxxxxxx**	**Bitmask for programmed channels**
C0	**2**	**xxxx**	**Current interpolation axis point 0**
C1	**2**	**xxxx**	**Current interpolation axis point 1**

Name	Length	Fmt	Description
C2	**2**	**xxxx**	**Current interpolation axis point 2**
C3	**2**	**xxxx**	**Current interpolation axis point 3**
C4	**2**	**xxxx**	**Current interpolation axis point 4**
T1	**2**	**xxxx**	**Temperature axis point 1**
T2	**2**	**xxxx**	**Temperature axis point 2**
CAL	**2**	**xxxx**	**Calibration value for CG270 self-calibration** Activation of the calibration function for all ASICs. 0 = Calibration deactivated 1 = Calibration used from default settings in SW/ ASIC other = Register value Scale_CAL
STEP	**1**	**xx**	**Step of CG270** **0_H:** CG270 BA **1_H:** CG270 BB

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	BD	xx	xx	01

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)

Code	Description
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**Current limit failure**

TSW_C0 - Variable Valve Train actuation (VVT3/VVT4)

Test step description

Date: 30.04.2015

This test step allows operation of 3-phase VVT3 or VVT4 operation.

This teststep is designed for Validation SW

The VVT3/VVT4 (Variable Valve Train) system needs to be activated for EMV tests. The actuator (a 3 phase servo motor) is continuously moved between 10% and 90% of it's operating range. The mechanical end position is found by moving the actuator to it's mechanical end position lock.

The test step creates a 3-phase PWM signal which is commutated to the desired motor velocity. The commutation of the 3-phases and the handling of switching the servo motor backwards and forwards within the 10-90% test range is done as a background service which allows other tests to be performed from SCC in the meantime.

The 3-phase PWM signal is driven through a TLE7183 driver to the servo motor. The status of the driver chip may be requested after the initialization at any time within the test sequence.

Since this test step does not include any position sensing of the actuator, all control parameters are based on times/frequencies.

The velocity of actuator motion is controlled by the commutation frequency of the PWM signal between the three phases (U,V,W).

The current is controlled by the PWM's duty cycle.

The operating range is defined by the rotation angle of the servo motor to move the actuator from one end position to the other.

Tip

Velocity and current for moving and holding of the actuator have to be selected small enough to ensure that no mechanical parts are damaged by the actuator's motion to the mechanical end position lock and the drawn current for motion/hold of the actuator does NOT create any overheating. Please refer to manufacturers recommendation for selection of these parameters.

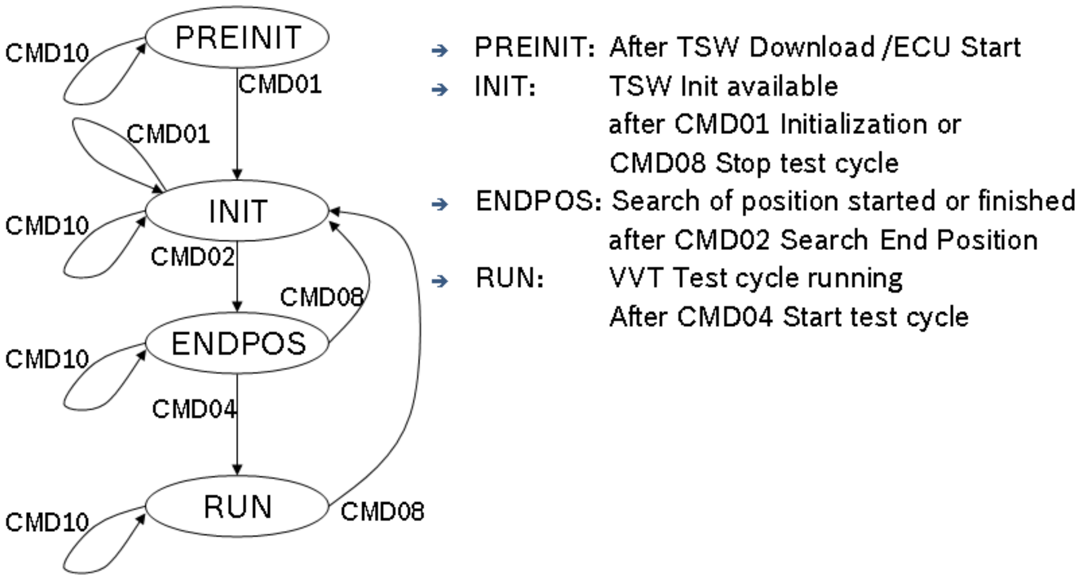
Tip

In order to prevent damage to mechanical parts, for any long term test cycles the lubrication of all mechanical parts needs to be ensured.

State machine control for TSW_C0

The TSW_C0 is controlled by a state machine since not at all times all commands can be executed. Please refer to the state diagram to know which command sequence is necesary for correct operation.

Figure 1: VVT3/VVT4: State machine control for TSW_C0



Test step commands

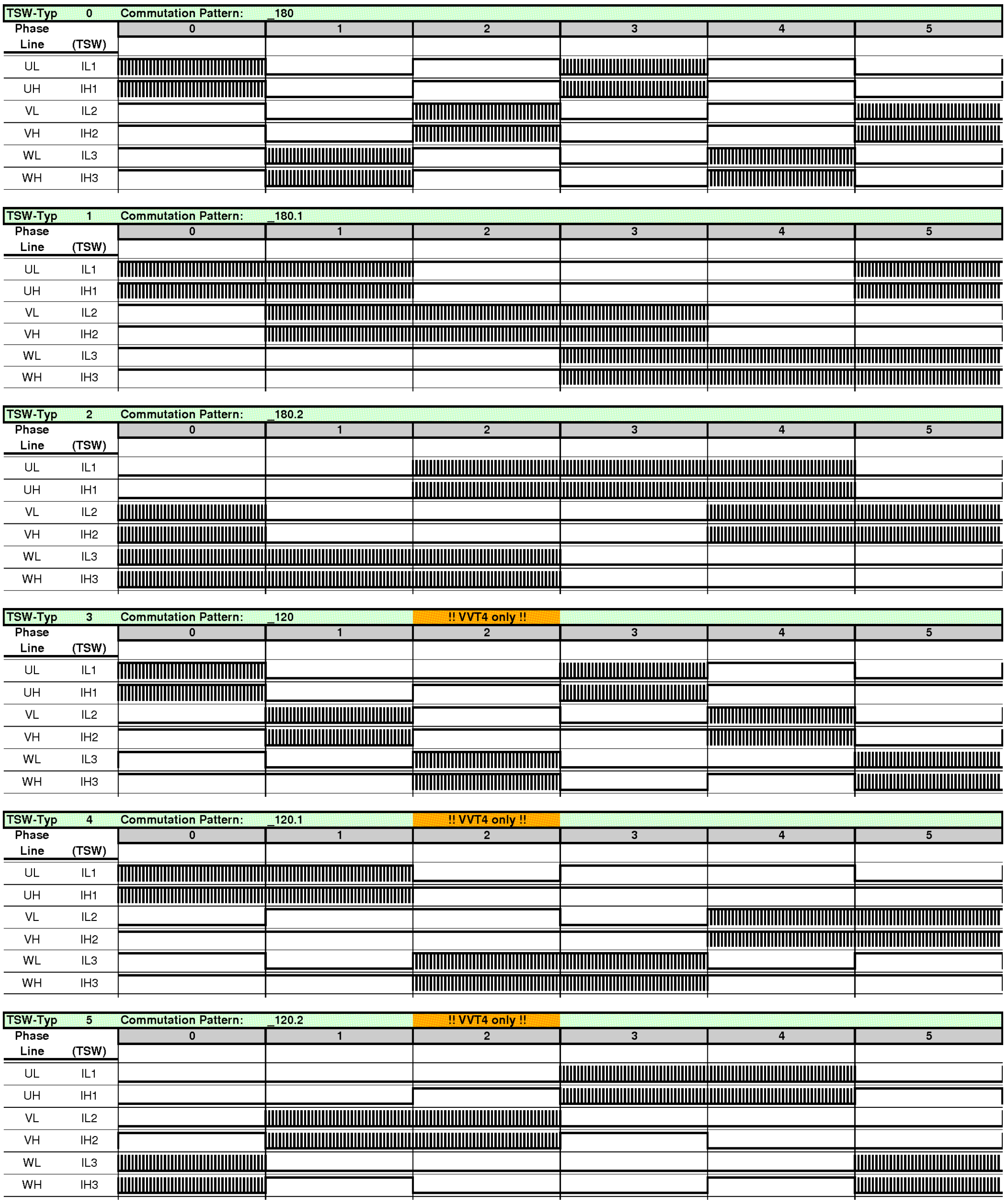
.

01 - VVT3/VVT4 Initialization

The initialization performs all necessary steps on the controller and TLE7183 to allow VVT3/VVT4 operation. It allows selection of all required port/pins, PWM and signal commutation parameters like type of commutation pattern, velocity of actuator motor and angle of motor's motion range.

Pls. refer to the commutation patterns according to the following table.

Figure 2: VVT3/VVT4: Selectable commutation patterns



Tip

Please note that the basic pins setup for the following mentioned uController pins is not performed by this teststep. Reason is that a previous setup according to the port assignment is not overwritten. Therefore all pins which are used within tsw_c0 functionality needs to be configured according to pin function, driver strength and slew rate with the GPIO teststep TSW_04 before TSW_C0 is used.

PWM and Commutation timer settings

All physical parameters like period, timeouts etc. use the RVVV format, where R is the resolution (see following table) and VVV is the value of the parameter. For example 2ms would be 0xD002.

Table 1. Resolution table

Value	Resolution
0x8	10ns

Value	Resolution
0x9	100ns
0xA	1µs
0xB	10µs
0xC	100µs
0xD	1ms
0xE	10ms
0xF	100ms

The following Timer Resources (GTM, TPU, EMIOS etc.) are available:

Table 2. Timer Resources

Value	IFX/JDP	FSL (not supported)
0x00	TOM0	TPU_A (not supported)
0x01	TOM1	TPU_B (not supported)
0x02	TOM2 (not supported)	TPU_C (not supported)
0x03	unused (not supported)	MIOS (not supported)

Tip

If a TOM module is used for VVT the TOM Channels 0...7 are occupied and can not be used for PWM input / ouput operation. This is related to GTM definition and therefore valid for IFX and JDP controllers equally.

Required previous successful command execution: If VVT already actuated: CMD08

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	C0	01	01	...

Parameters

Parameter set for VVT PIN initialization. Pin which are marked as optional need not to be used. If a pin is unused please use Port=Pin=0xFF as parameter in the test step call.

5	+0	+2	+4	+6	+8	+10
(1..n*)	**INH**	**EN1**	**EN2**	**ER1**	**ER2**	**IL1**

...	+12	+14	+16	+18	+20
(1..n*)	**IH1**	**IL2**	**IH2**	**IL3**	**IH3**

Parameter set for VVT PWM initialization

27	+0	+2	+4	+6
(1..n*)	**PWMSRC**	**PWMPER**	**PWMDCM**	**PWMDCH**

Parameter set for VVT Commutation initialization

35	+0	+1	+3	+5
(1..n*)	**CMTTYP**	**CMTVEL**	**CMTRNG**	**CMTPAU**

Details

Parameter set for VVT PIN initialization. Pin which are marked as optional need not to be used. If a pin is unused please use Port=Pin=0xFF as parameter in the test step call.

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>INH</td><td>2</td><td>xxxx</td><td>Port/Pin for signal INH
optional<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>15:8</td><td>Port number</td></tr><tr><td>7:0</td><td>Pin number</td></tr></table></td></tr>
<tr><td>EN1</td><td>2</td><td>xxxx</td><td>Port/Pin for signal ENA1
optional<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>15:8</td><td>Port number</td></tr><tr><td>7:0</td><td>Pin number</td></tr></table></td></tr>
<tr><td>EN2</td><td>2</td><td>xxxx</td><td>Port/Pin for signal ENA2
optional<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>15:8</td><td>Port number</td></tr><tr><td>7:0</td><td>Pin number</td></tr></table></td></tr>
<tr><td>ER1</td><td>2</td><td>xxxx</td><td>Port/Pin for signal ERR1
optional<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>15:8</td><td>Port number</td></tr><tr><td>7:0</td><td>Pin number</td></tr></table></td></tr>
<tr><td>ER2</td><td>2</td><td>xxxx</td><td>Port/Pin for signal ERR2
optional<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>15:8</td><td>Port number</td></tr><tr><td>7:0</td><td>Pin number</td></tr></table></td></tr>
<tr><td>IL1</td><td>2</td><td>xxxx</td><td>Port/Pin for signal IL1<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>15:8</td><td>Port number</td></tr><tr><td>7:0</td><td>Pin number</td></tr></table></td></tr>
<tr><td>IH1</td><td>2</td><td>xxxx</td><td>Port/Pin for signal IH1<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>15:8</td><td>Port number</td></tr></table></td></tr>
</table>

Name	Length	Fmt	Description
			Bit(s) / **Description** **7:0** — **Pin number**
IL2	**2**	**xxxx**	**Port/Pin for signal IL2** **Bit(s)** / **Description** **15:8** — **Port number** **7:0** — **Pin number**
IH2	**2**	**xxxx**	**Port/Pin for signal IH2** **Bit(s)** / **Description** **15:8** — **Port number** **7:0** — **Pin number**
IL3	**2**	**xxxx**	**Port/Pin for signal IL3** **Bit(s)** / **Description** **15:8** — **Port number** **7:0** — **Pin number**
IH3	**2**	**xxxx**	**Port/Pin for signal IH3** **Bit(s)** / **Description** **15:8** — **Port number** **7:0** — **Pin number**

Parameter set for VVT PWM initialization

Name	Length	Fmt	Description
PWMSRC	**2**	**xxxx**	**Parameter for PWM timer resource** Timer module that generates the signal. With TRESOUR=0xFFFF the timer resource will be selected automatically. Example: PWMSRC = 0x0101 => Select TOM1 and channel 1. **Bit(s)** / **Description** **15:8** — **Timer module (see table 'Timer Resources')** **7:0** — **Timer channel (only CH0 and CH1 are supported because of SPE internal connections)**
PWMPER	**2**	**xxxx**	**Period and resolution for PWM** "PERIOD" contains the period and the resolution of the PWM. Format: RVVV where R=Resolution, V=Value of period. See "Resolution table" in the introduction. Frequency of the period fp: fp = 1/R * 1/V
PWMDCM	**2**	**xxxx**	**Dutycycle of PWM (Actuator MOTION)**

Name	Length	Fmt	Description
			This value controls the current flow for MOVING the actuator motor. Resolution: 0.01% Range: 0%..100%(0x0000-0x2710)
PWMDCH	**2**	**xxxx**	**Dutycycle of PWM (Actuator HOLD)** This value controls the current flow for MOVING the actuator motor. Resolution: 0.01% Range: 0%..100%(0x0000-0x2710)

Parameter set for VVT Commutation initialization

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>CMTTYP</td><td>1</td><td>xx</td><td>Commutation pattern type
For details pls. refer to picture at start of CMD01.

Table 3. Commutation patterns
<table>
<tr><th>Value</th><th>Pattern</th><th>Valid for</th></tr>
<tr><td>0x00</td><td>Type_180</td><td>VVT3 & VVT4</td></tr>
<tr><td>0x01</td><td>Type_180.1</td><td>VVT3 & VVT4</td></tr>
<tr><td>0x02</td><td>Type_180.2</td><td>VVT3 & VVT4</td></tr>
<tr><td>0x03</td><td>Type_120</td><td>VVT4</td></tr>
<tr><td>0x04</td><td>Type_120.1</td><td>VVT4</td></tr>
<tr><td>0x05</td><td>Type_120.2</td><td>VVT4</td></tr>
</table>
</td></tr>
<tr><td>CMTVEL</td><td>2</td><td>xxxx</td><td>Velocity of actuator motor
Velocity in °/s of actuator motor</td></tr>
<tr><td>CMTRNG</td><td>2</td><td>xxxx</td><td>Actuator Range
Angle in ° of actuator motor to move from 0%-100% of actuator's operating range.</td></tr>
<tr><td>CMTPAU</td><td>2</td><td>xxxx</td><td>Pause time in continuous mode
Pause time to wait before automatically reversing actuation direction in continuous mode.Format: RVVV where R=Resolution, V=Value of period. See "Resolution table" in the introduction.Valid values: 0 to 10000 ms</td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	C0	xx	xx	01

.

02 - VVT3/VVT4 Search End Position

This command moves the servo motor to it's mechanical end position lock and stores this position as end of the operating range. All further motions will be based on this reference point and the initialized RANGE parameter of the initialization.

Tip

In order to ensure that no mechanical parts are damaged by the actuator's motion to the mechanical end position lock, the velocity (VELOC) and current(DCM/DCH) of the actuator motion have to be initialized slow/small enough!

Required previous successful command execution: CMD01 or CMD08(if VVT already actuated)

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	C0	02	01	...

Parameters

Parameter set for search of VVT end position

5	+0	+2
(1..n*)	**ANGLE**	**CFG**

Details

Parameter set for search of VVT end position

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>ANGLE</td><td>2</td><td>xxxx</td><td>Angle in ° of actuator motor's motion</td></tr>
<tr><td>CFG</td><td>1</td><td>xx</td><td>Configuration
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7</td><td>Don't store end position
0_B: Store end position
1_B: Do NOT store end position (manual move only)</td></tr>
<tr><td>6:1</td><td>unused</td></tr>
<tr><td>0</td><td>Direction of motion
0_B: BACKWARDS
1_B: FORWARDS</td></tr>
</table>
</td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	C0	xx	xx	01

.

04 - VVT3/VVT4 Start test cycle

This command initiates the automatic test cycle either continuously, or as single motion. The test step returns as soon as the motion has been initiated.

If continuous mode is selected, the actuator is continuously moved back and forth between 10% and 90% of it's operating range staying at each end position for the configured pause time (PAUSE). This test cycle is then performed as a background operation and does NOT require any more commands to be issued as long the test cycle shall run.

If single motion is selected, the actuator is moved only once from one end (10%/90%) to the other (90%/10%). If single motion command is initiated while a motion is still active, the command will set a flag for the actuator to finish it's current motion and then start a motion to the other direction. Any further call during an active motion will be ignored.

Required previous successful command execution: CMD2

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	C0	04	01	...

Parameters

Parameter set for START of test cycle

5	+0
(1..n*)	**CNTCYCM**

Details

Parameter set for START of test cycle

Name	Length	Fmt	Description
CNTCYCM	**1**	**xx**	**Cycle mode** **Bit(s)** – **Description** **7:1** – **unused** **0** – **Contiuous Cycle Mode** **0_B:** Single motion cycle **1_B:** Continuous motion cycle

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	C0	xx	xx	01

.

08 - VVT3/VVT4 Stop test cycle

This command stops the automatic test cycle.

Since stopping the test cycle allows the motor to free-wheel to a unknown position, the actuator's end position has to be re-searched again before a restart of the test cycle is allowed.

The selection of a regular stop will stop the running test cycle as soon as the current motion is finished and it's end position is reached. The call will return after the end position has been reached.

The selection of a emergency stop stops the current motion immediately by turning OFF the power stages.

Required previous successful command execution: CMD02 or CMD4

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	C0	08	01	...

Parameters

Parameter set for STOP of test cycle

5	+0
(1..n*)	**EMG**

Details

Parameter set for STOP of test cycle

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>EMG</td><td>1</td><td>xx</td><td>Emergency Stop
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:1</td><td>unused</td></tr>
<tr><td>0</td><td>Emergency Stop
0_B: Regular Stop = The actuator will finish it's current motion and then stop.
1_B: Emergency Stop = The actuator will stop immediately</td></tr>
</table>
</td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	C0	xx	xx	01

.

10 - VVT3/VVT4 Status request

This command requests status information of the actuator's current motion and also status of the driver TLE7183. Additionally the state of the TSW is returned.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	C0	10	01

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	C0	xx	xx	01	...

Parameters

Response set of VVT Status request

6	+0	+1	+2	+3
(1..n*)	**MOV**	**POS**	**ERR**	**STAT**

Details

Response set of VVT Status request

Name	Length	Fmt	Description
MOV	**1**	**xx**	**Status of actuator motion** If CMD01 (VVT3/VVT4 Initialization) not performed all bits are retured as 0b1 (MOV=0xFF) **Bit(s)** — **Description** **7:6** — **unused** **5** — **PAU : Actuator paused?** **0_B:** Actuator NOT paused **1_B:** Actuator paused **4** — **STP : Actuator stopped?** **0_B:** Actuator NOT stopped **1_B:** Actuator stopped **3** — **DIR : Dircetion of motion** **0_B:** BACKWARDS **1_B:** FORWARDS **2** — **END : Store End position?** **0_B:** Do NOT Store End position **1_B:** Store End position **1** — **SCM : Single Cycle Mode?** **0_B:** Single Cycle Mode NOT active **1_B:** Single Cycle Mode active **0** — **CCM : Continuous Cycle Mode?**

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td><table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td></td><td>0_B: Continuous Cycle Mode NOT active
1_B: Continuous Cycle Mode active</td></tr></table></td></tr>
<tr><td>POS</td><td>1</td><td>xx</td><td>Position of actuator motion in %

Resolution: 0.5%

Range: 0%-100% (0x00-0xC8)

If CMD01 (VVT3/VVT4 Initialization) not performed POS will be returned as 0xFF.</td></tr>
<tr><td>ERR</td><td>1</td><td>xx</td><td>Status of TLE8183 ERRx Lines
If ERR pins marked as unused at initialization the read result is always low.

Table 4. Error overview
<table><tr><th>ERR2</th><th>ERR1</th><th>Error Type</th></tr><tr><td>0</td><td>0</td><td>No Error</td></tr><tr><td>0</td><td>1</td><td>Overtemperature or overvoltage error</td></tr><tr><td>1</td><td>0</td><td>Short circuit detection or overcurrent error</td></tr><tr><td>1</td><td>1</td><td>Undervoltage</td></tr></table>
Tip
In case of a reported error of the TLE7183, the driver automatically shuts down the power stages. In order to restart a test cycle the test step requires new initialization.

If CMD01 (VVT3/VVT4 Initialization) not performed ERR will be returned as 0xFF.
<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>7:2</td><td>unused</td></tr><tr><td>1</td><td>Status of ERR2 line</td></tr><tr><td>0</td><td>Status of ERR1 line</td></tr></table></td></tr>
<tr><td>STAT</td><td>1</td><td>xx</td><td>State of TSW

Shows current state of TSW

For details please refer to diagram "VVT3/VVT4 State machine control for TSW_C0"</td></tr>
</table>

Name	Length	Fmt	Description	
			No.	**State name**
			0	PREINIT
			1	INIT
			2	ENDPOS
			3	RUN

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**Requested command not allowed in current TSW state. For details about the state machine please refer to diagram "VVT3/ VVT4 State machine control for TSW_C0"**
0x02	**GTM setting for PWM setup not valid**
0x03	**PWM Period to long**
0x04	**Definition of PWM PINs (IL1 to IL3, IH1 to IH3) is invalid, GTM or SPE connection not possible**
0x05	**Error at interrupt (de)registration**
0x06	**Definition of at least one ERRx PIN is invalid**
0x07	**Pause time (CMTPAU) is out of range**
0xFF	**HAL busy**
0xFE	**Generic HAL error**
0xFD	**HAL not compatible to current CPU**
0xFC	**Invalid params**
0xFB	**HAL module not implemented**
0xFA	**HAL function not implemented**
0xF9	**HAL timeout**
0xEF	**HAL-PWM: Timer overflow**

TSW_D0 - eMGS625 (DGDI-S) - Programmable injection device

Test step description

Date: 31.08.2016

Created: 18.01.2017 09:12.

Purpose: Test of DGDI programmable injection device, for control of power MOSFETs driving solenoid valves and electro-injectors.

Important: before use, initialize ports, SPI and setup clock for DGDI.

fixed SPI frame configuration: 16 bits per frame, burst mode active.

Test step commands

.

00 - Initialise DGDI

This command is used to initialise the communication to DGDI.

Table 1. Initialisation values of DGDI register written by command "Initialise/Reset SPI controller to DGDI"

Register	Address	Init Value
ck_per	0x01C0	0x0003
flags_direction	0x01C1	0x0000
flags_polarity	0x01C2	0x0000
flags_source	0x01C3	0x1000
ck_ofscmp_per	0x01C4	0x0000
spi_config	0x01C8	0x0003
backup_clock_status_reg	0x01C7	0x0000
driver_config	0x01C5	0x4000
reset_behaviour	0x01CE	0x01FF
hs1_diag_config_1	0x0153	0x2150
hs1_diag_config_2	0x0154	0x0008
hs2_diag_config_1	0x0156	0x2150
hs2_diag_config_2	0x0157	0x0008
hs3_diag_config_1	0x0159	0x2150
hs3_diag_config_2	0x015A	0x0008
hs4_diag_config_1	0x015C	0x2150
hs4_diag_config_2	0x015D	0x0008
hs5_diag_config_1	0x015F	0x2150
hs5_diag_config_2	0x0160	0x0008

Register	Address	Init Value
vds_threshold_hs	0x018A	0x36DB
ls1_diag_config_1	0x0140	0x2150
ls1_diag_config_2	0x0141	0x0008
ls2_diag_config_1	0x0143	0x2150
ls2_diag_config_2	0x0144	0x0008
ls3_diag_config_1	0x0146	0x2150
ls3_diag_config_2	0x0147	0x0008
ls4_diag_config_1	0x0149	0x2150
ls4_diag_config_2	0x014A	0x0008
ls5_diag_config_1	0x014C	0x2150
ls5_diag_config_2	0x014D	0x0008
ls6_diag_config_1	0x014F	0x2150
ls6_diag_config_2	0x0150	0x0008
vds_threshold_ls_1	0x018C	0x2492
vds_threshold_ls_2	0x018D	0x0002
fbk_sens_seq0_ch1	0x0180	0x0000
fbk_sens_seq1_ch1	0x0181	0xFD55
fbk_sens_seq0_ch2	0x0182	0x0000
fbk_sens_seq1_ch2	0x0183	0x0000
out_acc_seq0_ch1	0x0184	0x0FFF
out_acc_seq1_ch1	0x0185	0x0FFF
out_acc_seq0_ch2	0x0186	0x0000
out_acc_seq1_ch2	0x0187	0x0800
hs12_act	0x01A6	0xFFFF/0x7F7F
hs34_act	0x01A7	0xFFFF/0x7F7F
hs5_act	0x01A8	0x00FF/0x007F
Channel 2 DRAM50	0x0072	0x0000
Channel 2 DRAM51	0x0073	0x0000

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D0	00	01	...

Parameters

parameters to select a SPI controller for DGDI

5	+0	+1	+2	+3
(1*)	**PORTMT**	**PINMT**	**PORTMR**	**PINMR**

...	+4	+5	+6	+7
(1*)	**PORTCK**	**PINCK**	**PORTSL**	**PINSL**

optional parameter of initialise DGDI command

13	+0	+2
(0..n*)	**ADR**	**VAL**

Details

parameters to select a SPI controller for DGDI
Port and Pin parameters to select the SPI controller where DGDI is connected

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

optional parameter of initialise DGDI command
This can be used to initialise more registers or to change register values.

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>ADR</td><td>2</td><td>xxxx</td><td>Register address
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>15:10</td><td>reserved, must be zero</td></tr>
<tr><td>9:0</td><td>register address</td></tr>
</table></td></tr>
<tr><td>VAL</td><td>2</td><td>xxxx</td><td>Register value
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>15:0</td><td>register value</td></tr>
</table></td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	D0	xx	xx	01

.

01 - Load microcode

This command is used to download microcode data to DGDI.

Example (CFITerm):

LOADFILE DGDI_uCode.hex,,,45,80D00101ssttuuvvwwxxyyzzcc,C0D0000001,ff,last

downloads the microcode in file DGDI_uCode.hex to the DGDI connected to SPI with Ports/Pins "ssttuuvvwwxxyyzz" into CRAM of Channel part of "cc". The Flash enable is set after the download - part of "cc".

NOTE: Ensure that parameter Segmentlength in LOADFILE (#4) is 45!

Table 2. Initialisation values of DGDI register written by command "Load microcode"

Register	Address	Init Value
unlock_word	0x0103/0x0123	0xBEEF
code_width	0x0107/0x0127	taken from Microcode file header
checksum_h	0x0108/0x0128	taken from Microcode file header
checksum_l	0x0109/0x0129	taken from Microcode file header
seq0_entry_point	0x010A/0x012A	0x0000
seq1_entry_point	0x010B/0x012B	0x0000
diag_routine_addr	0x010C/0x012C	0x0082
driver_disabled_routine_addr	0x010D/0x012D	0x0104
sw_interrupt_routine_addr	0x010E/0x012E	0xC180(Ch1)/0x3006(Ch2)
flash_enable	0x0100/0x0120	0x0018 (setting depends on par CHAN byte 5 bit 2)
backup_clock_status_reg	0x01C7	0x0002

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D0	01	01	...

Parameters

parameters to select a SPI controller for DGDI

5	+0	+1	+2	+3
(1*)	**PORTMT**	**PINMT**	**PORTMR**	**PINMR**

...	+4	+5	+6	+7
(1*)	**PORTCK**	**PINCK**	**PORTSL**	**PINSL**

para. for channel

13	+0
(1*)	**CHAN**

para. for load microcode

14	+0
(1..n*)	DATA

Details

parameters to select a SPI controller for DGDI
Port and Pin parameters to select the SPI controller where DGDI is connected

Name	Length	Fmt	Description
PORTMT	1	xx	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	1	xx	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	1	xx	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	1	xx	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORTCK	1	xx	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	1	xx	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	1	xx	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PINSL	1	xx	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

para. for channel
parameters for channel configurarion

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>CHAN</td><td>1</td><td>xx</td><td>Channel configuration (bitcoded)
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>0</td><td>1: load Microcode to CRAM of Channel 1</td></tr>
<tr><td>1</td><td>1: load Microcode to CRAM of Channel 2</td></tr>
<tr><td>2</td><td>1: skip setting of Flash enable</td></tr>
<tr><td>3</td><td>1: load merged Microcode file (contain microcode for channels 1 and 2)</td></tr>
<tr><td>4</td><td>1: activate ciphering for load Microcode to CRAM of channel 1/2</td></tr>
<tr><td>7:5</td><td>reserved</td></tr>
</table></td></tr>
</table>

para. for load microcode

parameters for channel configurarion and microcode

Name	Length	Fmt	Description
DATA	**1**	**xx..**	**Microcode data** 2 - 32 data Bytes. Download of HEX file per "loadfile" function of cfiterm. Segmentlength = Prefix + 32 byte of Data

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	D0	xx	xx	01

.

02 - Read status

This command is used to readout the status of the microcode data from DGDI

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D0	02	01	...

Parameters

parameters to select a SPI controller for DGDI

5	+0	+1	+2	+3
(1*)	**PORTMT**	**PINMT**	**PORTMR**	**PINMR**

...	+4	+5	+6	+7
(1*)	**PORTCK**	**PINCK**	**PORTSL**	**PINSL**

para. for read status

13	+0
(1*)	**CHNR**

Details

parameters to select a SPI controller for DGDI
Port and Pin parameters to select the SPI controller where DGDI is connected

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal

Name	Length	Fmt	Description
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

para. for read status

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>CHNR</td><td>1</td><td>xx</td><td>channel number
for read of microcode status in CRAM
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>0</td><td>1: Microcode status from CRAM of Channel 1</td></tr>
<tr><td>1</td><td>1: Microcode status from CRAM of Channel 2</td></tr>
<tr><td>6:2</td><td>reserved</td></tr>
<tr><td>7</td><td>Presentation of extended diagnosis info from Microcode
0_B: disabled
1_B: enabled</td></tr>
</table>
</td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D0	xx	xx	01	...

Parameters

response of command "Read Status"

6	+0	+2	+3	+5	+7	+9
(1*)	**ERROR**	**VER**	**CRCH**	**CRCL**	**LEN**	**FLASH**

...	+11	+13
(1*)	**CLOCK**	**START**

Details

response of command "Read Status"

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>ERROR</td><td>2</td><td>xxxx</td><td>Error Status DGDI (bitcoded)

Note: Bit 7 - 15 only valid if Presentation of extended diagnosis info is enabled (command byte 7 bit 7)

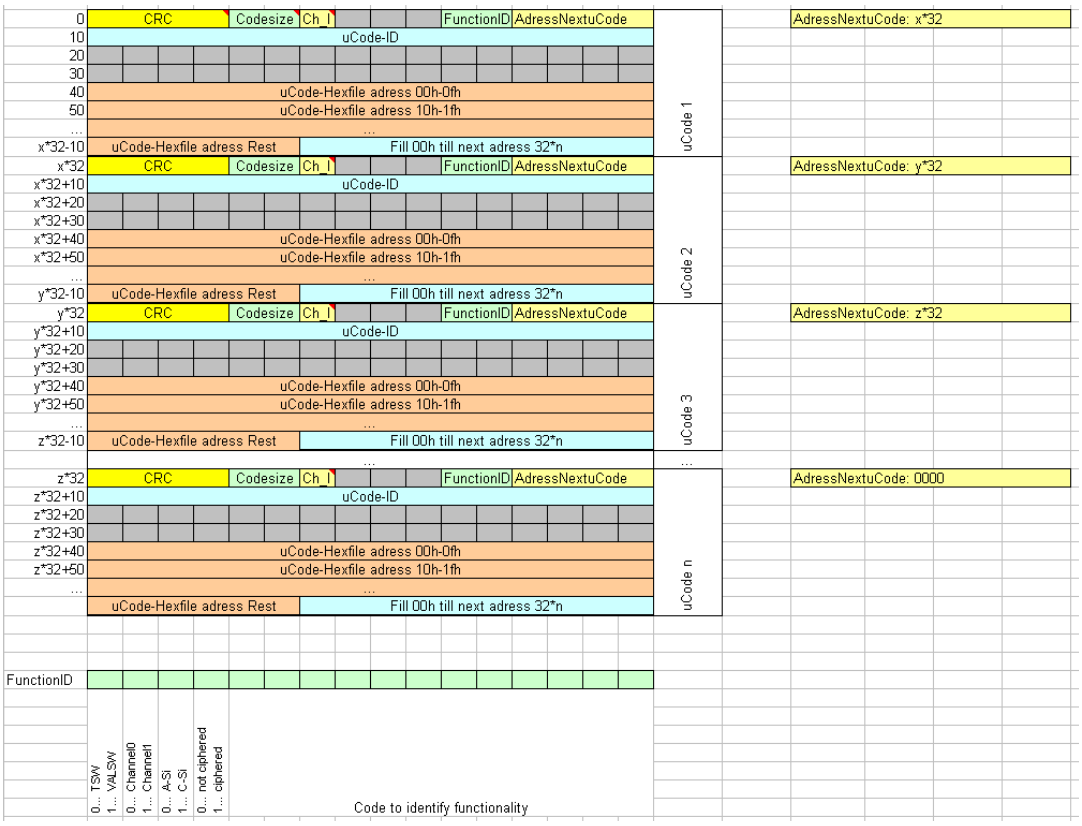
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>0</td><td>1: TSW not initialised</td></tr>
<tr><td>1</td><td>1: Microcode is not ready</td></tr>
<tr><td>2</td><td>1: Microcode has checksum error</td></tr>
<tr><td>3</td><td>1: DGDI has loss of external clock</td></tr>
<tr><td>6:4</td><td>reserved</td></tr>
<tr><td>7</td><td>1: DCDC-parameter CRC failed</td></tr>
<tr><td>8</td><td>1: Unimplemented test requested channel 1</td></tr>
<tr><td>9</td><td>1: Unimplemented test requested channel 2</td></tr>
<tr><td>10</td><td>1: Unexpected SW-Interrupt channel 1</td></tr>
<tr><td>11</td><td>1: Unexpected SW-Interrupt channel 2</td></tr>
<tr><td>12</td><td>1: Unexpected diagnosis-Interrupt channel 1</td></tr>
<tr><td>13</td><td>1: Unexpected diagnosis-Interrupt channel 2</td></tr>
<tr><td>14</td><td>1: Unexpected driver-disable-Interrupt channel 1</td></tr>
<tr><td>15</td><td>1: Unexpected driver-disable-Interrupt channel 2</td></tr>
</table></td></tr>
<tr><td>VER</td><td>1</td><td>xx..</td><td>Version DGDI microcode
16 bytes of version information.</td></tr>
<tr><td>CRCH</td><td>2</td><td>xxxx</td><td>DGDI register checksum high</td></tr>
<tr><td>CRCL</td><td>2</td><td>xxxx</td><td>DGDI register checksum low</td></tr>
<tr><td>LEN</td><td>2</td><td>xxxx</td><td>DGDI register code_width</td></tr>
<tr><td>FLASH</td><td>2</td><td>xxxx</td><td>DGDI register flash_enable</td></tr>
<tr><td>CLOCK</td><td>2</td><td>xxxx</td><td>DGDI register backup_clock_status_reg</td></tr>
<tr><td>START</td><td>2</td><td>xxxx</td><td>DGDI register start_config_reg</td></tr>
</table>

.

03 - Load microcode from ECU memory area

This command is used to download microcode data from ECU memory area to DGDI.

Figure 1. structure of microcode(s) in memory



The specified address needs to contain at least one valid microcode hex file including it’s header information which contains length, version, checksum, Function-ID etc. (see figure "structure of microcode(s) in memory").

The specified address may contain a list of multiple different microcode files which are linked through their header information. Any individual file is identified by the start address of the list and the Function-ID of the microcode. (see figure "structure of microcode(s) in memory").

The search for the specified Function-ID of the microcode will start at the specified address and trace through the list of available microcode files within the list until the specified Function ID is found. If no matching Function-ID is found, the test step returns an error.

The test step will download the microcode only, if the code length in the header block is between 0 and maximum number of CRAM cells.

In condition of a verification failure, the test step returns an error.

After the download, the DGDI device checks the downloaded microcode against the stored checksum to verify correct download to the device.

In condition of a verification failure of the DGDI device, the test step returns an error.

In order to check the activation status of the downloaded microcode after a download, the CRAM status should be checked for every affected channel! (see command "Read status").

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D0	03	01	...

Parameters

parameters to select a SPI controller for DGDI

5	+0	+1	+2	+3
(1*)	**PORTMT**	**PINMT**	**PORTMR**	**PINMR**

...	+4	+5	+6	+7
(1*)	**PORTCK**	**PINCK**	**PORTSL**	**PINSL**

para. for Chan, Addr and fct-ID

13	+0	+1	+5
(1*)	**CHAN**	**ADDR**	**FUNC_ID**

Details

parameters to select a SPI controller for DGDI
Port and Pin parameters to select the SPI controller where DGDI is connected

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

para. for Chan, Addr and fct-ID
parameters for Channel, Address and function ID

Name	Length	Fmt	Description
CHAN	**1**	**xx**	**Channel configuration (bitcoded)**

Bit(s)	Description
0	**1: load Microcode to CRAM of Channel 1**

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td><table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>1</td><td>1: load Microcode to CRAM of Channel 2</td></tr>
<tr><td>2</td><td>1: skip setting of Flash enable</td></tr>
<tr><td>7:3</td><td>reserved</td></tr>
</table></td></tr>
<tr><td>ADDR</td><td>4</td><td>xxxxxxxx</td><td>Start address of microcode data area
Physical address of ECU memory (Flash/RAM) which contains microcode image(s).</td></tr>
<tr><td>FUNC_ID</td><td>2</td><td>xxxx</td><td>Function ID of microcode to be downloaded
unique search pattern to specify microcode within list of available microcodes in memory</td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	D0	xx	xx	01

.

80 - Offset Compensation

This command is used to start the microcode in DGDI for offset compensation of the current measurement blocks.

> **Note:** The offset compensation has to be done for every used current measurement channels difference amplifier with a separat call of command "Offset Compensation". No signal must be applied to the amplifier while the calibration is done.
>
> If the calibration is ommitted or called while the difference amplifier input signal level is not zero, the accuracy of the input amplifier will be very low.

Table 3. Initialisation values of DGDI register written by command "Offset Compensation"

Register	Address	Init Value
cur_access_reg1	0x0188	0x001F
cur_access_reg2	0x0189	0x0000
oa_out1	0x01AA	0
oa_out2	0x01AB	0
ctrl_req_seq0	0x0101	Bit 2 set to start Microcode
DRAM0	0x0000	taken from command parameter
ck_ofscmp_per	0x01C4	0x0047

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D0	80	01	...

Parameters

parameters to select a SPI controller for DGDI

5	+0	+1	+2	+3
(1*)	**PORTMT**	**PINMT**	**PORTMR**	**PINMR**

...	+4	+5	+6	+7
(1*)	**PORTCK**	**PINCK**	**PORTSL**	**PINSL**

para. for timeout and offset values

13	+0	+2
(1*)	**TIM**	**COMP**

Details

parameters to select a SPI controller for DGDI
Port and Pin parameters to select the SPI controller where DGDI is connected

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

para. for timeout and offset values
parameters for time out and Offset compensation parameters

Name	Length	Fmt	Description
TIM	**2**	**xxxx**	**DGDI timeout** Selection of timeout on waiting for DGDI test status ready.
COMP	**2**	**xxxx**	**Offset compensation parameters**

Bit(s)	Description
3:0	**not used, must be set to 0**

Name	Length	Fmt	Description	
			Bit(s)	**Description**
			5:4	**Select amplification for current measurement** **00_B:** 5,79 **01_B:** 8,68 **10_B:** 12,53 **11_B:** 19,25
			7:6	**not used, must be set to 0**
			10:8	**Select current measurement channel** **000_B:** not used **001_B:** Channel 1 **010_B:** Channel 2 **011_B:** Channel 3 **100_B:** Channel 4
			15:11	**not used, must be set to 0**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D0	xx	xx	01	...

Parameters

Error response of command "Offset Compensation"

6	+0	+2	+4	+6	+8
(1*)	**ERROR**	**OFF1**	**OFF2**	**OFF3**	**OFF4**

Details

Error response of command "Offset Compensation"

Name	Length	Fmt	Description	
ERROR	**2**	**xxxx**	**Error Status DGDI (bitcoded)**	
			Bit(s)	**Description**
			0	**1: TSW not initialised**
			1	**1: Microcode is not ready**
			2	**1: Microcode has checksum error**
			3	**1: DGDI has loss of external clock**
			6:4	**reserved**
			7	**1: DCDC-parameter CRC failed**
			8	**1: Unimplemented test requested channel 1**

Name	Length	Fmt	Description	
			Bit(s)	**Description**
			9	**1: Unimplemented test requested channel 2**
			10	**1: Unexpected SW-Interrupt channel 1**
			11	**1: Unexpected SW-Interrupt channel 2**
			12	**1: Unexpected diagnosis-Interrupt channel 1**
			13	**1: Unexpected diagnosis-Interrupt channel 2**
			14	**1: Unexpected driver-disable-Interrupt channel 1**
			15	**1: Unexpected driver-disable-Interrupt channel 2**
OFF1	**2**	**xxxx**	**DGDI register offset1** Offset value for measurement block 1	
OFF2	**2**	**xxxx**	**DGDI register offset2** Offset value for measurement block 2	
OFF3	**2**	**xxxx**	**DGDI register offset3** Offset value for measurement block 3	
OFF4	**2**	**xxxx**	**DGDI register offset4** Offset value for measurement block 4	

.

81 - Initialise and start LSHS test

This command is used to initialise the configuration and to start the microcode in DGDI for LSHS test.

Table 4. Initialisation values of DGDI register written by command "Initialise and start LSHS test"

Register	Address	Init Value
slewrate_hs	0x018E	0x02AA
slewrate_ls	0x018F	previous value or 0x0FFF
flags_source	0x01C3	0x1000
start_config_reg	0x0104	taken from command parameter
cur_access_reg1	0x0188	0x001F
cur_access_reg2	0x0189	0x0018
oa_out1	0x01AA	taken from command parameter
oa_out2	0x01AB	taken from command parameter
meas_func1	0x01AC	taken from command parameter
ctrl_reg_seq0	0x0101	Bit 3 set to start Microcode
bias_regfile	0x01A4	taken from command parameter
DRAM0-DRAM13	0x0000-0x000D	taken from command parameter

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D0	81	01	...

Parameters

parameters to select a SPI controller for DGDI

5	+0	+1	+2	+3
(1*)	**PORTMT**	**PINMT**	**PORTMR**	**PINMR**

...	+4	+5	+6	+7
(1*)	**PORTCK**	**PINCK**	**PORTSL**	**PINSL**

para. for LSHS tests

13	+0	+2	+4
(1*)	**TIM**	**GENIO**	**GENM**

para. for LS and HS

19	+0
(1*)	**DRIV**

Details

parameters to select a SPI controller for DGDI
Port and Pin parameters to select the SPI controller where DGDI is connected

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

para. for LSHS tests

parameters for LSHS tests

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>TIM</td><td>2</td><td>xxxx</td><td>DGDI timeout
Selection of timeout on waiting for DGDI test status ready.</td></tr>
<tr><td>GENIO</td><td>2</td><td>xxxx</td><td>General DGDI input/output configuration parameters
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>2:0</td><td>Select sensitivity to start signals

000_B: not used
001_B: Start1
010_B: Start2
011_B: Start3
100_B: Start4
101_B: Start5
110_B: Start6</td></tr>
<tr><td>3</td><td>not used, must be set to 0</td></tr>
<tr><td>5:4</td><td>Gain setting of OAx amplifier

00_B: 1,33
01_B: 2,00
10_B: 3,00
11_B: 5,33</td></tr>
<tr><td>7:6</td><td>not used, must be set to 0</td></tr>
<tr><td>9:8</td><td>Select OAx for current measurement

00_B: OA1, OA2 high impedance
01_B: OA1 current measurement, OA2 high impedance
10_B: OA1 high impedance, OA2 current measurement
11_B: OA1, OA2 high impedance</td></tr>
<tr><td>15:10</td><td>not used, must be set to 0</td></tr>
</table></td></tr>
<tr><td>GENM</td><td>2</td><td>xxxx</td><td>General DGDI measurement configuration parameters
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>3:0</td><td>not used, must be set to 0</td></tr>
<tr><td>5:4</td><td>Select amplification for current measurement

00_B: 5,79
01_B: 8,68
10_B: 12,53
11_B: 19,25</td></tr>
</table></td></tr>
</table>

Name	Length	Fmt	Description
			Bit(s) / **Description**
			7:6 — **not used, must be set to 0**
			10:8 — **Select current measurement channel** **000_B:** not used **001_B:** Channel 1 **010_B:** Channel 2 **011_B:** Channel 3 **100_B:** Channel 4
			15:11 — **not used, must be set to 0**

para. for LS and HS
parameters for 6 LS and 5 HS

Name	Length	Fmt	Description
DRIV	**2**	**xxxx..**	**DGDI pre-driver configuration parameters** DGDI pre-driver configuration parameters for LS1 (exemplary for the following parameters) Repeat bytes 19 - 20 for all LS and HS predrivers of DGDI. The required sequence is LS2,LS3,LS4,LS5,LS6,HS1,HS2,HS3,HS4,HS5.

Bit(s)	Description
0	**Dependence on Start signal** **0_B:** Pre-driver is set to state according to Pre-driver level (Bit 2) **1_B:** Pre-driver is switched with Start signal **Note:** Start signal is selected with GENIO bit 0-2
1	**Inversion to state of Start signal** **0_B:** Pre-driver is not inverted to state of Start signal **1_B:** Pre-driver is inverted to state of Start signal
2	**Pre-driver level** **0_B:** Off **1_B:** On
3	**Bias** **0_B:** Off **1_B:** On
15:4	**not used, must be set to 0**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D0	xx	xx	01	...

Parameters

Error response of command "Initialise and start LSHS test"

6	+0
(1*)	**ERROR**

Details

Error response of command "Initialise and start LSHS test"

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>ERROR</td><td>2</td><td>xxxx</td><td>Error Status DGDI (bitcoded)
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>0</td><td>1: TSW not initialised</td></tr>
<tr><td>1</td><td>1: Microcode is not ready</td></tr>
<tr><td>2</td><td>1: Microcode has checksum error</td></tr>
<tr><td>3</td><td>1: DGDI has loss of external clock</td></tr>
<tr><td>6:4</td><td>reserved</td></tr>
<tr><td>7</td><td>1: DCDC-parameter CRC failed</td></tr>
<tr><td>8</td><td>1: Unimplemented test requested channel 1</td></tr>
<tr><td>9</td><td>1: Unimplemented test requested channel 2</td></tr>
<tr><td>10</td><td>1: Unexpected SW-Interrupt channel 1</td></tr>
<tr><td>11</td><td>1: Unexpected SW-Interrupt channel 2</td></tr>
<tr><td>12</td><td>1: Unexpected diagnosis-Interrupt channel 1</td></tr>
<tr><td>13</td><td>1: Unexpected diagnosis-Interrupt channel 2</td></tr>
<tr><td>14</td><td>1: Unexpected driver-disable-Interrupt channel 1</td></tr>
<tr><td>15</td><td>1: Unexpected driver-disable-Interrupt channel 2</td></tr>
</table></td></tr>
</table>

.

82 - Initialise and start LSHS test add. LS7

This command is used to initialise the configuration and to start the microcode in DGDI for LSHS test. like command "Initialise and start LSHS test" with additional parameter for LS7

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D0	82	01	...

Parameters

para. for LS and HS

5	+0
(1*)	**DRIV**

Details

para. for LS and HS
parameters for 6 LS and 5 HS and LS7

Name	Length	Fmt	Description
DRIV	**2**	**xxxx..**	**DGDI pre-driver configuration parameters** 2 Bytes each for all 6 LS, 5 HS and LS7 predrivers of DGDI. The required sequence is LS1,LS2,LS3,LS4,LS5,LS6,HS1,HS2,HS3,HS4,HS5,LS7.

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	D0	xx	xx	01

.

83 - DCDC converter setup

This command is used to initialise the configuration in DGDI for DCDC converter test.

Table 5. Initialisation values of DGDI register written by command "DCDC converter setup"

Register	Address	Init Value
dac4l	0x01A1	taken from command parameter
dac4h	0x01A2	taken from command parameter
dac4neg	0x01A3	taken from command parameter
boost_dac	0x019B	taken from command parameter
slewrate_ls	0x018F	taken from command parameter
cur_access_reg1	0x0188	0x001F
cur_access_reg2	0x0189	0x0018
oa_out1	0x01AA	taken from command parameter
oa_out2	0x01AB	taken from command parameter
meas_func1	0x01AC	taken from command parameter
ctrl_reg_seq0	0x0121	Bit 0 set to start Microcode
counter_34_prescaler	0x0131	0x5E00

Register	Address	Init Value
DRAM0-DRAM3	0x0040-0x0043	taken from command parameter
DRAM50	0x0072	taken from command parameter

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D0	83	01	...

Parameters

parameters to select a SPI controller for DGDI

5 (1*)	+0	+1	+2	+3
	PORTMT	**PINMT**	**PORTMR**	**PINMR**

... (1*)	+4	+5	+6	+7
	PORTCK	**PINCK**	**PORTSL**	**PINSL**

para. for DCDC setup

13 (1*)	+0	+2	+4	+6
	TIM	**CURM**	**CURL**	**CURN**

... (1*)	+8	+10	+12	+14
	BOOSTV	**TMIN**	**TMAX**	**SLEW**

Details

parameters to select a SPI controller for DGDI
Port and Pin parameters to select the SPI controller where DGDI is connected

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

para. for DCDC setup
parameters for DCDC setup

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>TIM</td><td>2</td><td>xxxx</td><td>DGDI timeout
Selection of timeout on waiting for DGDI test status ready.</td></tr>
<tr><td>CURM</td><td>2</td><td>xxxx</td><td>General DGDI current measurement parameters
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>1:0</td><td>Select amplification for current measurement
00_B: 5,79
01_B: 8,68
10_B: 12,53
11_B: 19,25
Attention: This gain is use for current control. Use only gain 12,53 or 19,25</td></tr>
<tr><td>3:2</td><td>not used, must be set to 0</td></tr>
<tr><td>5:4</td><td>Gain setting of OAx amplifier
00_B: 1,33
01_B: 2,00
10_B: 3,00
11_B: 5,33
Note: Value is only relevant, if current is measured at OAx pin.</td></tr>
<tr><td>7:6</td><td>not used, must be set to 0</td></tr>
<tr><td>9:8</td><td>Select OAx for current measurement
00_B: OA1, OA2 high impedance
01_B: OA1 current measurement, OA2 high impedance
10_B: OA1 high impedance, OA2 current measurement
11_B: OA1, OA2 high impedance</td></tr>
<tr><td>15:10</td><td>not used, must be set to 0</td></tr>
</table></td></tr>
<tr><td>CURL</td><td>2</td><td>xxxx</td><td>Current limit parameters for 2-point regulation
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:0</td><td>not used, must be set to 0</td></tr>
<tr><td>15:8</td><td>DAC value to select max current for 2-point regulation (dac4h register 1A2h)</td></tr>
</table></td></tr>
</table>

Name	Length	Fmt	Description
			Bit(s) / **Description** **Note:** The DAC value for the max current need to be calculated with formular: $DAC\_value = \frac{I_{\max} * G_{DA\_diff} * R_{Shunt} + 250mV}{9{,}77mV}$ **Attention:** Only values below 6 Ampere should be configured (but depends on hardware)!
CURN	**2**	**xxxx**	**Current limit parameters** **Bit(s)** / **Description** **3:0** — **DAC value to select Booster negativ over current (dac4neg register 1A3h).This value limits the current when the Booster is decharged.** **Note:** The DAC value for the booster negativ over current need to be calculated with formular: $DAC\_value = \frac{I_{boostover} * -2 * R_{Shunt} + 250mV}{156{,}25mV}$ **Note:** Handling of parameter is not supported in current Microcode version. **7:4** — **not used, must be set to 0** **15:8** — **DAC value to select min current for 2-point regulation (dac4l register 1A1h)** **Note:** The DAC value for the min current need to be calculated with formular: $DAC\_value = \frac{I_{\min} * G_{DA\_diff} * R_{Shunt} + 250mV}{9{,}77mV}$
BOOSTV	**2**	**xxxx**	**Booster voltage parameters** **Bit(s)** / **Description** **7:0** — **DAC value to select Booster Voltage (boostdac register 19Bh)** **Note:** The DAC value for the booster voltage need to be calculated with formular: $DAC\_value = \frac{V_{boost}}{313mV}$

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td><table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td></td><td>Note: Only values between 0x08 (2,5V) and 0xD0 (65V) are allowed</td></tr><tr><td>15:8</td><td>not used, must be set to 0</td></tr></table></td></tr>
<tr><td>TMIN</td><td>2</td><td>xxxx</td><td>Minimum charge time
Minimum time until booster voltage is reached.
Resolution is 1µs/LSB</td></tr>
<tr><td>TMAX</td><td>2</td><td>xxxx</td><td>Maximum charge time
Maximum time until booster voltage is reached.
Resolution is 2.5µs/LSB</td></tr>
<tr><td>SLEW</td><td>2</td><td>xxxx</td><td>Slew rate setting for LS7<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>0</td><td>Select value for R DSON of PMOS transistor to configure slew rate for switching on LS7. Value in Ohm.
0_B: 5.0 (1500V/µs)
1_B: 14.6 (300V/µs)</td></tr><tr><td>1</td><td>reserved, must be set to 0</td></tr><tr><td>3:2</td><td>not used, must be set to 0</td></tr><tr><td>4</td><td>Select value for R DSON of NMOS transistor to configure slew rate for switching off LS7. Value in Ohm.
0_B: 1.1 (1500V/µs)
1_B: 5.9 (300V/µs)</td></tr><tr><td>5</td><td>reserved, must be set to 0</td></tr><tr><td>15:6</td><td>not used, must be set to 0</td></tr></table></td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D0	xx	xx	01	...

Parameters

Error response of command "DCDC converter setup"

...	+0
(1*)	**ERROR**

Details

Error response of command "DCDC converter setup"

Name	Length	Fmt	Description
ERROR	**2**	**xxxx**	**Error Status (bitcoded)**

Name	Length	Fmt	Description	
			Bit(s)	**Description**
			0	**1: TSW not initialised**
			1	**1: Microcode is not ready**
			2	**1: Microcode has checksum error**
			3	**1: DGDI has loss of external clock**
			6:4	**reserved**
			7	**1: DCDC-parameter CRC failed**
			8	**1: Unimplemented test requested channel 1**
			9	**1: Unimplemented test requested channel 2**
			10	**1: Unexpected SW-Interrupt channel 1**
			11	**1: Unexpected SW-Interrupt channel 2**
			12	**1: Unexpected diagnosis-Interrupt channel 1**
			13	**1: Unexpected diagnosis-Interrupt channel 2**
			14	**1: Unexpected driver-disable-Interrupt channel 1**
			15	**1: Unexpected driver-disable-Interrupt channel 2**

.

84 - DCDC converter switching

This command is used to switch the DCDC converter on or off.

Table 6. Initialisation values of DGDI register written by command "DCDC converter switching"

Register	Address	Init Value
ctrl_reg_seq0	0x0121	Bit 1 set to start Microcode
DRAM4	0x0044	taken from command parameter

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D0	84	01	...

Parameters

parameters to select a SPI controller for DGDI

5	+0	+1	+2	+3
(1*)	**PORTMT**	**PINMT**	**PORTMR**	**PINMR**

...	+4	+5	+6	+7
(1*)	**PORTCK**	**PINCK**	**PORTSL**	**PINSL**

para. for DCDC conv on/off

13	+0	+2
(1*)	**TIM**	**DCDC**

Details

parameters to select a SPI controller for DGDI
Port and Pin parameters to select the SPI controller where DGDI is connected

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

para. for DCDC conv on/off
parameters for DCDC converter switching

Name	Length	Fmt	Description
TIM	**2**	**xxxx**	**DGDI timeout** Selection of timeout on waiting for DGDI test status ready.
DCDC	**2**	**xxxx**	**Switching DCDC converter**

Bit(s)	Description
0	**Select status of DCDC converter** **0_B:** Off **1_B:** On
3:1	**not used, must be set to 0**
4	**Time measurement for reaching Booster V before minimum Booster charge time**

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td></td><td>Bit(s)</td><td>Description</td></tr>
<tr><td></td><td></td><td></td><td></td><td>0_B: Off
1_B: On

Note: The minimum booster charge time is defined with command "DCDC converter setup" in parameter TMIN.</td></tr>
<tr><td></td><td></td><td></td><td>15:5</td><td>not used, must be set to 0</td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D0	xx	xx	01	...

Parameters

Error response of command "DCDC converter switching"

6	+0	+2
(1*)	**ERROR**	**STAT**

Details

Error response of command "DCDC converter switching"

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th colspan="2">Description</th></tr>
<tr><td>ERROR</td><td>2</td><td>xxxx</td><td colspan="2">Error Status DGDI (bitcoded)</td></tr>
<tr><td></td><td></td><td></td><td>Bit(s)</td><td>Description</td></tr>
<tr><td></td><td></td><td></td><td>0</td><td>1: TSW not initialised</td></tr>
<tr><td></td><td></td><td></td><td>1</td><td>1: Microcode is not ready</td></tr>
<tr><td></td><td></td><td></td><td>2</td><td>1: Microcode has checksum error</td></tr>
<tr><td></td><td></td><td></td><td>3</td><td>1: DGDI has loss of external clock</td></tr>
<tr><td></td><td></td><td></td><td>6:4</td><td>reserved</td></tr>
<tr><td></td><td></td><td></td><td>7</td><td>1: DCDC-parameter CRC failed</td></tr>
<tr><td></td><td></td><td></td><td>8</td><td>1: Unimplemented test requested channel 1</td></tr>
<tr><td></td><td></td><td></td><td>9</td><td>1: Unimplemented test requested channel 2</td></tr>
<tr><td></td><td></td><td></td><td>10</td><td>1: Unexpected SW-Interrupt channel 1</td></tr>
<tr><td></td><td></td><td></td><td>11</td><td>1: Unexpected SW-Interrupt channel 2</td></tr>
<tr><td></td><td></td><td></td><td>12</td><td>1: Unexpected diagnosis-Interrupt channel 1</td></tr>
<tr><td></td><td></td><td></td><td>13</td><td>1: Unexpected diagnosis-Interrupt channel 2</td></tr>
</table>

Name	Length	Fmt	Description	
			Bit(s)	**Description**
			14	**1: Unexpected driver-disable-Interrupt channel 1**
			15	**1: Unexpected driver-disable-Interrupt channel 2**
STAT	**2**	**xxxx**	**Status of test responsed from microcode (bitcoded)**	
			Bit(s)	**Description**
			0	**1: Booster voltage is reached before Tmin**
			1	**1: Booster voltage is not reached within Tmax**
			15:2	**reserved**

.

85 - ADC test

This command is used to read out the ADC measurement values of the current measurement blocks.

Table 7. Initialisation values of DGDI register written by command "ADC test"

Register	Address	Init Value
cur_access_reg1	0x0188	0x0007
ck_ofscmp_per	0x01C4	0x0047
oa_out1	0x01AA	taken from command parameter
oa_out2	0x01AB	taken from command parameter
meas_func1	0x01AC	taken from command parameter
ctrl_reg_seq0	0x0101	Bit 6 set to start Microcode
DRAM0-DRAM1	0x0000-0x0001	taken from command parameter

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D0	85	01	...

Parameters

parameters to select a SPI controller for DGDI

5	+0	+1	+2	+3
(1*)	**PORTMT**	**PINMT**	**PORTMR**	**PINMR**

...	+4	+5	+6	+7
(1*)	**PORTCK**	**PINCK**	**PORTSL**	**PINSL**

para. for ADC test

13	+0	+2	+4
(1*)	**TIM**	**BLOCK**	**SAMP**

Details

parameters to select a SPI controller for DGDI
Port and Pin parameters to select the SPI controller where DGDI is connected

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

para. for ADC test
parameters for ADC test

Name	Length	Fmt	Description
TIM	**2**	**xxxx**	**DGDI timeout** Selection of timeout on waiting for DGDI test status ready.
BLOCK	**2**	**xxxx**	**Current measurement block parameters** **Bit(s)** – **Description** **3:0** – **not used, must be set to 0** **5:4** – **Select amplification for current measurement** **00_B:** 5,79 **01_B:** 8,68 **10_B:** 12,53 **11_B:** 19,25 **7:6** – **not used, must be set to 0** **10:8** – **Select current measurement channel**

Name	Length	Fmt	Description
			Bit(s) / Description
			000_B: not used **001_B:** Channel 1 **010_B:** Channel 2 **011_B:** Channel 3 **100_B:** Channel 4
			15:11 — **not used, must be set to 0**
SAMP	**2**	**xxxx**	**ADC measurement parameters**
			Bit(s) / Description
			3:0 — **Number of samples in two's exponent** **0_H:** 1 sample **1_H:** 2 sample **2_H:** 4 sample **3_H:** 8 sample **4_H:** 16 sample **5_H:** 32 sample **6_H:** 64 sample **7_H:** 128 sample **8_H:** 256 sample **9_H:** 256 sample **A_H:** 256 sample **B_H:** 256 sample **C_H:** 256 sample **D_H:** 256 sample **E_H:** 256 sample **F_H:** 256 sample
			15:4 — **not used, must be set to 0**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D0	xx	xx	01	...

Parameters

Error response of command "ADC test"

6	+0	+2	+3
(1*)	**ERROR**	**AVER**	**ACCUM**

Details

Error response of command "ADC test"

Note: The ADC values in response bytes AVER and ACCUM can be transformed to current values with formular:

$$I = \frac{ADC_{value} * 9,77mV - 250mV}{G_{DA\_diff} * R_{Shunt}}$$

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>ERROR</td><td>2</td><td>xxxx</td><td>Error Status DGDI (bitcoded)
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>0</td><td>1: TSW not initialised</td></tr>
<tr><td>1</td><td>1: Microcode is not ready</td></tr>
<tr><td>2</td><td>1: Microcode has checksum error</td></tr>
<tr><td>3</td><td>1: DGDI has loss of external clock</td></tr>
<tr><td>6:4</td><td>reserved</td></tr>
<tr><td>7</td><td>1: DCDC-parameter CRC failed</td></tr>
<tr><td>8</td><td>1: Unimplemented test requested channel 1</td></tr>
<tr><td>9</td><td>1: Unimplemented test requested channel 2</td></tr>
<tr><td>10</td><td>1: Unexpected SW-Interrupt channel 1</td></tr>
<tr><td>11</td><td>1: Unexpected SW-Interrupt channel 2</td></tr>
<tr><td>12</td><td>1: Unexpected diagnosis-Interrupt channel 1</td></tr>
<tr><td>13</td><td>1: Unexpected diagnosis-Interrupt channel 2</td></tr>
<tr><td>14</td><td>1: Unexpected driver-disable-Interrupt channel 1</td></tr>
<tr><td>15</td><td>1: Unexpected driver-disable-Interrupt channel 2</td></tr>
</table></td></tr>
<tr><td>AVER</td><td>1</td><td>xx</td><td>Averaged samples
Average of all sampled ADC values</td></tr>
<tr><td>ACCUM</td><td>2</td><td>xxxx</td><td>Accumulated samples
Sum of all sampled ADC values. For reduced rounding error compared to averaged samples.</td></tr>
</table>

.

86 - Digital IO test

This command is used to switch the pins of the flag bus in DGDI to digital input. In the response the read status of the pins is provided.

Table 8. Initialisation values of DGDI register written by command "Digital IO test"

Register	Address	Init Value
flags_direction	0x01C1	0xFFFF

Register	Address	Init Value
flags_polarity	0x01C2	0x0000
flags_source	0x01C3	0x1FFF
ctrl_reg_seq0	0x0121	Bit 3 set to start Microcode

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D0	86	01	...

Parameters

parameters to select a SPI controller for DGDI

5	+0	+1	+2	+3
(1*)	**PORTMT**	**PINMT**	**PORTMR**	**PINMR**

...	+4	+5	+6	+7
(1*)	**PORTCK**	**PINCK**	**PORTSL**	**PINSL**

para. for dig IO test

13	+0	+2
(1*)	**TIM**	**MASK**

Details

parameters to select a SPI controller for DGDI
Port and Pin parameters to select the SPI controller where DGDI is connected

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

para. for dig IO test

parameters for digital IO test

Name	Length	Fmt	Description
TIM	**2**	**xxxx**	**DGDI timeout** Selection of timeout on waiting for DGDI test status ready.
MASK	**2**	**xxxx**	**Mask pins as digital input**

Bit(s)	Description
0	**Start1**
1	**Start2**
2	**Start3**
3	**Start4**
4	**Start5**
5	**Start6**
6	**Flag0**
7	**Flag1**
8	**Flag2**
9	**OA1**
10	**OA2**
11	**IrqB**
12	**Dbg**
12:0	**Description for all pins** **0_B:** not read **1_B:** read
15:13	**not used, must be set to 0**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D0	xx	xx	01	...

Parameters

Error response of command "Digital IO test"

6 (1*)	+0	+2
	ERROR	**STAT**

Details

Error response of command "Digital IO test"

Name	Length	Fmt	Description
ERROR	**2**	**xxxx**	**Error Status DGDI (bitcoded)**

Bit(s)	Description
0	**1: TSW not initialised**

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td></td><td>Bit(s)</td><td>Description</td></tr>
<tr><td></td><td></td><td></td><td>1</td><td>1: Microcode is not ready</td></tr>
<tr><td></td><td></td><td></td><td>2</td><td>1: Microcode has checksum error</td></tr>
<tr><td></td><td></td><td></td><td>3</td><td>1: DGDI has loss of external clock</td></tr>
<tr><td></td><td></td><td></td><td>6:4</td><td>reserved</td></tr>
<tr><td></td><td></td><td></td><td>7</td><td>1: DCDC-parameter CRC failed</td></tr>
<tr><td></td><td></td><td></td><td>8</td><td>1: Unimplemented test requested channel 1</td></tr>
<tr><td></td><td></td><td></td><td>9</td><td>1: Unimplemented test requested channel 2</td></tr>
<tr><td></td><td></td><td></td><td>10</td><td>1: Unexpected SW-Interrupt channel 1</td></tr>
<tr><td></td><td></td><td></td><td>11</td><td>1: Unexpected SW-Interrupt channel 2</td></tr>
<tr><td></td><td></td><td></td><td>12</td><td>1: Unexpected diagnosis-Interrupt channel 1</td></tr>
<tr><td></td><td></td><td></td><td>13</td><td>1: Unexpected diagnosis-Interrupt channel 2</td></tr>
<tr><td></td><td></td><td></td><td>14</td><td>1: Unexpected driver-disable-Interrupt channel 1</td></tr>
<tr><td></td><td></td><td></td><td>15</td><td>1: Unexpected driver-disable-Interrupt channel 2</td></tr>
<tr><td>STAT</td><td>2</td><td>xxxx</td><td colspan="2">Status of test responsed from microcode</td></tr>
<tr><td></td><td></td><td></td><td>Bit(s)</td><td>Description</td></tr>
<tr><td></td><td></td><td></td><td>0</td><td>Start1</td></tr>
<tr><td></td><td></td><td></td><td>1</td><td>Start2</td></tr>
<tr><td></td><td></td><td></td><td>2</td><td>Start3</td></tr>
<tr><td></td><td></td><td></td><td>3</td><td>Start4</td></tr>
<tr><td></td><td></td><td></td><td>4</td><td>Start5</td></tr>
<tr><td></td><td></td><td></td><td>5</td><td>Start6</td></tr>
<tr><td></td><td></td><td></td><td>6</td><td>Flag0</td></tr>
<tr><td></td><td></td><td></td><td>7</td><td>Flag1</td></tr>
<tr><td></td><td></td><td></td><td>8</td><td>Flag2</td></tr>
<tr><td></td><td></td><td></td><td>9</td><td>OA1</td></tr>
<tr><td></td><td></td><td></td><td>10</td><td>OA2</td></tr>
<tr><td></td><td></td><td></td><td>11</td><td>IrqB</td></tr>
<tr><td></td><td></td><td></td><td>12</td><td>Dbg</td></tr>
<tr><td></td><td></td><td></td><td>12:0</td><td>Description for all pins
0_B: Signal LOW or not read
1_B: Signal HIGH</td></tr>
<tr><td></td><td></td><td></td><td>15:13</td><td>reserved</td></tr>
</table>

.

87 - Digital output OAx test

This command is used to switch the pins OAx, configured as digital output, triggered with one of the start signals.

Table 9. Initialisation values of DGDI register written by command "Digital output OAx test"

Register	Address	Init Value
flags_direction	0x01C1	taken from command parameter
flags_polarity	0x01C2	taken from command parameter
flags_source	0x01C3	taken from command parameter
cur_access_reg1	0x0188	0x001F
cur_access_reg2	0x0189	0x0018
oa_out1	0x01AA	taken from command parameter
oa_out2	0x01AB	taken from command parameter
meas_func1	0x01AC	taken from command parameter
ctrl_reg_seq0	0x0101	Bit 4 set to start Microcode
DRAM0, DRAM29	0x0000, 0x001D	taken from command parameter

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D0	87	01	...

Parameters

parameters to select a SPI controller for DGDI

5	+0	+1	+2	+3
(1*)	**PORTMT**	**PINMT**	**PORTMR**	**PINMR**

...	+4	+5	+6	+7
(1*)	**PORTCK**	**PINCK**	**PORTSL**	**PINSL**

para. for dig O OAx test

13	+0	+2
(1*)	**TIM**	**GENIO**

Details

parameters to select a SPI controller for DGDI
Port and Pin parameters to select the SPI controller where DGDI is connected

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal

Name	Length	Fmt	Description
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

para. for dig O OAx test
parameters for digital output OAx test

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>TIM</td><td>2</td><td>xxxx</td><td>DGDI timeout
Selection of timeout on waiting for DGDI test status ready.</td></tr>
<tr><td>GENIO</td><td>2</td><td>xxxx</td><td>General DGDI input/output configuration parameters
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>2:0</td><td>Select sensitivity to start signals
000_B: not used
001_B: Start1
010_B: Start2
011_B: Start3
100_B: Start4
101_B: Start5
110_B: Start6</td></tr>
<tr><td>3</td><td>not used, must be set to 0</td></tr>
<tr><td>5:4</td><td>Gain setting of OAx amplifier (only low bit is used)
00_B: 1,33
01_B: 2,00
10_B: 1,33
11_B: 2,00</td></tr>
<tr><td>7:6</td><td>not used, must be set to 0</td></tr>
<tr><td>9:8</td><td>Select OAx for current measurement (choose only 1)
00_B: OA1, OA2 high impedance</td></tr>
</table></td></tr>
</table>

Name	Length	Fmt	Description	
			Bit(s)	**Description**
				01$_B$: OA1 digital output, OA2 high impedance **10$_B$:** OA1 high impedance, OA2 digital output **11$_B$:** OA1, OA2 high impedance
			15:10	**not used, must be set to 0**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D0	xx	xx	01	...

Parameters

Error response of command "Digital output OAx test"

6	+0
(1*)	**ERROR**

Details

Error response of command "Digital output OAx test"

Name	Length	Fmt	Description	
ERROR	**2**	**xxxx**	**Error Status DGDI (bitcoded)**	
			Bit(s)	**Description**
			0	**1: TSW not initialised**
			1	**1: Microcode is not ready**
			2	**1: Microcode has checksum error**
			3	**1: DGDI has loss of external clock**
			6:4	**reserved**
			7	**1: DCDC-parameter CRC failed**
			8	**1: Unimplemented test requested channel 1**
			9	**1: Unimplemented test requested channel 2**
			10	**1: Unexpected SW-Interrupt channel 1**
			11	**1: Unexpected SW-Interrupt channel 2**
			12	**1: Unexpected diagnosis-Interrupt channel 1**
			13	**1: Unexpected diagnosis-Interrupt channel 2**
			14	**1: Unexpected driver-disable-Interrupt channel 1**

Name	Length	Fmt	Description	
			Bit(s)	**Description**
			15	**1: Unexpected driver-disable-Interrupt channel 2**

.

88 - Advanced Digital IO test

This command is used to switch the pins of the flag bus in DGDI to digital input or output. In the response the read status of the pins is provided.

Table 10. Initialisation values of DGDI register written by command "Advanced Digital IO test"

Register	Address	Init Value
flags_direction	0x1C1	taken from command parameter
flags_polarity	0x1C2	0x0000
flags_source	0x1C3	0xFFFF
Channel 2 DRAM20	0x054	taken from command parameter
ctrl_reg_seq0	0x121	Bit 4 set to start Microcode

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D0	88	01	...

Parameters

parameters to select a SPI controller to the device

5	+0	+1	+2	+3
(1*)	**POMT**	**PIMT**	**POMR**	**PIMR**

...	+4	+5	+6	+7
(1*)	**POCK**	**PICK**	**POSL**	**PISL**

para. for dig IO test

13	+0	+2	+4	+6
(1*)	**TIM**	**IOSEL**	**OUTVAL**	**MASK**

Details

parameters to select a SPI controller to the device
Port and Pin parameters to select the SPI controller where the device is connected

Name	Length	Fmt	Description
POMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit Slave Receive) signal
PIMT	**1**	**xx**	**Pin number of MTSR signal**

Name	Length	Fmt	Description
			Pin number within POMT of MTSR (Master Transmit Slave Receive) (SI) signal
POMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive Slave Transmit) signal
PIMR	**1**	**xx**	**Pin number of MRST signal** Pin number within POMR of MRST (Master Receive Slave Transmit) (SO) signal
POCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PICK	**1**	**xx**	**Pin number of SCK signal** Pin number within POCK of SCK (Serial Clock) signal
POSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PISL	**1**	**xx**	**Pin number of SLSO signal** Pin number within POSL of SLSO (Slave Select) signal

para. for dig IO test
parameters for digital IO test

Name	Length	Fmt	Description
TIM	**2**	**xxxx**	**DGDI timeout** Selection of timeout on waiting for DGDI test status ready (Format: RVVV).
IOSEL	**2**	**xxxx**	**Input/Output selection**

Bit(s)	Description
15:13	**not used, must be set to 0**
12	**Dbg**
11	**OA2**
10	**OA1**
9	**IrqB**
8	**Start6**
7	**Start5**
6	**Start4**
5	**Start3**
4	**Start2**
3	**Start1**
2	**Flag2**
1	**Flag1**
0	**Flag0**
12:0	**Description for all pins** **0_B:** output **1_B:** input

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>OUTVAL</td><td>2</td><td>xxxx</td><td>Output value
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>15:13</td><td>not used, must be set to 0</td></tr>
<tr><td>12</td><td>Dbg</td></tr>
<tr><td>11</td><td>OA2</td></tr>
<tr><td>10</td><td>OA1</td></tr>
<tr><td>9</td><td>IrqB</td></tr>
<tr><td>8</td><td>Start6</td></tr>
<tr><td>7</td><td>Start5</td></tr>
<tr><td>6</td><td>Start4</td></tr>
<tr><td>5</td><td>Start3</td></tr>
<tr><td>4</td><td>Start2</td></tr>
<tr><td>3</td><td>Start1</td></tr>
<tr><td>2</td><td>Flag2</td></tr>
<tr><td>1</td><td>Flag1</td></tr>
<tr><td>0</td><td>Flag0</td></tr>
<tr><td>12:0</td><td>Description for all pins
0_B: LOW
1_B: HIGH</td></tr>
</table></td></tr>
<tr><td>MASK</td><td>2</td><td>xxxx</td><td>Mask pins as digital input
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>15:13</td><td>not used, must be set to 0</td></tr>
<tr><td>12</td><td>Dbg</td></tr>
<tr><td>11</td><td>OA2</td></tr>
<tr><td>10</td><td>OA1</td></tr>
<tr><td>9</td><td>IrqB</td></tr>
<tr><td>8</td><td>Start6</td></tr>
<tr><td>7</td><td>Start5</td></tr>
<tr><td>6</td><td>Start4</td></tr>
<tr><td>5</td><td>Start3</td></tr>
<tr><td>4</td><td>Start2</td></tr>
<tr><td>3</td><td>Start1</td></tr>
<tr><td>2</td><td>Flag2</td></tr>
<tr><td>1</td><td>Flag1</td></tr>
<tr><td>0</td><td>Flag0</td></tr>
<tr><td>12:0</td><td>Description for all pins
0_B: not read
1_B: read</td></tr>
</table></td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D0	xx	xx	01	...

Parameters

Error response of command "Digital IO test"

6	+0	+2
(1*)	**ERROR**	**STAT**

Details

Error response of command "Digital IO test"

Name	Length	Fmt	Description	
ERROR	**2**	**xxxx**	**Error Status DGDI (bitcoded)**	
			Bit(s)	**Description**
			15	**1: Unexpected driver-disable-Interrupt channel 2**
			14	**1: Unexpected driver-disable-Interrupt channel 1**
			13	**1: Unexpected diagnosis-Interrupt channel 2**
			12	**1: Unexpected diagnosis-Interrupt channel 1**
			11	**1: Unexpected SW-Interrupt channel 2**
			10	**1: Unexpected SW-Interrupt channel 1**
			9	**1: Unimplemented test requested channel 2**
			8	**1: Unimplemented test requested channel 1**
			7	**1: DCDC-parameter CRC failed**
			6:4	**reserved**
			3	**1: DGDI has loss of external clock**
			2	**1: Microcode has checksum error**
			1	**1: Microcode is not ready**
			0	**1: TSW not initialised**
STAT	**2**	**xxxx**	**Status of test responsed from microcode**	
			Bit(s)	**Description**
			15:13	**reserved**
			12	**Dbg**
			11	**OA2**

Name	Length	Fmt	Description	
			Bit(s)	**Description**
			10	**OA1**
			9	**IrqB**
			8	**Start6**
			7	**Start5**
			6	**Start4**
			5	**Start3**
			4	**Start2**
			3	**Start1**
			2	**Flag2**
			1	**Flag1**
			0	**Flag0**
			12:0	**Description for all pins** **0_B:** Signal LOW or not read **1_B:** Signal HIGH

.

89 - Edge counting on digital input

This command is used to count the risinge edges of a digital input pin. In the response the number of rising edges are provided.

Table 11. Initialisation values of DGDI register written by command "Edge counting on digital input"

Register	Address	Init Value
counter_34_prescaler	0x111	taken from command parameter
flags_direction	0x1C1	taken from command parameter
flags_polarity	0x1C2	0x0000
flags_source	0x1C3	0x1FFF
DRAM0	0x000	taken from command parameter
DRAM1	0x001	taken from command parameter
DRAM50	0x032	0x0000
DRAM51	0x033	0x0000
ctrl_reg_seq0	0x101	Bit 8 set to start Microcode

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D0	89	01	...

Parameters

parameters to select a SPI controller to the device

5	+0	+1	+2	+3
(1*)	**POMT**	**PIMT**	**POMR**	**PIMR**

...	+4	+5	+6	+7
(1*)	**POCK**	**PICK**	**POSL**	**PISL**

para. for edge counting

13	+0	+2	+3	+4
(1*)	**TIM**	**IPN**	**MWRES**	**MWTIM**

Details

parameters to select a SPI controller to the device
Port and Pin parameters to select the SPI controller where the device is connected

Name	Length	Fmt	Description
POMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit Slave Receive) signal
PIMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within POMT of MTSR (Master Transmit Slave Receive) (SI) signal
POMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive Slave Transmit) signal
PIMR	**1**	**xx**	**Pin number of MRST signal** Pin number within POMR of MRST (Master Receive Slave Transmit) (SO) signal
POCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PICK	**1**	**xx**	**Pin number of SCK signal** Pin number within POCK of SCK (Serial Clock) signal
POSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PISL	**1**	**xx**	**Pin number of SLSO signal** Pin number within POSL of SLSO (Slave Select) signal

para. for edge counting
parameters for edge counting

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>TIM</td><td>2</td><td>xxxx</td><td>DGDI timeout
Selection of timeout on waiting for DGDI test status ready (Format: RVVV).</td></tr>
<tr><td>IPN</td><td>1</td><td>xx</td><td>Input pin number
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:4</td><td>not used, must be set to 0</td></tr>
<tr><td>3:0</td><td>Input pin number

0000_B: Flag0</td></tr>
</table></td></tr>
</table>

Name	Length	Fmt	Description
			Bit(s) / **Description** **0001$_B$:** Flag1 **0010$_B$:** Flag2 **0011$_B$:** Start1 **0100$_B$:** Start2 **0101$_B$:** Start3 **0110$_B$:** Start4 **0111$_B$:** Start5 **1000$_B$:** Start6 **1001$_B$:** IrqB **1010$_B$:** OA1 **1011$_B$:** OA2 **1100$_B$:** Dbg **1101$_B$:** not used **1110$_B$:** not used **1111$_B$:** not used
MWRES	**1**	**xx**	**Measurement window resolution** **Bit(s)** / **Description** **7:4** — **not used, must be set to 0** **3:0** — **Resolution of measurement time** **0000$_B$:** 1/6µs **0001$_B$:** 1/3µs **0010$_B$:** 1/2µs **0011$_B$:** 2/3µs **0100$_B$:** 5/6µs **0101$_B$:** 1µs **0110$_B$:** 7/6µs **0111$_B$:** 4/3µs **1000$_B$:** 3/2µs **1001$_B$:** 5/3µs **1010$_B$:** 11/6µs **1011$_B$:** 2µs **1100$_B$:** 13/6µs **1101$_B$:** 7/3µs **1110$_B$:** 5/2µs **1111$_B$:** 8/3µs
MWTIM	**2**	**xxxx**	**Measurement window time** Measurement time. Resolution is 1 MWRES/LSB

Name	Length	Fmt	Description
			Note for MWRES, MWTIM: Expected NRE = MWTIM * (MWRES + 1) / 6MHz * (Expected frequency on pin)

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D0	xx	xx	01	...

Parameters

Error response of command "Edge counting on digital input"

6	+0	+2
(1*)	**ERROR**	**NRE**

Details

Error response of command "Edge counting on digital input"

Name	Length	Fmt	Description
ERROR	**2**	**xxxx**	**Error Status DGDI (bitcoded)**
			Bit(s) — **Description**
			15 — **1: Unexpected driver-disable-Interrupt channel 2**
			14 — **1: Unexpected driver-disable-Interrupt channel 1**
			13 — **1: Unexpected diagnosis-Interrupt channel 2**
			12 — **1: Unexpected diagnosis-Interrupt channel 1**
			11 — **1: Unexpected SW-Interrupt channel 2**
			10 — **1: Unexpected SW-Interrupt channel 1**
			9 — **1: Unimplemented test requested channel 2**
			8 — **1: Unimplemented test requested channel 1**
			7 — **1: DCDC-parameter CRC failed**
			6:4 — **reserved**
			3 — **1: DGDI has loss of external clock**
			2 — **1: Microcode has checksum error**
			1 — **1: Microcode is not ready**
			0 — **1: TSW not initialised**
NRE	**4**	**xxxxxxxx**	**Number of risinge edges**

.

A4 - DCDC copy feedback signal

This command is used to start and stop the copying of the DCDC feedback signal Cur4l_fbk to a Start signal. This command is used in case of two ASIC's and coupled DCDC's. The used Start signal is switched to output and must therefore be configured as input signal on the CPU before using this function. Prior the usage an offset compensation with command 0x80 should take place. Also prior the DCDC should be configured with command 0x83 (DCDC converter setup) with the same values as the other DCDC but the value for the dac4l register (1A1h) set to the mean value of dac4l (1A1h) and dac4h (1A2h) of the other DCDC. The detected level changes are counted in DRAM61 (high to low) and DRAM62 (low to high).

Table 12. Initialisation values of DGDI register written by command "DCDC copy feedback signal"

Register	Address	Init Value
ctrl_reg_seq0	0x0121	Bit 2 set to start Microcode
flags_direction	0x01C1	0x1FFF with 1 bit set to zero (depends on STX)
flags_source	0x01C3	0x1FFF
Channel 2 DRAM6	0x0046	taken from command parameter
Channel 2 DRAM61	0x007D	0x0000
Channel 2 DRAM62	0x007E	0x0000

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D0	A4	01	...

Parameters

parameters to select a SPI controller for DGDI

5	+0	+1	+2	+3
(1*)	**PORTMT**	**PINMT**	**PORTMR**	**PINMR**

...	+4	+5	+6	+7
(1*)	**PORTCK**	**PINCK**	**PORTSL**	**PINSL**

para. for DCDC copy feedback signal

13	+0	+2
(1*)	**TIM**	**STX**

Details

parameters to select a SPI controller for DGDI
Port and Pin parameters to select the SPI controller where DGDI is connected

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal**

Name	Length	Fmt	Description
			Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

para. for DCDC copy feedback signal
parameters for DCDC copy feedback signal

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>TIM</td><td>2</td><td>xxxx</td><td>DGDI timeout
Selection of timeout on waiting for DGDI test status ready.</td></tr>
<tr><td>STX</td><td>1</td><td>xx</td><td>Used start signal
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>3:0</td><td>Select used start signal

0000$_B$: Stop copying of feedback signal
0001$_B$: Start copying of feedback signal to start1
0010$_B$: Start copying of feedback signal to start2
0011$_B$: Start copying of feedback signal to start3
0100$_B$: Start copying of feedback signal to start4
0101$_B$: Start copying of feedback signal to start5
0110$_B$: Start copying of feedback signal to start6
1111$_B$: no output signal, but count feedback changes</td></tr>
<tr><td>7:4</td><td>not used, must be set to 0</td></tr>
</table></td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D0	xx	xx	01	...

Parameters

Error response of command "DCDC copy feedback signal"

6	+0
(1*)	**ERROR**

Details

Error response of command "DCDC copy feedback signal"

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>ERROR</td><td>2</td><td>xxxx</td><td>Error Status DGDI (bitcoded)
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>0</td><td>1: TSW not initialised</td></tr>
<tr><td>1</td><td>1: Microcode is not ready</td></tr>
<tr><td>2</td><td>1: Microcode has checksum error</td></tr>
<tr><td>3</td><td>1: DGDI has loss of external clock</td></tr>
<tr><td>6:4</td><td>reserved</td></tr>
<tr><td>7</td><td>1: DCDC-parameter CRC failed</td></tr>
<tr><td>8</td><td>1: Unimplemented test requested channel 1</td></tr>
<tr><td>9</td><td>1: Unimplemented test requested channel 2</td></tr>
<tr><td>10</td><td>1: Unexpected SW-Interrupt channel 1</td></tr>
<tr><td>11</td><td>1: Unexpected SW-Interrupt channel 2</td></tr>
<tr><td>12</td><td>1: Unexpected diagnosis-Interrupt channel 1</td></tr>
<tr><td>13</td><td>1: Unexpected diagnosis-Interrupt channel 2</td></tr>
<tr><td>14</td><td>1: Unexpected driver-disable-Interrupt channel 1</td></tr>
<tr><td>15</td><td>1: Unexpected driver-disable-Interrupt channel 2</td></tr>
</table></td></tr>
</table>

.

AF - Power stage operating function

This command is used to trigger an injection operating function or, without boost phase, a QCV (or similar) operating function.

HINT: Set I_BOOST to zero to skip boost phase!

Figure 2. injection current over time

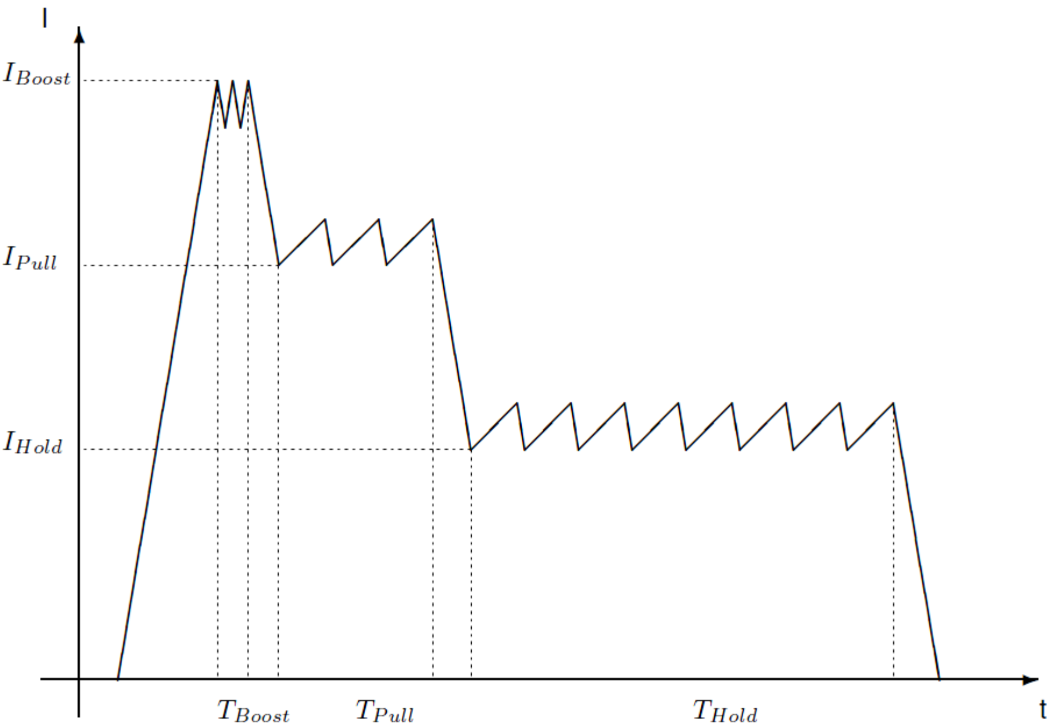


Table 13. Initialisation values of DGDI register written by command "PS operating function"

Register	Address	Init Value
start_config_reg	0x0104	0x0000
cur_access_reg1	0x0188	0x001F
cur_access_reg2	0x0189	0x0000
bias_regfile	0x01A4	taken from command parameter
ctrl_reg_seq0	0x0101	Bit 9 set to start Microcode
DRAM00-DRAM10	0x0000-0x000A	taken from command parameter

Note: All currents can be transformed to ADC values in command with following formular:

$$ADC_{value} = \frac{I * G_{DA\_diff} * R_{Shunt} + 250mV}{9,77mV}$$

To set G_{DA_diff} use MBLOCK [9:8] (amplification for current measurement).

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D0	AF	01	...

Parameters

parameters to select a SPI controller for DGDI

5 (1*)	+0	+1	+2	+3
	PORTMT	**PINMT**	**PORTMR**	**PINMR**

... (1*)	+4	+5	+6	+7
	PORTCK	**PINCK**	**PORTSL**	**PINSL**

para. for PS operating function

13 (1*)	+0	+2	+4	+6	+8	+10
	TIM	**SCFG**	**MBLOCK**	**I_BOOST**	**I_PULL**	**I_HOLD**

... (1*)	+12	+14	+16	+18	+20	+22
	TU_MIN	**TU_MAX**	**T_BOOST**	**T_PULL**	**T_DECAY**	**T_HOLD**

... (1*)	+24	+26
	T_IDLE	**CYCLES**

Details

parameters to select a SPI controller for DGDI
Port and Pin parameters to select the SPI controller where DGDI is connected

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

para. for PS operating function

parameters for PS operating function

Name	Length	Fmt	Description
TIM	**2**	**xxxx**	**DGDI timeout** Selection of timeout on waiting for DGDI test status ready. Format: RVVV.
SCFG	**2**	**xxxx**	**Switch configuration** **Bit(s)** / **Description** **3:0** — **HSBat, HSBoost and LS** **0_H:** hs1, hs2, ls1 **1_H:** hs1, hs2, ls2 **2_H:** hs1, hs2, ls3 **3_H:** hs1, hs2, ls4 **4_H:** hs1, hs2, ls5 **5_H:** hs3, hs4, ls4 **6_H:** hs3, hs4, ls5 **7_H:** hs3, hs4, ls6 **8_H:** hs5, ls5, ls2 **9_H:** hs5, ---, ls3 **A_H:** not used **B_H:** not used **C_H:** not used **D_H:** not used **E_H:** not used **F_H:** not used **13:4** — **not used, must be set to 0** **14** — **Deactivate WDA communication during operation cycles to suppress jitter between cycles** **0_B:** communication unchanged **1_B:** communication deactivated **15** — **Bias configuration for used HSBoost/LS combination(switch configuration 0-8) or HSBat/LS combination (switch configuration 9)** **0_B:** Bias off **1_B:** Bias on
MBLOCK	**2**	**xxxx**	**Current measurement block parameters** **Bit(s)** / **Description** **2:0** — **Select current measurement channel** **000_B:** not used **001_B:** Channel 1

Name	Length	Fmt	Description
			Bit(s) | **Description** | **010_B:** Channel 2 | **011_B:** Channel 3 | **100_B:** Channel 4 **7:3** | **not used, must be set to 0** **9:8** | **Select amplification for current measurement (G_DA_diff)** | **00_B:** 5,79 | **01_B:** 8,68 | **10_B:** 12,53 | **11_B:** 19,25 **15:10** | **not used, must be set to 0**
I_BOOST	**2**	**xxxx**	**Boost current** Current of boost phase (ADC value). Set to zero to skip boost phase. **Bit(s)** | **Description** **7:0** | **low regulation point** **15:8** | **high regulation point**
I_PULL	**2**	**xxxx**	**Pull current** Current of pull phase (ADC value) **Bit(s)** | **Description** **7:0** | **low regulation point** **15:8** | **high regulation point**
I_HOLD	**2**	**xxxx**	**Hold current** Current of hold phase (ADC value) **Bit(s)** | **Description** **7:0** | **low regulation point** **15:8** | **high regulation point**
TU_MIN	**2**	**xxxx**	**Minimal uptime** Minimal time to reach I_BOOST/I_PULL. If I_BOOST/I_PULL is reached before this time, a shortcut is assumed and error is returned. Resolution is 1/6µs/LSB.
TU_MAX	**2**	**xxxx**	**Maximal uptime** Maximal time to reach I_BOOST/I_PULL. If I_BOOST/I_PULL is not reached before this time, a open circuit is assumed and error is returned. Resolution is 1/6µs/LSB.
T_BOOST	**2**	**xxxx**	**Boost time** Time, I_BOOST is maintained. Resolution is 1/6µs/LSB.
T_PULL	**2**	**xxxx**	**Pull time**

Name	Length	Fmt	Description
			Time, I_PULL is maintained. Resolution is 1/6µs/ LSB.
T_DECAY	**2**	**xxxx**	**Fast decay time** Time, LS is switched off after pull phase (fast decay). Typically 20µs (0x0078). Resolution is 1/6µs/LSB.
T_HOLD	**2**	**xxxx**	**Hold time** Time, I_HOLD is maintained. Resolution is 1/6µs/ LSB.
T_IDLE	**2**	**xxxx**	**Idle time** Time before next cycle. Format: RVVV.
CYCLES	**1**	**xx**	**Cycles** Number of cycles

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D0	xx	xx	01	...

Parameters

Error response of command "PS operating function"

6	+0	+2
(1*)	**ERROR**	**STAT**

Details

Error response of command "PS operating function"

Name	Length	Fmt	Description
ERROR	**2**	**xxxx**	**Error Status DGDI (bitcoded)**

Bit(s)	Description
0	**1: TSW not initialised**
1	**1: Microcode is not ready**
2	**1: Microcode has checksum error**
3	**1: DGDI has loss of external clock**
6:4	**reserved**
7	**1: DCDC-parameter CRC failed**
8	**1: Unimplemented test requested channel 1**
9	**1: Unimplemented test requested channel 2**
10	**1: Unexpected SW-Interrupt channel 1**

Name	Length	Fmt	Description
			Bit(s) / **Description** **11**: **1: Unexpected SW-Interrupt channel 2** **12**: **1: Unexpected diagnosis-Interrupt channel 1** **13**: **1: Unexpected diagnosis-Interrupt channel 2** **14**: **1: Unexpected driver-disable-Interrupt channel 1** **15**: **1: Unexpected driver-disable-Interrupt channel 2**
STAT	**2**	**xxxx**	**Status of test responsed from microcode (bitcoded)** **Bit(s)** / **Description** **0**: **1: Boost/Pull current I_BOOST/ I_PULL is reached before TU_MIN** **1**: **1: Boost/Pull current I_BOOST/ I_PULL is not reached within TU_MAX** **2**: **1: Switch configuration not available** **15:3**: **reserved**

.

C0 - VCCP regulator off/on cycling

This command is used to trigger a VCCP regulator off/on cycling. This test is part of the extended device screening.

NOTE: Ensure that hardware use no external VCCP regulator!

NOTE: Ensure that all bootstrap capacitors are charged or disabled (see bootstrap_charged register 0x1A5, bit 4-0)!

Table 14. Initialisation values of DGDI register written by command "VCCP regulator off/on cycling"

Register	Address	Init Value
driver_config	0x01C5	Bit 14 toggled due to command parameters

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D0	C0	01	...

Parameters

parameters to select a SPI controller for DGDI

5	+0	+1	+2	+3
(1*)	**PORTMT**	**PINMT**	**PORTMR**	**PINMR**

...	+4	+5	+6	+7
(1*)	**PORTCK**	**PINCK**	**PORTSL**	**PINSL**

parameters for VCCP regulator off/on cycling

13	+0	+2	+4	+6
(1*)	**TIM**	**T_OFF**	**T_ON**	**CYCLES**

Details

parameters to select a SPI controller for DGDI
Port and Pin parameters to select the SPI controller where DGDI is connected

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

parameters for VCCP regulator off/on cycling
configuration parameters for VCCP regulator off/on cycling

Name	Length	Fmt	Description
TIM	**2**	**xxxx**	**DGDI timeout** Selection of timeout on waiting for DGDI test status ready. Format: RVVV. (Currently not used.)
T_OFF	**2**	**xxxx**	**Off time** Time the regulator is switched off. Format: RVVV.
T_ON	**2**	**xxxx**	**On time** Time the regulator is switched on. Format: RVVV.

Name	Length	Fmt	Description
CYCLES	**1**	**xx**	**Cycles** Number of cycles

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D0	xx	xx	01	...

Parameters

response of command "VCCP regulator off/on cycling"

...	+0
(1*)	**DRVSTAT**

Details

response of command "VCCP regulator off/on cycling"

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>DRVSTAT</td><td>1</td><td>xx</td><td>Uv vccp of driver_status register
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>0</td><td>bit 0 of register 0x1D2</td></tr>
<tr><td>7:1</td><td>reserved, set to 0</td></tr>
</table></td></tr>
</table>

.

C1 - DCDC ramp up with variable slew rate settings

This command is used to start the DC/DC converter with different slew rate settings. It is nearly the same as command 0x83 (DCDC converter switching) but uses pairs of booster voltage (BOOSTV) and used slew rate (SLEW). This test is part of the extended device screening.

NOTE: This command ignores the parameters BOOSTV and SLEW from command 0x82.

Table 15. Initialisation values of DGDI register written by command "DCDC ramp up with variable slew rate settings"

Register	Address	Init Value
slewrate_ls	0x018F	taken from command parameter
boost_dac	0x019B	taken from command parameter
DRAM4	0x0044	taken from command parameter
ctrl_reg_seq0	0x0121	Bit 1 set to start Microcode

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D0	C1	01	...

Parameters

parameters to select a SPI controller for DGDI

5	+0	+1	+2	+3
(1*)	**PORTMT**	**PINMT**	**PORTMR**	**PINMR**

...	+4	+5	+6	+7
(1*)	**PORTCK**	**PINCK**	**PORTSL**	**PINSL**

parameters for DCDC ramp up with variable slew rate settings

...	+0
(1*)	**TIM**

additional parameter of command

15	+0	+1
(1..n*)	**BOOSTV**	**SLEW**

Details

parameters to select a SPI controller for DGDI
Port and Pin parameters to select the SPI controller where DGDI is connected

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal**

Name	Length	Fmt	Description
			Port number of SLSO (Slave Select) signal
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

parameters for DCDC ramp up with variable slew rate settings
configuration parameters for DCDC ramp up with variable slew rate settings

Name	Length	Fmt	Description
TIM	**2**	**xxxx**	**DGDI timeout** Selection of timeout on waiting for DGDI test status ready. Format: RVVV.

additional parameter of command

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>BOOSTV</td><td>1</td><td>xx</td><td>Booster voltage parameters
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:0</td><td>DAC value to select Booster Voltage (boostdac register 19Bh)

Note: The DAC value for the booster voltage need to be calculated with formular:
$DAC\_value = \frac{V_{boost}}{313mV}$

Note: Only values between 0x08 (2,5V) and 0xD0 (65V) are allowed</td></tr>
</table></td></tr>
<tr><td>SLEW</td><td>1</td><td>xx</td><td>Slew rate setting for LS7
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>0</td><td>Select value for R DSON of PMOS transistor to configure slew rate for switching on LS7. Value in Ohm.

0_B: 5.0 (1500V/µs)
1_B: 14.6 (300V/µs)</td></tr>
<tr><td>1</td><td>reserved, must be set to 0</td></tr>
<tr><td>3:2</td><td>not used, must be set to 0</td></tr>
<tr><td>4</td><td>Select value for R DSON of NMOS transistor to configure slew rate for switching off LS7. Value in Ohm.

0_B: 1.1 (1500V/µs)
1_B: 5.9 (300V/µs)</td></tr>
<tr><td>5</td><td>reserved, must be set to 0</td></tr>
<tr><td>7:6</td><td>not used, must be set to 0</td></tr>
</table></td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D0	xx	xx	01	...

Parameters

response of command "DCDC ramp up with variable slew rate settings"

6 (1*)	+0	+2
	ERROR	**STAT**

additional response of command "DCDC ramp up with variable slew rate settings"

... (1..n*)	+0
	TIME

Details

response of command "DCDC ramp up with variable slew rate settings"

Name	Length	Fmt	Description
ERROR	**2**	**xxxx**	**Error Status DGDI (bitcoded)**

Bit(s)	Description
0	**1: TSW not initialised**
1	**1: Microcode is not ready**
2	**1: Microcode has checksum error**
3	**1: DGDI has loss of external clock**
6:4	**reserved**
7	**1: DCDC-parameter CRC failed**
8	**1: Unimplemented test requested channel 1**
9	**1: Unimplemented test requested channel 2**
10	**1: Unexpected SW-Interrupt channel 1**
11	**1: Unexpected SW-Interrupt channel 2**
12	**1: Unexpected diagnosis-Interrupt channel 1**
13	**1: Unexpected diagnosis-Interrupt channel 2**
14	**1: Unexpected driver-disable-Interrupt channel 1**
15	**1: Unexpected driver-disable-Interrupt channel 2**

Name	Length	Fmt	Description
STAT	**2**	**xxxx**	**Status of test responsed from microcode (bitcoded)**

Name	Length	Fmt	Description	
			Bit(s)	**Description**
			0	**1: Booster voltage is reached before Tmin**
			1	**1: Booster voltage is not reached within Tmax**
			15:2	**reserved**

additional response of command "DCDC ramp up with variable slew rate settings"

Name	Length	Fmt	Description
TIME	**2**	**xxxx**	**Ramp up time:** DCDC ramp-up time. Resolution is 2.5µs/LSB.

.

C2 - HSx/LSx cycling

This command is used to trigger a HSx or LSx cycling. This test is part of the extended device screening.

Table 16. Initialisation values of DGDI register written by command "HSx/LSx cyclings"

Register	Address	Init Value
cur_access_reg1	0x0188	0x000F
slewrate_hs	0x018E	taken from command parameter
slewrate_ls	0x018F	taken from command parameter
bias_config	0x01A4	taken from command parameter
DRAM0-DRAM7	0x0000-0x0007	taken from command parameter
ctrl_reg_seq0	0x0101	Bit 10, 11 and 12 set to start Microcode

Note: All currents can be transformed to ADC values in command with following formular:

$$ADC_{value} = \frac{I * G_{DA\_diff} * R_{Shunt} + 250mV}{9,77mV}$$

To set G_{DA_diff} use MBLOCK [9:8] (amplification for current measurement).

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D0	C2	01	...

Parameters

parameters to select a SPI controller for DGDI

5	+0	+1	+2	+3
(1*)	**PORTMT**	**PINMT**	**PORTMR**	**PINMR**

...	+4	+5	+6	+7
(1*)	**PORTCK**	**PINCK**	**PORTSL**	**PINSL**

parameters for HSx/LSx cycling

13	+0	+2	+3
(1*)	**TIM**	**MODE**	**REGINI**

...	+4	+6	+8
(1*)	**SCFG**	**MBLOCK**	**I_CYCLE**

...	+10	+12	+14
(1*)	**T_WAIT**	**T_DELAY**	**T_CYCLE**

additional parameters for HSx/LSx cycling

...	+0
(1..n*)	**SLEW**

Details

parameters to select a SPI controller for DGDI
Port and Pin parameters to select the SPI controller where DGDI is connected

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

parameters for HSx/LSx cycling
configuration parameters for HSx/LSx cyclings

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>TIM</td><td>2</td><td>xxxx</td><td>DGDI timeout
Selection of timeout on waiting for DGDI test status ready. Format: RVVV.</td></tr>
<tr><td>MODE</td><td>1</td><td>xx</td><td>Mode:
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>1:0</td><td>Cycling mode
00_B: HS VBatt cycling
01_B: HS VBoost cycling
10_B: LS cycling, HS on VBatt
11_B: LS cycling, HS on VBoost</td></tr>
<tr><td>7:2</td><td>not used, must be set to 0</td></tr>
</table></td></tr>
<tr><td>REGINI</td><td>1</td><td>xx</td><td>Register setting before test:
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>1:0</td><td>Slewrate for the not cycling switch
00_B: slew rate according to datasheet
01_B: slew rate according to datasheet
10_B: slew rate according to datasheet
11_B: slew rate according to datasheet</td></tr>
<tr><td>3:2</td><td>not used, must be set to 0</td></tr>
<tr><td>4</td><td>Bias HS setting (bias_config register bit 0-4)
0_B: Bias off
1_B: Bias on</td></tr>
<tr><td>5</td><td>Bias LS setting (bias_config register bit 5-10)
0_B: Bias off
1_B: Bias on</td></tr>
<tr><td>6</td><td>Bias HS strong setting (bias_config register bit 11-12)
0_B: Bias off
1_B: Bias on</td></tr>
<tr><td>7</td><td>not used, must be set to 0</td></tr>
</table></td></tr>
<tr><td>SCFG</td><td>2</td><td>xxxx</td><td>Switch configuration</td></tr>
</table>

Name	Length	Fmt	Description
			Bit(s) / **Description**
			3:0 **HSBat, HSBoost and LS** **0_H:** hs1, hs2, ls1 **1_H:** hs1, hs2, ls2 **2_H:** hs1, hs2, ls3 **3_H:** hs1, hs2, ls4 **4_H:** hs1, hs2, ls5 **5_H:** hs3, hs4, ls4 **6_H:** hs3, hs4, ls5 **7_H:** hs3, hs4, ls6 **8_H:** hs5, ls5, ls2 **9_H:** hs5, ---, ls3 **A_H:** not used **B_H:** not used **C_H:** not used **D_H:** not used **E_H:** not used **F_H:** not used
			15:4 **not used, must be set to 0**
MBLOCK	**2**	**xxxx**	**Current measurement block parameters**
			Bit(s) / **Description**
			2:0 **Select current measurement channel** **000_B:** not used **001_B:** Channel 1 **010_B:** Channel 2 **011_B:** Channel 3 **100_B:** Channel 4
			7:3 **not used, must be set to 0**
			9:8 **Select amplification for current measurement (G_DA_diff)** **00_B:** 5.79 **01_B:** 8.68 **10_B:** 12.53 **11_B:** 19.25
			15:10 **not used, must be set to 0**
I_CYCLE	**2**	**xxxx**	**Cycle current and/or minimum on time:** Current of cycle (ADC value)

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td><table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>7:0</td><td>low regulation point (only for HS cycling)</td></tr><tr><td>7:0</td><td>minimum LS on time in 1/6µs (only for LS cycling)</td></tr><tr><td>15:8</td><td>high regulation point</td></tr></table></td></tr>
<tr><td>T_WAIT</td><td>2</td><td>xxxx</td><td>Wait time
Time for LS off (only for LS cycling). Resolution is 1/6µs/LSB.</td></tr>
<tr><td>T_DELAY</td><td>2</td><td>xxxx</td><td>Delay time
Time between LS on and HS on (HS cycling) or between HS on and LS on (LS cycling). Resolution is 1/6µs/LSB.</td></tr>
<tr><td>T_CYCLE</td><td>2</td><td>xxxx</td><td>Cycle time
Time for a cycle with one slew rate setting. Resolution is 1/6µs/LSB.</td></tr>
</table>

additional parameters for HSx/LSx cycling

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>SLEW</td><td>1</td><td>xx</td><td>Slew rate setting:<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>1:0</td><td>Setting of the slew rate for this cycle
00_B: slew rate according to datasheet
01_B: slew rate according to datasheet
10_B: slew rate according to datasheet
11_B: slew rate according to datasheet</td></tr><tr><td>7:2</td><td>not used, must be set to 0</td></tr></table></td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D0	xx	xx	01	...

Parameters

Error response of command "HSx/LSx cycling"

6	+0	+2	+4	+6	+8
(1*)	**ERROR**	**STAT**	**ERR1**	**ERR2**	**CNUM**

Details

Error response of command "HSx/LSx cycling"

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>ERROR</td><td>2</td><td>xxxx</td><td>Error Status DGDI (bitcoded)
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>0</td><td>1: TSW not initialised</td></tr>
<tr><td>1</td><td>1: Microcode is not ready</td></tr>
<tr><td>2</td><td>1: Microcode has checksum error</td></tr>
<tr><td>3</td><td>1: DGDI has loss of external clock</td></tr>
<tr><td>6:4</td><td>reserved</td></tr>
<tr><td>7</td><td>1: DCDC-parameter CRC failed</td></tr>
<tr><td>8</td><td>1: Unimplemented test requested channel 1</td></tr>
<tr><td>9</td><td>1: Unimplemented test requested channel 2</td></tr>
<tr><td>10</td><td>1: Unexpected SW-Interrupt channel 1</td></tr>
<tr><td>11</td><td>1: Unexpected SW-Interrupt channel 2</td></tr>
<tr><td>12</td><td>1: Unexpected diagnosis-Interrupt channel 1</td></tr>
<tr><td>13</td><td>1: Unexpected diagnosis-Interrupt channel 2</td></tr>
<tr><td>14</td><td>1: Unexpected driver-disable-Interrupt channel 1</td></tr>
<tr><td>15</td><td>1: Unexpected driver-disable-Interrupt channel 2</td></tr>
</table></td></tr>
<tr><td>STAT</td><td>2</td><td>xxxx</td><td>Status of test responsed from microcode (bitcoded)
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>0</td><td>1: Open circuit detected (no load connected)</td></tr>
<tr><td>15:1</td><td>reserved</td></tr>
</table></td></tr>
<tr><td>ERR1</td><td>2</td><td>xxxx</td><td>Content of register err_seq1ch1 part 1 (0x164)</td></tr>
<tr><td>ERR2</td><td>2</td><td>xxxx</td><td>Content of register err_seq1ch1 part 2 (0x165)</td></tr>
<tr><td>CNUM</td><td>1</td><td>xx</td><td>Number of started cycles with different slew rates</td></tr>
</table>

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	TSW execution was successful
0x01	Command interface version not supported.
0x02	Invalid argument(s)

Code	Description
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**No SPI controller on given ports found.**
0x02	**HAL parameter error**
0x03	**SPI communication failed**
0x04	**DGDI communication failed**
0x05	**Setup of microcode failed**
0x06	**Microcode invalid (CRC failure)**
0x07	**Microcode not found**
0x08	**confict of DCDC configuration and LS7, Channel 4 test**

TSW_D1 - eMGS3325 (DGDI-S Gen.2) - Programmable injection device

Test step description

Date: 17.11.2016

Created: 18.01.2017 09:12.

Purpose: Test of eMGS3325 (DGDI-S Gen.2) programmable injection device, for control of power MOSFETs driving solenoid valves and electro-injectors.

Important: before use, initialize ports, SPI and setup a 1MHz clock for the device.

fixed SPI frame configuration: 16 bits per frame, burst mode active.

Test step commands

.

00 - Initialise device

This command is used to bring a basic setup of the registers into the device.

Table 1. Initialisation values of eMGS3325 register"

Register	Address	Init Value
selection_reg	0x3FF	0x0100
ck_per	0x1A0	0x0003
spi_config	0x1A9	0x0003
backup_clock_status_reg	0x1A8	0x01F8
sw_interrupt_routine_addr_ch1	0x10E	0xC180
sw_interrupt_routine_addr_ch2	0x12E	0x3006
sw_interrupt_routine_addr_ch3	0x14E	0x3006
ck_ofscmp_per	0x1A4	0x0047
counter34_prescaler_ch3	0x151	0xBBBB
start_config_part2_ch1	0x104	0x0002
out_acc_seq0_ch1	0x160	0xFF7F
out_acc_seq1_ch1	0x161	0xFF7F
cur_access_reg_ch1	0x166	0x00FF
reset_behaviour	0x1AE	0xFF3F
hs1_diag_config_1	0x1D8	0x2150
hs1_diag_config_2	0x1D9	0x0008
hs2_diag_config_1	0x1DB	0x2150
hs2_diag_config_2	0x1DC	0x0008
hs3_diag_config_1	0x1DE	0x2150
hs3_diag_config_2	0x1DF	0x0008
hs4_diag_config_1	0x1E1	0x2150

Register	Address	Init Value
hs4_diag_config_2	0x1E2	0x0008
hs5_diag_config_1	0x1E4	0x2150
hs5_diag_config_2	0x1E5	0x0008
hs6_diag_config_1	0x1E7	0x2150
hs6_diag_config_2	0x1E8	0x0008
hs7_diag_config_1	0x1EA	0x2150
hs7_diag_config_2	0x1EB	0x0008
vds_threshold_hs_1	0x16B	0x3333
vds_threshold_hs_2	0x16C	0x0333
vsrc_threshold_hs_1	0x16D	0x2222
vsrc_threshold_hs_2	0x16E	0x0222
fbk_sens_seq0_1_ch1	0x154	0x0000
fbk_sens_seq0_2_ch1	0x155	0x0000
fbk_sens_seq1_1_ch1	0x156	0x007F
fbk_sens_seq1_2_ch1	0x157	0x00FF
fbk_sens_seq0_1_ch2	0x158	0x0000
fbk_sens_seq0_2_ch2	0x159	0x0000
fbk_sens_seq1_1_ch2	0x15A	0x0000
fbk_sens_seq1_2_ch2	0x15B	0x0000
fbk_sens_seq0_1_ch3	0x15C	0x0000
fbk_sens_seq0_2_ch3	0x15D	0x0000
fbk_sens_seq1_1_ch3	0x15E	0x0000
fbk_sens_seq1_2_ch3	0x15F	0x0000
vds_threshold_ls_1	0x16F	0x2222
vds_threshold_ls_2	0x170	0x2222
ls1_diag_config_1	0x1C0	0x2150
ls1_diag_config_2	0x1C1	0x0008
ls2_diag_config_1	0x1C3	0x2150
ls2_diag_config_2	0x1C4	0x0008
ls3_diag_config_1	0x1C6	0x2150
ls3_diag_config_2	0x1C7	0x0008
ls4_diag_config_1	0x1C9	0x2150
ls4_diag_config_2	0x1CA	0x0008
ls5_diag_config_1	0x1CC	0x2150
ls5_diag_config_2	0x1CD	0x0008
ls6_diag_config_1	0x1CF	0x2150
ls6_diag_config_2	0x1D0	0x0008
ls7_diag_config_1	0x1D2	0x2150
ls7_diag_config_2	0x1D3	0x0008
ls8_diag_config_1	0x1D5	0x2150
ls8_diag_config_2	0x1D6	0x0008

Register	Address	Init Value
seq0_entry_point	0x10A/0x12A/0x14A	0x0000
seq1_entry_point	0x10B/0x12B/0x14B	0x0000
diag_routine_addr	0x10C/0x12C/0x14C	0x0082
driver_disabled_routine_addr	0x10D/0x12D/0x14D	0x0104
DRAM00 - DRAM63 Ch1	0x000-0x03F	0x0000
DRAM00 - DRAM63 Ch2	0x040-0x07F	0x0000
DRAM00 - DRAM63 Ch3	0x080-0x0BF	0x0000

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D1	00	01	...

Parameters

parameters of initialise device command

5 (1*)	+0	+1	+2	+3
	POMT	**PIMT**	**POMR**	**PIMR**

... (1*)	+4	+5	+6	+7
	POCK	**PICK**	**POSL**	**PISL**

optional parameter of initialise device command

13 (0..n*)	+0	+2
	ADR	**VAL**

Details

parameters of initialise device command
Port and Pin parameters to select the SPI controller where the device is connected

Name	Length	Fmt	Description
POMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit Slave Receive) signal
PIMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within POMT of MTSR (Master Transmit Slave Receive) (SI) signal
POMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive Slave Transmit) signal
PIMR	**1**	**xx**	**Pin number of MRST signal** Pin number within POMR of MRST (Master Receive Slave Transmit) (SO) signal
POCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PICK	**1**	**xx**	**Pin number of SCK signal** Pin number within POCK of SCK (Serial Clock) signal

Name	Length	Fmt	Description
POSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PISL	**1**	**xx**	**Pin number of SLSO signal** Pin number within POSL of SLSO (Slave Select) signal

optional parameter of initialise device command
This can be used to initialise more registers or to change register values.

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>ADR</td><td>2</td><td>xxxx</td><td>Register address<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>15:10</td><td>reserved, must be zero</td></tr><tr><td>9:0</td><td>register address</td></tr></table></td></tr>
<tr><td>VAL</td><td>2</td><td>xxxx</td><td>Register value<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>15:0</td><td>register value</td></tr></table></td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D1	xx	xx	01	...

Parameters

Response of init device command

...	+0
(1*)	**CHIPID**

Details

Response of init device command

Name	Length	Fmt	Description
CHIPID	**2**	**xxxx**	**Chip identifier and revision** Chip identifier and revision from device identifier register (1B6h)(AEXX/AFXXh for eMGS3325)

.

01 - Load microcode

This command is used to download microcode into the device.

Example (CFITerm):

LOADFILE uCode.hex,,,45,80D10101ssttuuvvwwxxyyzzcc,C0D1000001,ff,last

downloads the microcode in file uCode.hex to the eMGS3325 connected to SPI with Ports/ Pins "ssttuuvvwwxxyyzz" into CRAM of Channel part of "cc". The Flash enable is set after the download, if not disabled in "cc".

NOTE: Ensure that parameter Segmentlength in LOADFILE (#4) is 45!

Table 2. Values of DGDI register written by command "Load microcode"

Register	Address	Value
unlock_word	0x113/0x133/0x153	0xBEEF
checksum_h	0x108/0x128/0x148	taken from Microcode file header
checksum_l	0x109/0x129/0x149	taken from Microcode file header
code_width	0x107/0x127/0x147	taken from Microcode file header
flash_enable	0x100/0x120/0x140	0x0018 (depends on CFG bit 7)
backup_clock_status_reg	0x1A8	0x0002

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D1	01	01	...

Parameters

parameters to select a SPI controller to the device

5	+0	+1	+2	+3
(1*)	**POMT**	**PIMT**	**POMR**	**PIMR**

...	+4	+5	+6	+7
(1*)	**POCK**	**PICK**	**POSL**	**PISL**

parameters of load microcode command

13	+0	+1
(1*)	**CFG**	**DATA**

Details

parameters to select a SPI controller to the device
Port and Pin parameters to select the SPI controller where the device is connected

Name	Length	Fmt	Description
POMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit Slave Receive) signal
PIMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within POMT of MTSR (Master Transmit Slave Receive) (SI) signal
POMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive Slave Transmit) signal
PIMR	**1**	**xx**	**Pin number of MRST signal** Pin number within POMR of MRST (Master Receive Slave Transmit) (SO) signal
POCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal

Name	Length	Fmt	Description
PICK	**1**	**xx**	**Pin number of SCK signal** Pin number within POCK of SCK (Serial Clock) signal
POSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PISL	**1**	**xx**	**Pin number of SLSO signal** Pin number within POSL of SLSO (Slave Select) signal

parameters of load microcode command

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>CFG</td><td>1</td><td>xx</td><td>Channel configuration (bitcoded)
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7</td><td>1: skip setting of Flash enable</td></tr>
<tr><td>6:5</td><td>reserved, must be zero</td></tr>
<tr><td>4</td><td>1: no check of chip revision with FunctionID bit 11 (see command 03)</td></tr>
<tr><td>3</td><td>1: load merged Microcode file (contain microcode for channels 1, 2 and 3)</td></tr>
<tr><td>2</td><td>1: load Microcode to CRAM of Channel 3</td></tr>
<tr><td>1</td><td>1: load Microcode to CRAM of Channel 2</td></tr>
<tr><td>0</td><td>1: load Microcode to CRAM of Channel 1</td></tr>
</table></td></tr>
<tr><td>DATA</td><td>1</td><td>xx..</td><td>Microcode data
32 data Bytes. Download of HEX file per "loadfile" function of cfiterm. Segmentlength = Prefix + 32 byte of Data</td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	D1	xx	xx	01

.

02 - Read status

This command is used to readout the status of the microcode data from the device

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D1	02	01	...

Parameters

parameters to select a SPI controller to the device

5 (1..n*)	+0 POMT	+1 PIMT	+2 POMR	+3 PIMR
... (1..n*)	+4 POCK	+5 PICK	+6 POSL	+7 PISL

parameters of read status command

... (1..n*)	+0 CFG

Details

parameters to select a SPI controller to the device
Port and Pin parameters to select the SPI controller where the device is connected

Name	Length	Fmt	Description
POMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit Slave Receive) signal
PIMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within POMT of MTSR (Master Transmit Slave Receive) (SI) signal
POMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive Slave Transmit) signal
PIMR	**1**	**xx**	**Pin number of MRST signal** Pin number within POMR of MRST (Master Receive Slave Transmit) (SO) signal
POCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PICK	**1**	**xx**	**Pin number of SCK signal** Pin number within POCK of SCK (Serial Clock) signal
POSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PISL	**1**	**xx**	**Pin number of SLSO signal** Pin number within POSL of SLSO (Slave Select) signal

parameters of read status command

Name	Length	Fmt	Description
CFG	**1**	**xx**	**Channel configuration (bitcoded)**

Bit(s)	Description
7:3	**reserved, must be zero**
2	**1: Microcode status of Channel 3**
1	**1: Microcode status of Channel 2**
0	**1: Microcode status of Channel 1**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D1	xx	xx	01	...

Parameters

response of command "Read Status"

6	+0	+2	+4
(1*)	**ERROR**	**VER**	**CRCH**

...	+6	+8	+10
(1*)	**CRCL**	**LEN**	**FLASH**

...	+12	+14	+16
(1*)	**CLOCK**	**START1**	**START2**

Details

response of command "Read Status"

Name	Length	Fmt	Description
ERROR	**2**	**xxxx**	**Error Status from device (bitcoded)**

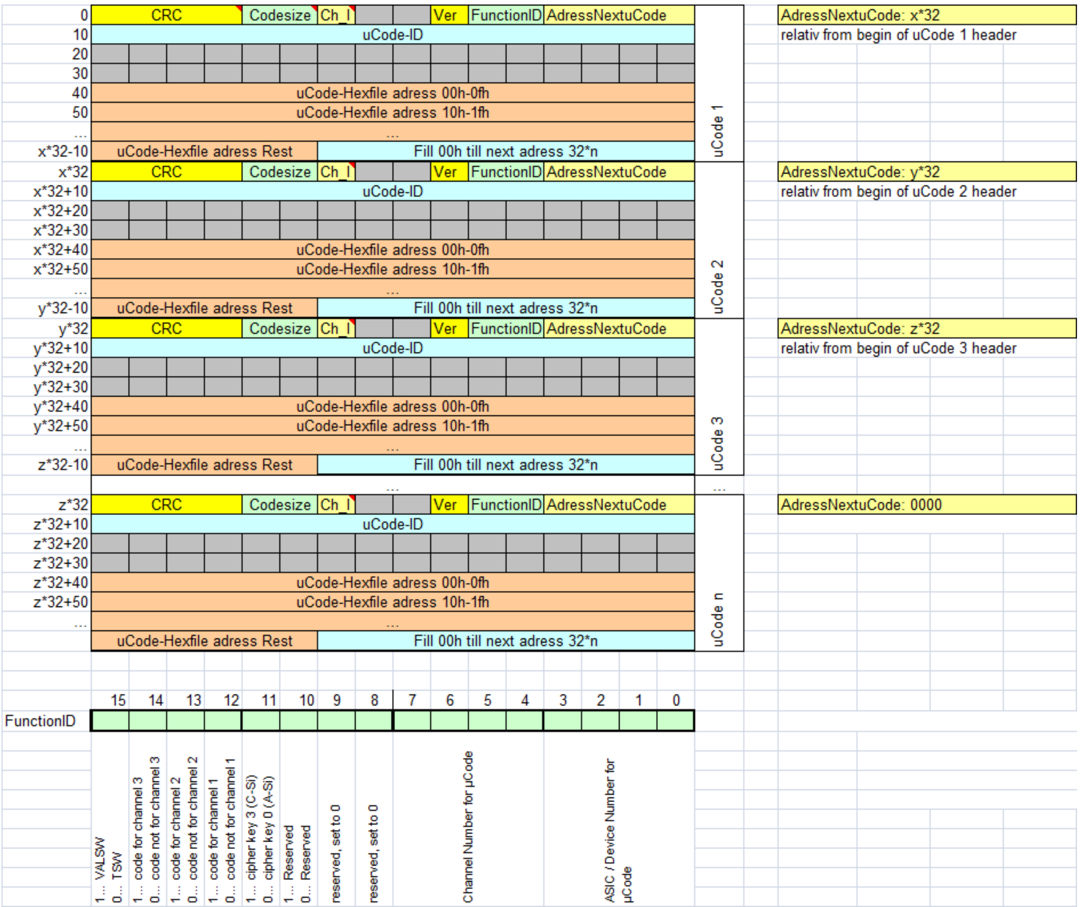
Bit(s)	Description
15	**1: Unexpected driver-disable-Interrupt channel 3**
14	**1: Unexpected diagnosis-Interrupt channel 3**
13	**1: Unexpected SW-Interrupt channel 3**
12	**1: Unimplemented test requested channel 3**
11	**1: Unexpected driver-disable-Interrupt channel 2**
10	**1: Unexpected diagnosis-Interrupt channel 2**
9	**1: Unexpected SW-Interrupt channel 2**
8	**1: Unimplemented test requested channel 2**
7	**1: Unexpected driver-disable-Interrupt channel 1**
6	**1: Unexpected diagnosis-Interrupt channel 1**
5	**1: Unexpected SW-Interrupt channel 1**
4	**1: Unimplemented test requested channel 1**
3	**1: DCDC-parameter CRC failed**

Name	Length	Fmt	Description
			Bit(s) / **Description**
			2: **1: DGDI has loss of external clock**
			1: **1: Microcode has checksum error**
			0: **1: Microcode is not ready**
VER	**2**	**xxxx**	**Version DGDI microcode**
CRCH	**2**	**xxxx**	**DGDI register checksum high**
CRCL	**2**	**xxxx**	**DGDI register checksum low**
LEN	**2**	**xxxx**	**DGDI register code_width**
FLASH	**2**	**xxxx**	**DGDI register flash_enable**
CLOCK	**2**	**xxxx**	**DGDI register backup_clock_status_reg**
START1	**2**	**xxxx**	**DGDI register start_config_reg, part 1**
START2	**2**	**xxxx**	**DGDI register start_config_reg, part 2**

.

03 - Load microcode from memory

This command is used to download microcode data from ECU memory area to DGDI. **Figure 1.** structure of microcode(s) in memory



The specified address needs to contain at least one valid microcode hex file including it's header information which contains length, version, checksum, Function-ID etc. (see figure "structure of microcode(s) in memory").

The specified address may contain a list of multiple different microcode files which are linked through their header information. Any individual file is identified by the start address of the list and the Function-ID of the microcode. (see figure "structure of microcode(s) in memory").

The search for the specified Function-ID of the microcode will start at the specified address and trace through the list of available microcode files within the list until the specified Function ID is found. If no matching Function-ID is found, the test step returns an error.

The test step will download the microcode only, if the code length in the header block is between 0 and maximum number of CRAM cells.

In condition of a verification failure, the test step returns an error.

After the download, the DGDI device checks the downloaded microcode against the stored checksum to verify correct download to the device.

In condition of a verification failure of the DGDI device, the test step returns an error.

In order to check the activation status of the downloaded microcode after a download, the CRAM status should be checked for every affected channel! (see command "Read status").

Important: In difference to eMGS625 (TSW_D0) the next uCode is addressed relative.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D1	03	01	...

Parameters

parameters to select a SPI controller to the device

5	+0	+1	+2	+3
(1*)	**POMT**	**PIMT**	**POMR**	**PIMR**

...	+4	+5	+6	+7
(1*)	**POCK**	**PICK**	**POSL**	**PISL**

parameters of load microcode from memory command

13	+0	+1	+5
(1..n*)	**CFG**	**ADDR**	**FUNC_ID**

Details

parameters to select a SPI controller to the device
Port and Pin parameters to select the SPI controller where the device is connected

Name	Length	Fmt	Description
POMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit Slave Receive) signal
PIMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within POMT of MTSR (Master Transmit Slave Receive) (SI) signal

Name	Length	Fmt	Description
POMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive Slave Transmit) signal
PIMR	**1**	**xx**	**Pin number of MRST signal** Pin number within POMR of MRST (Master Receive Slave Transmit) (SO) signal
POCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PICK	**1**	**xx**	**Pin number of SCK signal** Pin number within POCK of SCK (Serial Clock) signal
POSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PISL	**1**	**xx**	**Pin number of SLSO signal** Pin number within POSL of SLSO (Slave Select) signal

parameters of load microcode from memory command
parameters for Cfg, Address and function ID

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>CFG</td><td>1</td><td>xx</td><td>Channel configuration (bitcoded)
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7</td><td>1: skip setting of Flash enable</td></tr>
<tr><td>6:3</td><td>reserved, must be zero</td></tr>
<tr><td>2</td><td>1: load Microcode to CRAM of Channel 3</td></tr>
<tr><td>1</td><td>1: load Microcode to CRAM of Channel 2</td></tr>
<tr><td>0</td><td>1: load Microcode to CRAM of Channel 1</td></tr>
</table></td></tr>
<tr><td>ADDR</td><td>4</td><td>xxxxxxxx</td><td>Start address of microcode data area
Physical address of ECU memory (Flash/RAM) which contains microcode image(s).</td></tr>
<tr><td>FUNC_ID</td><td>2</td><td>xxxx</td><td>Function ID of microcode to be downloaded
unique search pattern to specify microcode within list of available microcodes in memory</td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	D1	xx	xx	01

.

40 - Offset compensation

This command is used to start the microcode in the device for offset compensation of the current measurement blocks.

> **Note:** The offset compensation has to be done for every used current measurement channels difference amplifier with a separat call of command "Offset Compensation". No signal must be applied to the amplifier while the calibration is done.
>
> If the calibration is ommitted or called while the difference amplifier input signal level is not zero, the accuracy of the input amplifier will be very low.

Table 3. Initialisation values of DGDI register written by command "Offset Compensation"

Register	Address	Init Value
oa_out1_config	0x197	0x0000
oa_out2_config	0x198	0x0000
oa_out3_config	0x199	0x0000
DRAM0	0x000	taken from command parameter
ctrl_req_seq0	0x101	Bit 2 set to start Microcode

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D1	40	01	...

Parameters

parameters to select a SPI controller to the device

5	+0	+1	+2	+3
(1*)	**POMT**	**PIMT**	**POMR**	**PIMR**

...	+4	+5	+6	+7
(1*)	**POCK**	**PICK**	**POSL**	**PISL**

para. for timeout and offset values

13	+0	+2
(1..n*)	**TIM**	**CFG**

Details

parameters to select a SPI controller to the device
Port and Pin parameters to select the SPI controller where the device is connected

Name	Length	Fmt	Description
POMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit Slave Receive) signal
PIMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within POMT of MTSR (Master Transmit Slave Receive) (SI) signal
POMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive Slave Transmit) signal
PIMR	**1**	**xx**	**Pin number of MRST signal**

Name	Length	Fmt	Description
			Pin number within POMR of MRST (Master Receive Slave Transmit) (SO) signal
POCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PICK	**1**	**xx**	**Pin number of SCK signal** Pin number within POCK of SCK (Serial Clock) signal
POSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PISL	**1**	**xx**	**Pin number of SLSO signal** Pin number within POSL of SLSO (Slave Select) signal

para. for timeout and offset values
parameters for time out and Offset compensation parameters

Name	Length	Fmt	Description
TIM	**2**	**xxxx**	**device timeout** Selection of timeout on waiting for device test status ready (Format: RVVV).
CFG	**2**	**xxxx**	**Offset compensation parameters**

Bit(s)	Description
15:11	**not used, must be set to 0**
10:8	**Select current measurement channel** **000_B:** not used **001_B:** Channel 1 **010_B:** Channel 2 **011_B:** Channel 3 **100_B:** Channel 4 **101_B:** Channel 5 **110_B:** Channel 6
7:6	**not used, must be set to 0**
5:4	**Select amplification for current measurement** **00_B:** 5.79 **01_B:** 8.68 **10_B:** 12.53 **11_B:** 19.25
3:0	**not used, must be set to 0**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D1	xx	xx	01	...

Parameters

Error response of command "Offset Compensation"

6	+0	+2
(1*)	**ERROR**	**OFF**

Details

Error response of command "Offset Compensation"

Name	Length	Fmt	Description
ERROR	**2**	**xxxx**	**Error Status DGDI (bitcoded)** **Bit(s)** – **Description** **15** – **1: Unexpected driver-disable-Interrupt channel 3** **14** – **1: Unexpected diagnosis-Interrupt channel 3** **13** – **1: Unexpected SW-Interrupt channel 3** **12** – **1: Unimplemented test requested channel 3** **11** – **1: Unexpected driver-disable-Interrupt channel 2** **10** – **1: Unexpected diagnosis-Interrupt channel 2** **9** – **1: Unexpected SW-Interrupt channel 2** **8** – **1: Unimplemented test requested channel 2** **7** – **1: Unexpected driver-disable-Interrupt channel 1** **6** – **1: Unexpected diagnosis-Interrupt channel 1** **5** – **1: Unexpected SW-Interrupt channel 1** **4** – **1: Unimplemented test requested channel 1** **3** – **1: DCDC-parameter CRC failed** **2** – **1: DGDI has loss of external clock** **1** – **1: Microcode has checksum error** **0** – **1: Microcode is not ready**
OFF	**2**	**xxxx**	**DGDI register offset** Offset value for selected measurement block

.

41 - Initialise and start LSHS test

This command is used to initialise the configuration and to start the microcode in DGDI for LSHS test.

Table 4. Initialisation values of DGDI register written by command "Initialise and start LSHS test"

Register	Address	Init Value
hs_slewrate	0x171	0x2AAA
ls_slewrate_part1	0x172	0x0FFF
flags_direction	0x1A1	0xFFFF
flags_polarity	0x1A2	0x0000
flags_source	0x1A3	0xA01F
start_config_part1_ch1	0x103	taken from command parameter
oa_out1_config	0x197	taken from command parameter
oa_out2_config	0x198	taken from command parameter
oa_out3_config	0x199	taken from command parameter
meas_func1	0x19A	taken from command parameter
hs_bias_config	0x18B	taken from command parameter
ls_bias_config	0x18C	taken from command parameter
DRAM0-DRAM16	0x000-0x010	taken from command parameter
ctrl_reg_seq0	0x101	Bit 3 set to start Microcode

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D1	41	01	...

Parameters

parameters to select a SPI controller to the device

5 (1*)	+0	+1	+2	+3
	POMT	**PIMT**	**POMR**	**PIMR**

... (1*)	+4	+5	+6	+7
	POCK	**PICK**	**POSL**	**PISL**

para. for LSHS tests

13 (1*)	+0	+2	+4
	TIM	**GENIO**	**GENM**

para. for LS and HS

... (1*)	+0
	DRIV

Details

parameters to select a SPI controller to the device

Port and Pin parameters to select the SPI controller where the device is connected

Name	Length	Fmt	Description
POMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit Slave Receive) signal
PIMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within POMT of MTSR (Master Transmit Slave Receive) (SI) signal
POMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive Slave Transmit) signal
PIMR	**1**	**xx**	**Pin number of MRST signal** Pin number within POMR of MRST (Master Receive Slave Transmit) (SO) signal
POCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PICK	**1**	**xx**	**Pin number of SCK signal** Pin number within POCK of SCK (Serial Clock) signal
POSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PISL	**1**	**xx**	**Pin number of SLSO signal** Pin number within POSL of SLSO (Slave Select) signal

para. for LSHS tests
parameters for LSHS tests

Name	Length	Fmt	Description
TIM	**2**	**xxxx**	**DGDI timeout** Selection of timeout on waiting for DGDI test status ready (Format: RVVV).
GENIO	**2**	**xxxx**	**General DGDI input/output configuration parameters**

Bit(s)	Description
15:10	**not used, must be set to 0**
9:8	**Select OAx for current measurement** **00_B:** OA1, OA2, OA3 high impedance **01_B:** OA1 current measurement, OA2, OA3 high impedance **10_B:** OA2 current measurement, OA1, OA3 high impedance **11_B:** OA3 current measurement, OA1, OA2 high impedance
7:6	**not used, must be set to 0**
5:4	**Gain setting of OAx amplifier** **00_B:** 1.33

Name	Length	Fmt	Description
			Bit(s) / **Description**
			01_B: 2.00 **10_B:** 3.00 **11_B:** 5.33
			3:0 — **Select sensitivity to start signals** **0000_B:** not used **0001_B:** Start1 **0010_B:** Start2 **0011_B:** Start3 **0100_B:** Start4 **0101_B:** Start5 **0110_B:** Start6 **0111_B:** Start7 **1000_B:** Start8
GENM	**2**	**xxxx**	**General DGDI measurement configuration parameters**
			Bit(s) / **Description**
			15:11 — **not used, must be set to 0**
			10:8 — **Select current measurement channel** **000_B:** not used **001_B:** Channel 1 **010_B:** Channel 2 **011_B:** Channel 3 **100_B:** Channel 4 **101_B:** Channel 5 **110_B:** Channel 6
			7:6 — **not used, must be set to 0**
			5:4 — **Select amplification for current measurement** **00_B:** 5.79 **01_B:** 8.68 **10_B:** 12.53 **11_B:** 19.25
			3:0 — **not used, must be set to 0**

para. for LS and HS
parameters for 8 LS and 7 HS

Name	Length	Fmt	Description
DRIV	**2**	**xxxx..**	**DGDI pre-driver configuration parameters** DGDI pre-driver configuration parameters for LS1 (exemplary for the following parameters)

Name	Length	Fmt	Description
			Repeat bytes 19 - 20 for all LS and HS predrivers of DGDI. The required sequence is LS2,LS3,LS4,LS5,LS6,LS7,LS8,HS1,HS2,HS3,HS4,HS5,HS6,HS7.

Bit(s)	Description
15:4	**not used, must be set to 0**
3	**Bias** **0_B:** Off **1_B:** On
2	**Pre-driver level** **0_B:** Off **1_B:** On
1	**Inversion to state of Start signal** **0_B:** Pre-driver is not inverted to state of Start signal **1_B:** Pre-driver is inverted to state of Start signal
0	**Dependence on Start signal** **0_B:** Pre-driver is set to state according to Pre-driver level (Bit 2) **1_B:** Pre-driver is switched with Start signal **Note:** Start signal is selected with GENIO bit 0-2

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D1	xx	xx	01	...

Parameters

Response of command "Initialise and start LSHS test"

6	+0	+2	+4
(1*)	**ERROR**	**LS_FBS**	**HS_FBS**

Details

Response of command "Initialise and start LSHS test"

Name	Length	Fmt	Description
ERROR	**2**	**xxxx**	**Error Status DGDI (bitcoded)**

Name	Length	Fmt	Description	
			Bit(s)	**Description**
			15	**1: Unexpected driver-disable-Interrupt channel 3**
			14	**1: Unexpected diagnosis-Interrupt channel 3**
			13	**1: Unexpected SW-Interrupt channel 3**
			12	**1: Unimplemented test requested channel 3**
			11	**1: Unexpected driver-disable-Interrupt channel 2**
			10	**1: Unexpected diagnosis-Interrupt channel 2**
			9	**1: Unexpected SW-Interrupt channel 2**
			8	**1: Unimplemented test requested channel 2**
			7	**1: Unexpected driver-disable-Interrupt channel 1**
			6	**1: Unexpected diagnosis-Interrupt channel 1**
			5	**1: Unexpected SW-Interrupt channel 1**
			4	**1: Unimplemented test requested channel 1**
			3	**1: DCDC-parameter CRC failed**
			2	**1: DGDI has loss of external clock**
			1	**1: Microcode has checksum error**
			0	**1: Microcode is not ready**
LS_FBS	**2**	**xxxx**	**LS feedback states after predriver setting**	
			Bit(s)	**Description**
			15:8	**reserved, set to zero**
			7	**LS8 VDS feedback state**
			6	**LS7 VDS feedback state**
			5	**LS6 VDS feedback state**
			4	**LS5 VDS feedback state**
			3	**LS4 VDS feedback state**
			2	**LS3 VDS feedback state**
			1	**LS2 VDS feedback state**
			0	**LS1 VDS feedback state**
HS_FBS	**2**	**xxxx**	**HS feedback states after predriver setting**	
			Bit(s)	**Description**
			15	**reserved, set to zero**

Name	Length	Fmt	Description	
			Bit(s)	**Description**
			14	**HS7 SRC feedback state**
			13	**HS6 SRC feedback state**
			12	**HS5 SRC feedback state**
			11	**HS4 SRC feedback state**
			10	**HS3 SRC feedback state**
			9	**HS2 SRC feedback state**
			8	**HS1 SRC feedback state**
			7	**reserved, set to zero**
			6	**HS7 VDS feedback state**
			5	**HS6 VDS feedback state**
			4	**HS5 VDS feedback state**
			3	**HS4 VDS feedback state**
			2	**HS3 VDS feedback state**
			1	**HS2 VDS feedback state**
			0	**HS1 VDS feedback state**

.

42 - ADC test

This command is used to read out the ADC measurement values of the current measurement blocks.

Table 5. Initialisation values of DGDI register written by command "ADC test"

Register	Address	Init Value
flags_source	0x1A3	taken from command parameter
DRAM0-DRAM1	0x000-0x001	taken from command parameter
ctrl_reg_seq0	0x101	Bit 6 set to start Microcode

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D1	42	01	...

Parameters

parameters to select a SPI controller to the device

5	+0	+1	+2	+3
(1*)	**POMT**	**PIMT**	**POMR**	**PIMR**

...	+4	+5	+6	+7
(1*)	**POCK**	**PICK**	**POSL**	**PISL**

para. for ADC test

13	+0	+2	+4
(1..n*)	**TIM**	**BLOCK**	**SAMP**

Details

parameters to select a SPI controller to the device
Port and Pin parameters to select the SPI controller where the device is connected

Name	Length	Fmt	Description
POMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit Slave Receive) signal
PIMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within POMT of MTSR (Master Transmit Slave Receive) (SI) signal
POMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive Slave Transmit) signal
PIMR	**1**	**xx**	**Pin number of MRST signal** Pin number within POMR of MRST (Master Receive Slave Transmit) (SO) signal
POCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PICK	**1**	**xx**	**Pin number of SCK signal** Pin number within POCK of SCK (Serial Clock) signal
POSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PISL	**1**	**xx**	**Pin number of SLSO signal** Pin number within POSL of SLSO (Slave Select) signal

para. for ADC test
parameters for ADC test

Name	Length	Fmt	Description
TIM	**2**	**xxxx**	**DGDI timeout** Selection of timeout on waiting for DGDI test status ready (Format: RVVV).
BLOCK	**2**	**xxxx**	**Current measurement block parameters**

Bit(s)	Description
15:11	**not used, must be set to 0**
10:8	**Select current measurement channel** **000_B:** not used **001_B:** Channel 1 **010_B:** Channel 2 **011_B:** Channel 3 **100_B:** Channel 4

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td><table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td></td><td>101B: Channel 5
110B: Channel 6</td></tr>
<tr><td>7:6</td><td>not used, must be set to 0</td></tr>
<tr><td>5:4</td><td>Select amplification for current measurement
00B: 5.79
01B: 8.68
10B: 12.53
11B: 19.25</td></tr>
<tr><td>3:0</td><td>not used, must be set to 0</td></tr>
</table></td></tr>
<tr><td>SAMP</td><td>2</td><td>xxxx</td><td>ADC measurement parameters<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>15:4</td><td>not used, must be set to 0</td></tr>
<tr><td>3:0</td><td>Number of samples in two’s exponent
0H: 1 sample
1H: 2 sample
2H: 4 sample
3H: 8 sample
4H: 16 sample
5H: 32 sample
6H: 64 sample
7H: 128 sample
8H: 256 sample
9H: 256 sample
AH: 256 sample
BH: 256 sample
CH: 256 sample
DH: 256 sample
EH: 256 sample
FH: 256 sample</td></tr>
</table></td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D1	xx	xx	01	...

Parameters

Error response of command "ADC test"

6	+0	+2	+3
(1*)	**ERROR**	**AVER**	**ACCUM**

Details

Error response of command "ADC test"

Note: The ADC values in response bytes AVER and ACCUM can be transformed to current values with formular:

$$I = \frac{ADC_{value} * 9{,}77mV - 250mV}{G_{DA\_diff} * R_{Shunt}}$$

Name	Length	Fmt	Description
ERROR	**2**	**xxxx**	**Error Status DGDI (bitcoded)**

Bit(s)	Description
15	**1: Unexpected driver-disable-Interrupt channel 3**
14	**1: Unexpected diagnosis-Interrupt channel 3**
13	**1: Unexpected SW-Interrupt channel 3**
12	**1: Unimplemented test requested channel 3**
11	**1: Unexpected driver-disable-Interrupt channel 2**
10	**1: Unexpected diagnosis-Interrupt channel 2**
9	**1: Unexpected SW-Interrupt channel 2**
8	**1: Unimplemented test requested channel 2**
7	**1: Unexpected driver-disable-Interrupt channel 1**
6	**1: Unexpected diagnosis-Interrupt channel 1**
5	**1: Unexpected SW-Interrupt channel 1**
4	**1: Unimplemented test requested channel 1**
3	**1: DCDC-parameter CRC failed**
2	**1: DGDI has loss of external clock**
1	**1: Microcode has checksum error**
0	**1: Microcode is not ready**

Name	Length	Fmt	Description
AVER	**1**	**xx**	**Averaged samples** Average of all sampled ADC values
ACCUM	**2**	**xxxx**	**Accumulated samples**

Name	Length	Fmt	Description
			Sum of all sampled ADC values. For reduced rounding error compared to averaged samples.

.

43 - Digital IO test

This command is used to switch the pins of the flag bus in DGDI to digital input. In the respose the read status of the pins is provided.

Table 6. Initialisation values of DGDI register written by command "Digital IO test"

Register	Address	Init Value
flags_direction	0x1A1	0xFFFF
flags_polarity	0x1A2	0x0000
flags_source	0x1A3	0xFFFF
ctrl_reg_seq0	0x101	Bit 5 set to start Microcode

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D1	43	01	...

Parameters

parameters to select a SPI controller to the device

5	+0	+1	+2	+3
(1*)	**POMT**	**PIMT**	**POMR**	**PIMR**

...	+4	+5	+6	+7
(1*)	**POCK**	**PICK**	**POSL**	**PISL**

para. for dig IO test

13	+0	+2
(1*)	**TIM**	**MASK**

Details

parameters to select a SPI controller to the device
Port and Pin parameters to select the SPI controller where the device is connected

Name	Length	Fmt	Description
POMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit Slave Receive) signal

Name	Length	Fmt	Description
PIMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within POMT of MTSR (Master Transmit Slave Receive) (SI) signal
POMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive Slave Transmit) signal
PIMR	**1**	**xx**	**Pin number of MRST signal** Pin number within POMR of MRST (Master Receive Slave Transmit) (SO) signal
POCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PICK	**1**	**xx**	**Pin number of SCK signal** Pin number within POCK of SCK (Serial Clock) signal
POSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PISL	**1**	**xx**	**Pin number of SLSO signal** Pin number within POSL of SLSO (Slave Select) signal

para. for dig IO test
parameters for digital IO test

Name	Length	Fmt	Description
TIM	**2**	**xxxx**	**DGDI timeout** Selection of timeout on waiting for DGDI test status ready (Format: RVVV).
MASK	**2**	**xxxx**	**Mask pins as digital input**

Bit(s)	Description
15	**Dbg**
14	**OA2**
13	**IrqB**
12	**Start8**
11	**Start7**
10	**Start6**
9	**Start5**
8	**Start4**
7	**Start3**
6	**Start2**
5	**Start1**
4	**Flag4**
3	**Flag3**
2	**Flag2**
1	**Flag1**
0	**Flag0**
15:0	**Description for all pins**

Name	Length	Fmt	Description
			Bit(s) | Description
			| **0_B:** not read
			| **1_B:** read

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D1	xx	xx	01	...

Parameters

Error response of command "Digital IO test"

6	+0	+2
(1*)	**ERROR**	**STAT**

Details

Error response of command "Digital IO test"

Name	Length	Fmt	Description	
ERROR	**2**	**xxxx**	**Error Status DGDI (bitcoded)**	
			Bit(s)	**Description**
			15	**1: Unexpected driver-disable-Interrupt channel 3**
			14	**1: Unexpected diagnosis-Interrupt channel 3**
			13	**1: Unexpected SW-Interrupt channel 3**
			12	**1: Unimplemented test requested channel 3**
			11	**1: Unexpected driver-disable-Interrupt channel 2**
			10	**1: Unexpected diagnosis-Interrupt channel 2**
			9	**1: Unexpected SW-Interrupt channel 2**
			8	**1: Unimplemented test requested channel 2**
			7	**1: Unexpected driver-disable-Interrupt channel 1**
			6	**1: Unexpected diagnosis-Interrupt channel 1**
			5	**1: Unexpected SW-Interrupt channel 1**
			4	**1: Unimplemented test requested channel 1**

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td></td><td>Bit(s)</td><td>Description</td></tr>
<tr><td></td><td></td><td></td><td>3</td><td>1: DCDC-parameter CRC failed</td></tr>
<tr><td></td><td></td><td></td><td>2</td><td>1: DGDI has loss of external clock</td></tr>
<tr><td></td><td></td><td></td><td>1</td><td>1: Microcode has checksum error</td></tr>
<tr><td></td><td></td><td></td><td>0</td><td>1: Microcode is not ready</td></tr>
<tr><td>STAT</td><td>2</td><td>xxxx</td><td colspan="2">Status of test responsed from microcode</td></tr>
<tr><td></td><td></td><td></td><td>Bit(s)</td><td>Description</td></tr>
<tr><td></td><td></td><td></td><td>15</td><td>Dbg</td></tr>
<tr><td></td><td></td><td></td><td>14</td><td>OA2</td></tr>
<tr><td></td><td></td><td></td><td>13</td><td>IrqB</td></tr>
<tr><td></td><td></td><td></td><td>12</td><td>Start8</td></tr>
<tr><td></td><td></td><td></td><td>11</td><td>Start7</td></tr>
<tr><td></td><td></td><td></td><td>10</td><td>Start6</td></tr>
<tr><td></td><td></td><td></td><td>9</td><td>Start5</td></tr>
<tr><td></td><td></td><td></td><td>8</td><td>Start4</td></tr>
<tr><td></td><td></td><td></td><td>7</td><td>Start3</td></tr>
<tr><td></td><td></td><td></td><td>6</td><td>Start2</td></tr>
<tr><td></td><td></td><td></td><td>5</td><td>Start1</td></tr>
<tr><td></td><td></td><td></td><td>4</td><td>Flag4</td></tr>
<tr><td></td><td></td><td></td><td>3</td><td>Flag3</td></tr>
<tr><td></td><td></td><td></td><td>2</td><td>Flag2</td></tr>
<tr><td></td><td></td><td></td><td>1</td><td>Flag1</td></tr>
<tr><td></td><td></td><td></td><td>0</td><td>Flag0</td></tr>
<tr><td></td><td></td><td></td><td>15:0</td><td>Description for all pins
0_B: Signal LOW or not read
1_B: Signal HIGH</td></tr>
</table>

.

44 - Advanced Digital IO test

This command is used to switch the pins of the flag bus in DGDI to digital input or output. In the respose the read status of the pins is provided.

Table 7. Initialisation values of DGDI register written by command "Digital IO test"

Register	Address	Init Value
flags_direction	0x1A1	taken from command parameter
flags_polarity	0x1A2	0x0000
flags_source	0x1A3	0xFFFF
DRAM50	0x032	taken from command parameter
ctrl_reg_seq0	0x101	Bit 7 set to start Microcode

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D1	44	01	...

Parameters

parameters to select a SPI controller to the device

5	+0	+1	+2	+3
(1*)	**POMT**	**PIMT**	**POMR**	**PIMR**

...	+4	+5	+6	+7
(1*)	**POCK**	**PICK**	**POSL**	**PISL**

para. for dig IO test

13	+0	+2	+4	+6
(1*)	**TIM**	**IOSEL**	**OUTVAL**	**MASK**

Details

parameters to select a SPI controller to the device
Port and Pin parameters to select the SPI controller where the device is connected

Name	Length	Fmt	Description
POMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit Slave Receive) signal
PIMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within POMT of MTSR (Master Transmit Slave Receive) (SI) signal
POMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive Slave Transmit) signal
PIMR	**1**	**xx**	**Pin number of MRST signal** Pin number within POMR of MRST (Master Receive Slave Transmit) (SO) signal
POCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PICK	**1**	**xx**	**Pin number of SCK signal** Pin number within POCK of SCK (Serial Clock) signal
POSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PISL	**1**	**xx**	**Pin number of SLSO signal** Pin number within POSL of SLSO (Slave Select) signal

para. for dig IO test

parameters for digital IO test

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>TIM</td><td>2</td><td>xxxx</td><td>DGDI timeout
Selection of timeout on waiting for DGDI test status ready (Format: RVVV).</td></tr>
<tr><td>IOSEL</td><td>2</td><td>xxxx</td><td>Input/Output selection
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>15</td><td>Dbg</td></tr>
<tr><td>14</td><td>OA2</td></tr>
<tr><td>13</td><td>IrqB</td></tr>
<tr><td>12</td><td>Start8</td></tr>
<tr><td>11</td><td>Start7</td></tr>
<tr><td>10</td><td>Start6</td></tr>
<tr><td>9</td><td>Start5</td></tr>
<tr><td>8</td><td>Start4</td></tr>
<tr><td>7</td><td>Start3</td></tr>
<tr><td>6</td><td>Start2</td></tr>
<tr><td>5</td><td>Start1</td></tr>
<tr><td>4</td><td>Flag4</td></tr>
<tr><td>3</td><td>Flag3</td></tr>
<tr><td>2</td><td>Flag2</td></tr>
<tr><td>1</td><td>Flag1</td></tr>
<tr><td>0</td><td>Flag0</td></tr>
<tr><td>15:0</td><td>Description for all pins
0_B: output
1_B: input</td></tr>
</table></td></tr>
<tr><td>OUTVAL</td><td>2</td><td>xxxx</td><td>Output value
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>15</td><td>Dbg</td></tr>
<tr><td>14</td><td>OA2</td></tr>
<tr><td>13</td><td>IrqB</td></tr>
<tr><td>12</td><td>Start8</td></tr>
<tr><td>11</td><td>Start7</td></tr>
<tr><td>10</td><td>Start6</td></tr>
<tr><td>9</td><td>Start5</td></tr>
<tr><td>8</td><td>Start4</td></tr>
<tr><td>7</td><td>Start3</td></tr>
<tr><td>6</td><td>Start2</td></tr>
<tr><td>5</td><td>Start1</td></tr>
<tr><td>4</td><td>Flag4</td></tr>
<tr><td>3</td><td>Flag3</td></tr>
</table></td></tr>
</table>

Name	Length	Fmt	Description

Bit(s)	Description
2	**Flag2**
1	**Flag1**
0	**Flag0**
15:0	**Description for all pins** **0_B:** LOW **1_B:** HIGH

Name	Length	Fmt	Description
MASK	**2**	**xxxx**	**Mask pins as digital input**

Bit(s)	Description
15	**Dbg**
14	**OA2**
13	**IrqB**
12	**Start8**
11	**Start7**
10	**Start6**
9	**Start5**
8	**Start4**
7	**Start3**
6	**Start2**
5	**Start1**
4	**Flag4**
3	**Flag3**
2	**Flag2**
1	**Flag1**
0	**Flag0**
15:0	**Description for all pins** **0_B:** not read **1_B:** read

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D1	xx	xx	01	...

Parameters

Error response of command "Digital IO test"

6 (1*)	+0	+2
	ERROR	**STAT**

Details

Error response of command "Digital IO test"

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>ERROR</td><td>2</td><td>xxxx</td><td>Error Status DGDI (bitcoded)
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>15</td><td>1: Unexpected driver-disable-Interrupt channel 3</td></tr>
<tr><td>14</td><td>1: Unexpected diagnosis-Interrupt channel 3</td></tr>
<tr><td>13</td><td>1: Unexpected SW-Interrupt channel 3</td></tr>
<tr><td>12</td><td>1: Unimplemented test requested channel 3</td></tr>
<tr><td>11</td><td>1: Unexpected driver-disable-Interrupt channel 2</td></tr>
<tr><td>10</td><td>1: Unexpected diagnosis-Interrupt channel 2</td></tr>
<tr><td>9</td><td>1: Unexpected SW-Interrupt channel 2</td></tr>
<tr><td>8</td><td>1: Unimplemented test requested channel 2</td></tr>
<tr><td>7</td><td>1: Unexpected driver-disable-Interrupt channel 1</td></tr>
<tr><td>6</td><td>1: Unexpected diagnosis-Interrupt channel 1</td></tr>
<tr><td>5</td><td>1: Unexpected SW-Interrupt channel 1</td></tr>
<tr><td>4</td><td>1: Unimplemented test requested channel 1</td></tr>
<tr><td>3</td><td>1: DCDC-parameter CRC failed</td></tr>
<tr><td>2</td><td>1: DGDI has loss of external clock</td></tr>
<tr><td>1</td><td>1: Microcode has checksum error</td></tr>
<tr><td>0</td><td>1: Microcode is not ready</td></tr>
</table></td></tr>
<tr><td>STAT</td><td>2</td><td>xxxx</td><td>Status of test responsed from microcode
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>15</td><td>Dbg</td></tr>
<tr><td>14</td><td>OA2</td></tr>
<tr><td>13</td><td>IrqB</td></tr>
<tr><td>12</td><td>Start8</td></tr>
<tr><td>11</td><td>Start7</td></tr>
<tr><td>10</td><td>Start6</td></tr>
<tr><td>9</td><td>Start5</td></tr>
<tr><td>8</td><td>Start4</td></tr>
<tr><td>7</td><td>Start3</td></tr>
</table></td></tr>
</table>

Name	Length	Fmt	Description	
			Bit(s)	**Description**
			6	**Start2**
			5	**Start1**
			4	**Flag4**
			3	**Flag3**
			2	**Flag2**
			1	**Flag1**
			0	**Flag0**
			15:0	**Description for all pins** **0_B:** Signal LOW or not read **1_B:** Signal HIGH

.

50 - DCDC converter setup

This command is used to initialise the configuration in DGDI for DCDC converter test.

Table 8. Initialisation values of DGDI register written by command "DCDC converter setup"

Register	Address	Init Value
boost_dac	0x17F	taken from command parameter
out_acc_seq0_ch3	0x164	taken from command parameter
out_acc_seq1_ch3	0x165	taken from command parameter
dac12	0x185	taken from command parameter
dac34	0x186	taken from command parameter
dac5l5h	0x187	taken from command parameter
dac5neg	0x188	taken from command parameter
dac6l6h	0x189	taken from command parameter
dac6neg	0x18A	taken from command parameter
cur_access_reg_ch3	0x168	taken from command parameter
vds7_dcdc_config	0x182	taken from command parameter
vds8_dcdc_config	0x183	taken from command parameter
dac_rxtx_cr_config_ch3	0x152	taken from command parameter
ls_slewrate_part2	0x173	taken from command parameter
oa_out1_config	0x197	taken from command parameter
oa_out2_config	0x198	taken from command parameter
oa_out3_config	0x199	taken from command parameter
meas_func1	0x19A	taken from command parameter
DRAM0-DRAM3 Ch3	0x080-0x083	taken from command parameter
DRAM13 Ch3	0x08D	taken from command parameter
DRAM16-DRAM19 Ch3	0x090-0x093	taken from command parameter

Register	Address	Init Value
DRAM29 Ch3	0x09D	taken from command parameter
ctrl_reg_seq0/1	0x141/0x142	Bit 0 set to start Microcode

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D1	50	01	...

Parameters

parameters to select a SPI controller to the device

5 (1*)	+0	+1	+2	+3
	POMT	**PIMT**	**POMR**	**PIMR**

... (1*)	+4	+5	+6	+7
	POCK	**PICK**	**POSL**	**PISL**

para. for DCDC setup

13 (1*)	+0	+2	+3	+5
	TIM	**CFG**	**CURM**	**CURL**

... (1*)	+7	+9	+11	+13
	CURN	**BOOSTV**	**TMAX**	**SLEW**

Details

parameters to select a SPI controller to the device
Port and Pin parameters to select the SPI controller where the device is connected

Name	Length	Fmt	Description
POMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit Slave Receive) signal
PIMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within POMT of MTSR (Master Transmit Slave Receive) (SI) signal
POMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive Slave Transmit) signal
PIMR	**1**	**xx**	**Pin number of MRST signal** Pin number within POMR of MRST (Master Receive Slave Transmit) (SO) signal
POCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PICK	**1**	**xx**	**Pin number of SCK signal** Pin number within POCK of SCK (Serial Clock) signal
POSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal

Name	Length	Fmt	Description
PISL	**1**	**xx**	**Pin number of SLSO signal** Pin number within POSL of SLSO (Slave Select) signal

para. for DCDC setup
parameters for DCDC setup

Name	Length	Fmt	Description
TIM	**2**	**xxxx**	**DGDI timeout** Selection of timeout on waiting for DGDI test status ready (Format: RVVV).
CFG	**1**	**xx**	**Hardware configuration (bitcoded)** <table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>7:4</td><td>**reserved, must be zero**</td></tr><tr><td>3:1</td><td>**Select DCDC feedback signal** **000_B:** normal (VsenseX5 for LS7, VsenseX6 for LS8) **001_B:** twisted (VsenseX6 for LS7, VsenseX5 for LS8) **010_B:** VsenseX1, resonance converter **011_B:** VsenseX2, resonance converter **100_B:** VsenseX3, resonance converter **101_B:** VsenseX4, resonance converter **110_B:** VsenseX5, resonance converter **111_B:** VsenseX6, resonance converter</td></tr><tr><td>0</td><td>**Select DCDC control signal** **0_B:** LS7 **1_B:** LS8</td></tr></table>
CURM	**2**	**xxxx**	**General DGDI current measurement parameters** <table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>15:10</td><td>**not used, must be set to 0**</td></tr><tr><td>9:8</td><td>**Select OAx for current measurement** **00_B:** OA1, OA2, OA3 high impedance **01_B:** OA1 current measurement, OA2, OA3 high impedance **10_B:** OA2 current measurement, OA1, OA3 high impedance</td></tr></table>

Name	Length	Fmt	Description
			Bit(s) / **Description**
			11_B: OA3 current measurement, OA1, OA2 high impedance
			7:6 — **not used, must be set to 0**
			5:4 — **Gain setting of OAx amplifier** **00_B:** 1.33 **01_B:** 2.00 **10_B:** 3.00 **11_B:** 5.33 **Note:** Value is only relevant, if current is measured at OAx pin.
			3:2 — **not used, must be set to 0**
			1:0 — **Select amplification for current measurement** **00_B:** 5.79 **01_B:** 8.68 **10_B:** 12.53 **11_B:** 19.25 **Attention:** This gain is use for current control. Use only gain 12.53 or 19.25
CURL	**2**	**xxxx**	**Current limit parameters for 2-point regulation**
			Bit(s) / **Description**
			15:8 — **DAC value to select max current for 2-point regulation (dac5/6h register)** **Note:** The DAC value for the max current need to be calculated with formular: $DAC\_value = \frac{I_{\max} * G_{DA\_diff} * R_{Shunt} + 250mV}{9{,}77mV}$ **Attention:** Only values below 6 Ampere should be configured (but depends on hardware)!
			7:0 — **not used, must be set to 0**
CURN	**2**	**xxxx**	**Current limit parameters / VDS timeout (in case of resonance converter)**
			Bit(s) / **Description**
			15:8 — **DAC value to select min current for 2-point regulation (dac5/6l register)**

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td></td><td>Note: The DAC value for the min current need to be calculated with formular:
$DAC\_value = \frac{I_{\min} * G_{DA\_diff} * R_{Shunt} + 250mV}{9{,}77mV}$</td></tr>
<tr><td>7:4</td><td>not used, must be set to 0</td></tr>
<tr><td>3:0</td><td>DAC value to select Booster negativ over current (dac5/6neg register).This value limits the current when the Booster is decharged.

Note: The DAC value for the booster negativ over current need to be calculated with formular:
$DAC\_value = \frac{I_{boostover} * -2 * R_{Shunt} + 250mV}{156{,}25mV}$

Note: Handling of parameter is not supported in current Microcode version.</td></tr>
<tr><td>15:0</td><td>VDS timeout (in case of resonance converter). Resolution is 1/6µs/LSB.</td></tr>
</table>
</td></tr>
<tr><td>BOOSTV</td><td>2</td><td>xxxx</td><td>Booster voltage parameters
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>15:8</td><td>not used, must be set to 0</td></tr>
<tr><td>7:0</td><td>DAC value to select Booster Voltage (boostdac register 17Fh)

Note: The DAC value for the booster voltage need to be calculated with formular:
$DAC\_value = \frac{V_{boost}}{313mV}$

Note: Only values between 0x08 (2.5V) and 0xD0 (65V) are allowed</td></tr>
</table>
</td></tr>
<tr><td>TMAX</td><td>2</td><td>xxxx</td><td>Maximum charge time
Maximum time until booster voltage is reached. Resolution is 2µs/LSB</td></tr>
<tr><td>SLEW</td><td>2</td><td>xxxx</td><td>Slew rate setting for LS7/8
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>15:6</td><td>not used, must be set to 0</td></tr>
<tr><td>5:4</td><td>Select value for R DSON of NMOS transistor to configure slew rate for switching off LS7/8. Value in Ohm.</td></tr>
</table>
</td></tr>
</table>

Name	Length	Fmt	Description	
			Bit(s)	**Description**
				00$_B$: 1.1 (1500V/µs) **01$_B$:** 5.9 (300V/µs) **10$_B$:** 35 (50V/µs) **11$_B$:** 69 (25V/µs)
			3:2	**not used, must be set to 0**
			1:0	**Select value for R DSON of PMOS transistor to configure slew rate for switching on LS7/8. Value in Ohm.** **00$_B$:** 5.0 (1500V/µs) **01$_B$:** 14.6 (300V/µs) **10$_B$:** 85 (50V/µs) **11$_B$:** 170 (25V/µs)

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D1	xx	xx	01	...

Parameters

Error response of command "DCDC converter setup"

...	+0
(1*)	**ERROR**

Details

Error response of command "DCDC converter setup"

Name	Length	Fmt	Description	
ERROR	**2**	**xxxx**	**Error Status (bitcoded)**	
			Bit(s)	**Description**
			15	**1: Unexpected driver-disable-Interrupt channel 3**
			14	**1: Unexpected diagnosis-Interrupt channel 3**
			13	**1: Unexpected SW-Interrupt channel 3**
			12	**1: Unimplemented test requested channel 3**
			11	**1: Unexpected driver-disable-Interrupt channel 2**
			10	**1: Unexpected diagnosis-Interrupt channel 2**

Name	Length	Fmt	Description	
			Bit(s)	**Description**
			9	**1: Unexpected SW-Interrupt channel 2**
			8	**1: Unimplemented test requested channel 2**
			7	**1: Unexpected driver-disable-Interrupt channel 1**
			6	**1: Unexpected diagnosis-Interrupt channel 1**
			5	**1: Unexpected SW-Interrupt channel 1**
			4	**1: Unimplemented test requested channel 1**
			3	**1: DCDC-parameter CRC failed**
			2	**1: DGDI has loss of external clock**
			1	**1: Microcode has checksum error**
			0	**1: Microcode is not ready**

.

51 - DCDC converter on/off

This command is used to switch the DCDC converter on or off.

Table 9. Initialisation values of DGDI register written by command "DCDC converter switching"

Register	Address	Init Value
DRAM15/DRAM31 Ch3	0x08F/0x09F	taken from command parameter
ctrl_reg_seq0/1	0x141/0x142	Bit 7 set to start Microcode

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D1	51	01	...

Parameters

parameters to select a SPI controller to the device

5	+0	+1	+2	+3
(1*)	**POMT**	**PIMT**	**POMR**	**PIMR**

...	+4	+5	+6	+7
(1*)	**POCK**	**PICK**	**POSL**	**PISL**

para. for DCDC conv on/off

13	+0	+2	+3
(1*)	**TIM**	**CFG**	**DCDC**

Details

parameters to select a SPI controller to the device
Port and Pin parameters to select the SPI controller where the device is connected

Name	Length	Fmt	Description
POMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit Slave Receive) signal
PIMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within POMT of MTSR (Master Transmit Slave Receive) (SI) signal
POMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive Slave Transmit) signal
PIMR	**1**	**xx**	**Pin number of MRST signal** Pin number within POMR of MRST (Master Receive Slave Transmit) (SO) signal
POCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PICK	**1**	**xx**	**Pin number of SCK signal** Pin number within POCK of SCK (Serial Clock) signal
POSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PISL	**1**	**xx**	**Pin number of SLSO signal** Pin number within POSL of SLSO (Slave Select) signal

para. for DCDC conv on/off
parameters for DCDC converter switching

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>TIM</td><td>2</td><td>xxxx</td><td>DGDI timeout
Selection of timeout on waiting for DGDI test status ready (Format: RVVV).</td></tr>
<tr><td>CFG</td><td>1</td><td>xx</td><td>Hardware configuration (bitcoded)
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:1</td><td>reserved, must be zero</td></tr>
<tr><td>0</td><td>Select DCDC control signal
0_B: LS7
1_B: LS8</td></tr>
</table></td></tr>
<tr><td>DCDC</td><td>2</td><td>xxxx</td><td>Switching DCDC converter
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>15:1</td><td>not used, must be set to 0</td></tr>
<tr><td>0</td><td>Select status of DCDC converter</td></tr>
</table></td></tr>
</table>

Name	Length	Fmt	Description	
			Bit(s)	**Description**
				0_B: Off **1_B:** On

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D1	xx	xx	01	...

Parameters

Error response of command "DCDC converter switching"

6	+0	+2
(1*)	**ERROR**	**STAT**

response of command "DCDC converter on"

...	+0
(1*)	**TIME**

Details

Error response of command "DCDC converter switching"

Name	Length	Fmt	Description	
ERROR	**2**	**xxxx**	**Error Status DGDI (bitcoded)**	
			Bit(s)	**Description**
			15	**1: Unexpected driver-disable-Interrupt channel 3**
			14	**1: Unexpected diagnosis-Interrupt channel 3**
			13	**1: Unexpected SW-Interrupt channel 3**
			12	**1: Unimplemented test requested channel 3**
			11	**1: Unexpected driver-disable-Interrupt channel 2**
			10	**1: Unexpected diagnosis-Interrupt channel 2**
			9	**1: Unexpected SW-Interrupt channel 2**
			8	**1: Unimplemented test requested channel 2**
			7	**1: Unexpected driver-disable-Interrupt channel 1**
			6	**1: Unexpected diagnosis-Interrupt channel 1**

Name	Length	Fmt	Description
			Bit(s) / **Description**
			5 — **1: Unexpected SW-Interrupt channel 1**
			4 — **1: Unimplemented test requested channel 1**
			3 — **1: DCDC-parameter CRC failed**
			2 — **1: DGDI has loss of external clock**
			1 — **1: Microcode has checksum error**
			0 — **1: Microcode is not ready**
STAT	**2**	**xxxx**	**Status of test responsed from microcode (bitcoded)**
			Bit(s) / **Description**
			15:2 — **reserved, set to zero**
			1 — **1: DCDC-parameter CRC failed**
			0 — **1: Booster voltage is not reached within Tmax**

response of command "DCDC converter on"

Name	Length	Fmt	Description
TIME	**2**	**xxxx**	**DCDC startup time. Resolution is 2µs/LSB**

.

52 - DCDC count current feedback signal

This command is used to count the DCDC current feedback signal cur56l_fbk changes. This command is used in case of two ASIC's and coupled DCDC's. Prior the usage an offset compensation with command 0x40 should take place. Also prior the DCDC should be configured with command 0x50 (DCDC converter setup) with the same values as the other DCDC but the value for the dac5/6l register (187h/189h) and dac5/6h (187h/189h) set to the mean value of dac5/6l (187h/189h) and dac5/6h (187h/189h) of the other DCDC.

Table 10. Initialisation values of DGDI register written by command "DCDC count current feedback signal"

Register	Address	Init Value
DRAM04/DRAM20 Ch3	0x084/0x094	taken from command parameter
ctrl_reg_seq0/1	0x141/0x142	Bit 6 set to start Microcode

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D1	52	01	...

Parameters

parameters to select a SPI controller to the device

5	+0	+1	+2	+3
(1*)	**POMT**	**PIMT**	**POMR**	**PIMR**

...	+4	+5	+6	+7
(1*)	**POCK**	**PICK**	**POSL**	**PISL**

para. for DCDC count current feedback signal

13	+0	+2	+3
(1*)	**TIM**	**CFG**	**MWT**

Details

parameters to select a SPI controller to the device
Port and Pin parameters to select the SPI controller where the device is connected

Name	Length	Fmt	Description
POMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit Slave Receive) signal
PIMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within POMT of MTSR (Master Transmit Slave Receive) (SI) signal
POMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive Slave Transmit) signal
PIMR	**1**	**xx**	**Pin number of MRST signal** Pin number within POMR of MRST (Master Receive Slave Transmit) (SO) signal
POCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PICK	**1**	**xx**	**Pin number of SCK signal** Pin number within POCK of SCK (Serial Clock) signal
POSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PISL	**1**	**xx**	**Pin number of SLSO signal** Pin number within POSL of SLSO (Slave Select) signal

para. for DCDC count current feedback signal
parameters for DCDC count current feedback signal

Name	Length	Fmt	Description
TIM	**2**	**xxxx**	**DGDI timeout** Selection of timeout on waiting for DGDI test status ready (Format: RVVV).
CFG	**1**	**xx**	**Hardware configuration (bitcoded)** Selection of the dedicated feedback signal. **Bit(s)** / **Description** **7:1** / **reserved, must be zero**

Name	Length	Fmt	Description
			Bit(s): 0 — **Select DCDC control signal** 0_B: LS7 1_B: LS8
MWT	**2**	**xxxx**	**Measurement window time** Time window where the edge counting take place. Resolution is 2µs/LSB

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D1	xx	xx	01	...

Parameters

Error response of command "DCDC count current feedback signal"

6	+0	+2	+4	+6
(1*)	**ERROR**	**STAT**	**CNTL**	**CNTH**

Details

Error response of command "DCDC count current feedback signal"

Name	Length	Fmt	Description
ERROR	**2**	**xxxx**	**Error Status DGDI (bitcoded)**

Bit(s)	Description
15	**1: Unexpected driver-disable-Interrupt channel 3**
14	**1: Unexpected diagnosis-Interrupt channel 3**
13	**1: Unexpected SW-Interrupt channel 3**
12	**1: Unimplemented test requested channel 3**
11	**1: Unexpected driver-disable-Interrupt channel 2**
10	**1: Unexpected diagnosis-Interrupt channel 2**
9	**1: Unexpected SW-Interrupt channel 2**
8	**1: Unimplemented test requested channel 2**
7	**1: Unexpected driver-disable-Interrupt channel 1**
6	**1: Unexpected diagnosis-Interrupt channel 1**

Name	Length	Fmt	Description
			Bit(s) / **Description**
			5 — **1: Unexpected SW-Interrupt channel 1**
			4 — **1: Unimplemented test requested channel 1**
			3 — **1: DCDC-parameter CRC failed**
			2 — **1: DGDI has loss of external clock**
			1 — **1: Microcode has checksum error**
			0 — **1: Microcode is not ready**
STAT	**2**	**xxxx**	**Status of test responsed from microcode (bitcoded)** **Bit(s)** / **Description** **15:0** — **reserved, set to zero**
CNTL	**2**	**xxxx**	**High to low edge counter**
CNTH	**2**	**xxxx**	**Low to high edge counter**

.

C0 - VCCP regulator off/on cycling

This command is used to trigger a VCCP regulator off/on cycling. This test is part of the extended device screening.

NOTE: Ensure that hardware use no external VCCP regulator!

NOTE: Ensure that all bootstrap capacitors are charged or disabled (see bootstrap_charged register 0x18D, bit 6-0)!

Table 11. Initialisation values of DGDI register written by command "VCCP regulator off/on cycling"

Register	Address	Init Value
driver_config_part2	0x01A6	Bit 15 toggled due to command parameters

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D1	C0	01	...

Parameters

parameters to select a SPI controller for DGDI

5	+0	+1	+2	+3
(1*)	**POMT**	**PIMT**	**POMR**	**PIMR**

...	+4	+5	+6	+7
(1*)	**POCK**	**PICK**	**POSL**	**PISL**

parameters for VCCP regulator off/on cycling

13	+0	+2	+4	+6
(1*)	**TIM**	**T_OFF**	**T_ON**	**CYCLES**

Details

parameters to select a SPI controller for DGDI
Port and Pin parameters to select the SPI controller where the device is connected

Name	Length	Fmt	Description
POMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit Slave Receive) signal
PIMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within POMT of MTSR (Master Transmit Slave Receive) (SI) signal
POMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive Slave Transmit) signal
PIMR	**1**	**xx**	**Pin number of MRST signal** Pin number within POMR of MRST (Master Receive Slave Transmit) (SO) signal
POCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PICK	**1**	**xx**	**Pin number of SCK signal** Pin number within POCK of SCK (Serial Clock) signal
POSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PISL	**1**	**xx**	**Pin number of SLSO signal** Pin number within POSL of SLSO (Slave Select) signal

parameters for VCCP regulator off/on cycling
configuration parameters for VCCP regulator off/on cycling

Name	Length	Fmt	Description
TIM	**2**	**xxxx**	**DGDI timeout** Selection of timeout on waiting for DGDI test status ready. Format: RVVV. (Currently not used.)
T_OFF	**2**	**xxxx**	**Off time** Time the regulator is switched off. Format: RVVV.
T_ON	**2**	**xxxx**	**On time** Time the regulator is switched on. Format: RVVV.
CYCLES	**1**	**xx**	**Cycles** Number of cycles

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D1	xx	xx	01	...

Parameters

response of command "VCCP regulator off/on cycling"

...	+0
(1*)	**DRVSTAT**

Details

response of command "VCCP regulator off/on cycling"

Name	Length	Fmt	Description
DRVSTAT	**1**	**xx**	**Uv vccp of driver_status register** **Bit(s)** / **Description** **7:1** — **reserved, set to 0** **0** — **bit 0 of register 0x1B2**

.

C1 - DCDC ramp up with variable slew rate settings

This command is used to start the DC/DC converter with different slew rate settings. It is nearly the same as command 0x51 (DCDC converter on/off) but uses pairs of booster voltage (BOOSTV) and used slew rate (SLEW). This test is part of the extended device screening.

NOTE: This command ignores the parameters BOOSTV and SLEW from command 0x50.

NOTE: Because of the shared register boost_dac (17Fh) two DCDC converter on one device are not independent.

Table 12. Initialisation values of DGDI register written by command "DCDC ramp up with variable slew rate settings"

Register	Address	Init Value
ls_slewrate_part2	0x173	taken from command parameter
boost_dac	0x17F	taken from command parameter
DRAM15/DRAM31 Ch3	0x08F/0x09F	taken from command parameter
ctrl_reg_seq0/1	0x141/0x142	Bit 7 set to start Microcode

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D1	C1	01	...

Parameters

parameters to select a SPI controller for DGDI

5	+0	+1	+2	+3
(1*)	**POMT**	**PIMT**	**POMR**	**PIMR**

...	+4	+5	+6	+7
(1*)	**POCK**	**PICK**	**POSL**	**PISL**

parameters for DCDC ramp up with variable slew rate settings

13	+0	+2
(1*)	**TIM**	**CFG**

additional parameters for command

16	+0	+1
(1..n*)	**BOOSTV**	**SLEW**

Details

parameters to select a SPI controller for DGDI
Port and Pin parameters to select the SPI controller where the device is connected

Name	Length	Fmt	Description
POMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit Slave Receive) signal
PIMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within POMT of MTSR (Master Transmit Slave Receive) (SI) signal
POMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive Slave Transmit) signal
PIMR	**1**	**xx**	**Pin number of MRST signal** Pin number within POMR of MRST (Master Receive Slave Transmit) (SO) signal
POCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PICK	**1**	**xx**	**Pin number of SCK signal** Pin number within POCK of SCK (Serial Clock) signal
POSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PISL	**1**	**xx**	**Pin number of SLSO signal** Pin number within POSL of SLSO (Slave Select) signal

parameters for DCDC ramp up with variable slew rate settings
configuration parameters for DCDC ramp up with variable slew rate settings

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>TIM</td><td>2</td><td>xxxx</td><td>DGDI timeout
Selection of timeout on waiting for DGDI test status ready. Format: RVVV.</td></tr>
<tr><td>CFG</td><td>1</td><td>xx</td><td>Hardware configuration (bitcoded)
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:1</td><td>reserved, must be zero</td></tr>
</table></td></tr>
</table>

Name	Length	Fmt	Description
			Bit(s) / **Description** **0**: **Select DCDC control signal** **0_B:** LS7 **1_B:** LS8

additional parameters for command

Name	Length	Fmt	Description
BOOSTV	**1**	**xx**	**Booster voltage parameters** **Bit(s)** / **Description** **7:0**: **DAC value to select Booster Voltage (boostdac register 17Fh)** **Note:** The DAC value for the booster voltage need to be calculated with formular: $DAC\_value = \frac{V_{boost}}{313mV}$ **Note:** Only values between 0x08 (2,5V) and 0xD0 (65V) are allowed
SLEW	**1**	**xx**	**Slew rate setting for LS7/8** **Bit(s)** / **Description** **7:6**: **not used, must be set to 0** **5**: **reserved, must be set to 0** **4**: **Select value for R DSON of NMOS transistor to configure slew rate for switching off LS7/8. Value in Ohm.** **0_B:** 1.1 (1500V/µs) **1_B:** 5.9 (300V/µs) **3:2**: **not used, must be set to 0** **1**: **reserved, must be set to 0** **0**: **Select value for R DSON of PMOS transistor to configure slew rate for switching on LS7/8. Value in Ohm.** **0_B:** 5.0 (1500V/µs) **1_B:** 14.6 (300V/µs)

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D1	xx	xx	01	...

Parameters

response of command "DCDC ramp up with variable slew rate settings"

6	+0	+2
(1*)	**ERROR**	**STAT**

additional response of command "DCDC ramp up with variable slew rate settings"

...	+0
(1..n*)	**TIME**

Details

response of command "DCDC ramp up with variable slew rate settings"

Name	Length	Fmt	Description
ERROR	**2**	**xxxx**	**Error Status DGDI (bitcoded)**
			Bit(s) / **Description**
			15: **1: Unexpected driver-disable-Interrupt channel 3**
			14: **1: Unexpected diagnosis-Interrupt channel 3**
			13: **1: Unexpected SW-Interrupt channel 3**
			12: **1: Unimplemented test requested channel 3**
			11: **1: Unexpected driver-disable-Interrupt channel 2**
			10: **1: Unexpected diagnosis-Interrupt channel 2**
			9: **1: Unexpected SW-Interrupt channel 2**
			8: **1: Unimplemented test requested channel 2**
			7: **1: Unexpected driver-disable-Interrupt channel 1**
			6: **1: Unexpected diagnosis-Interrupt channel 1**
			5: **1: Unexpected SW-Interrupt channel 1**
			4: **1: Unimplemented test requested channel 1**
			3: **1: DCDC-parameter CRC failed**
			2: **1: DGDI has loss of external clock**
			1: **1: Microcode has checksum error**
			0: **1: Microcode is not ready**
STAT	**2**	**xxxx**	**Status of test responsed from microcode (bitcoded)**

Name	Length	Fmt	Description
			Bit(s) / **Description**
			15:2 — **reserved, set to zero**
			1 — **1: DCDC-parameter CRC failed**
			0 — **1: Booster voltage is not reached within Tmax**

additional response of command "DCDC ramp up with variable slew rate settings"

Name	Length	Fmt	Description
TIME	**2**	**xxxx**	**Ramp up time:** DCDC ramp-up time. Resolution is 2µs/LSB.

.

C2 - HSx/LSx cycling

This command is used to trigger a HSx or LSx cycling. This test is part of the extended device screening.

Table 13. Initialisation values of DGDI register written by command "HSx/LSx cyclings"

Register	Address	Init Value
cur_access_reg_ch1	0x166	0xFFFF
hs_slewrate	0x171	taken from command parameter
ls_slewrate_part1	0x172	taken from command parameter
hs_bias_config	0x18B	taken from command parameter
ls_bias_config	0x18C	taken from command parameter
DRAM0-DRAM7	0x000-0x007	taken from command parameter
ctrl_reg_seq0	0x101	Bit 10, 11 and 12 set to start Microcode

Note: All currents can be transformed to ADC values in command with following formular:

$$ADC_{value} = \frac{I * G_{DA\_diff} * R_{Shunt} + 250mV}{9{,}77mV}$$

To set G_{DA_diff} use MBLOCK [9:8] (amplification for current measurement).

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D1	C2	01	...

Parameters

parameters to select a SPI controller for DGDI

5	+0	+1	+2	+3
(1*)	**POMT**	**PIMT**	**POMR**	**PIMR**

...	+4	+5	+6	+7
(1*)	**POCK**	**PICK**	**POSL**	**PISL**

parameters for HSx/LSx cycling

13	+0	+2	+3
(1*)	**TIM**	**MODE**	**REGINI**

...	+4	+6	+8
(1*)	**SCFG**	**MBLOCK**	**I_CYCLE**

...	+10	+12	+14
(1*)	**T_WAIT**	**T_DELAY**	**T_CYCLE**

additional parameters for HSx/LSx cycling

...	+0
(1..n*)	**SLEW**

Details

parameters to select a SPI controller for DGDI
Port and Pin parameters to select the SPI controller where the device is connected

Name	Length	Fmt	Description
POMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit Slave Receive) signal
PIMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within POMT of MTSR (Master Transmit Slave Receive) (SI) signal
POMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive Slave Transmit) signal
PIMR	**1**	**xx**	**Pin number of MRST signal** Pin number within POMR of MRST (Master Receive Slave Transmit) (SO) signal
POCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PICK	**1**	**xx**	**Pin number of SCK signal** Pin number within POCK of SCK (Serial Clock) signal
POSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PISL	**1**	**xx**	**Pin number of SLSO signal** Pin number within POSL of SLSO (Slave Select) signal

parameters for HSx/LSx cycling
configuration parameters for HSx/LSx cyclings

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>TIM</td><td>2</td><td>xxxx</td><td>DGDI timeout
Selection of timeout on waiting for DGDI test status ready. Format: RVVV.</td></tr>
<tr><td>MODE</td><td>1</td><td>xx</td><td>Mode:
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:2</td><td>not used, must be set to 0</td></tr>
<tr><td>1:0</td><td>Cycling mode
00_B: HS VBatt cycling
01_B: HS VBoost cycling
10_B: LS cycling, HS on VBatt
11_B: LS cycling, HS on VBoost</td></tr>
</table></td></tr>
<tr><td>REGINI</td><td>1</td><td>xx</td><td>Register setting before test:
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7</td><td>not used, must be set to 0</td></tr>
<tr><td>6</td><td>Bias HS strong setting (hs_bias_config register bit 9, 11)
0_B: Bias off
1_B: Bias on</td></tr>
<tr><td>5</td><td>Bias LS setting (ls_bias_config register bit 0-5)
0_B: Bias off
1_B: Bias on</td></tr>
<tr><td>4</td><td>Bias HS setting (hs_bias_config register bit 0-6)
0_B: Bias off
1_B: Bias on</td></tr>
<tr><td>3:2</td><td>not used, must be set to 0</td></tr>
<tr><td>1:0</td><td>Slewrate for the not cycling switch
00_B: slew rate according to datasheet
01_B: slew rate according to datasheet
10_B: slew rate according to datasheet
11_B: slew rate according to datasheet</td></tr>
</table></td></tr>
<tr><td>SCFG</td><td>2</td><td>xxxx</td><td>Switch configuration</td></tr>
</table>

Name	Length	Fmt	Description	
			Bit(s)	**Description**
			15:5	**not used, must be set to 0**
			4:0	**HSBat, HSBoost and LS** **00_H:** hs1, hs2, ls1 **01_H:** hs1, hs2, ls2 **02_H:** hs1, hs2, ls3 **03_H:** hs1, hs2, ls4 **04_H:** hs1, hs2, ls5 **05_H:** hs1, hs2, ls6 **06_H:** hs3, hs4, ls1 **07_H:** hs3, hs4, ls2 **08_H:** hs3, hs4, ls3 **09_H:** hs3, hs4, ls4 **$0A_H$:** hs3, hs4, ls5 **$0B_H$:** hs3, hs4, ls6 **$0C_H$:** hs5, hs6, ls1 **$0D_H$:** hs5, hs6, ls2 **$0E_H$:** hs5, hs6, ls3 **$0F_H$:** hs5, hs6, ls4 **10_H:** hs5, hs6, ls5 **11_H:** hs5, hs6, ls6 **12_H:** hs7, ---, ls7 **13_H:** hs7, ---, ls8 **14_H:** not used **15_H:** not used **16_H:** not used **17_H:** not used **18_H:** not used **19_H:** not used **$1A_H$:** not used **$1B_H$:** not used **$1C_H$:** not used **$1D_H$:** not used **$1E_H$:** not used **$1F_H$:** not used
MBLOCK	**2**	**xxxx**	**Current measurement block parameters**	
			Bit(s)	**Description**
			15:10	**not used, must be set to 0**
			9:8	**Select amplification for current measurement (G_DA_diff)**

Name	Length	Fmt	Description
			Bit(s) / **Description** (cont.) **00$_B$:** 5.79 **01$_B$:** 8.68 **10$_B$:** 12.53 **11$_B$:** 19.25 **7:3** — **not used, must be set to 0** **2:0** — **Select current measurement channel** **000$_B$:** not used **001$_B$:** Channel 1 **010$_B$:** Channel 2 **011$_B$:** Channel 3 **100$_B$:** Channel 4 **101$_B$:** Channel 5 **110$_B$:** Channel 6
I_CYCLE	**2**	**xxxx**	**Cycle current and/or minimum on time:** Current of cycle (ADC value) **Bit(s)** / **Description** **15:8** — **high regulation point** **7:0** — **low regulation point (only for HS cycling)** **7:0** — **minimum LS on time in 1/6µs (only for LS cycling)**
T_WAIT	**2**	**xxxx**	**Wait time** Time for LS off (only for LS cycling). Resolution is 1/6µs/LSB.
T_DELAY	**2**	**xxxx**	**Delay time** Time between LS on and HS on (HS cycling) or between HS on and LS on (LS cycling). Resolution is 1/6µs/LSB.
T_CYCLE	**2**	**xxxx**	**Cycle time** Time for a cycle with one slew rate setting. Resolution is 1/6µs/LSB.

additional parameters for HSx/LSx cycling

Name	Length	Fmt	Description
SLEW	**1**	**xx**	**Slew rate setting:** **Bit(s)** / **Description** **7:2** — **not used, must be set to 0** **1:0** — **Setting of the slew rate for this cycle** **00$_B$:** slew rate according to datasheet

Name	Length	Fmt	Description
			Bit(s) / Description
			01_B: slew rate according to datasheet
			10_B: slew rate according to datasheet
			11_B: slew rate according to datasheet

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D1	xx	xx	01	...

Parameters

Error response of command "HSx/LSx cycling"

6	+0	+2	+4	+6	+8	+10
(1*)	**ERROR**	**STAT**	**ERR1**	**ERR2**	**ERR3**	**CNUM**

Details

Error response of command "HSx/LSx cycling"

Name	Length	Fmt	Description
ERROR	**2**	**xxxx**	**Error Status DGDI (bitcoded)**

Bit(s)	Description
15	**1: Unexpected driver-disable-Interrupt channel 3**
14	**1: Unexpected diagnosis-Interrupt channel 3**
13	**1: Unexpected SW-Interrupt channel 3**
12	**1: Unimplemented test requested channel 3**
11	**1: Unexpected driver-disable-Interrupt channel 2**
10	**1: Unexpected diagnosis-Interrupt channel 2**
9	**1: Unexpected SW-Interrupt channel 2**
8	**1: Unimplemented test requested channel 2**
7	**1: Unexpected driver-disable-Interrupt channel 1**
6	**1: Unexpected diagnosis-Interrupt channel 1**

Name	Length	Fmt	Description
			Bit(s) / **Description**
			5: 1: Unexpected SW-Interrupt channel 1
			4: 1: Unimplemented test requested channel 1
			3: 1: DCDC-parameter CRC failed
			2: 1: DGDI has loss of external clock
			1: 1: Microcode has checksum error
			0: 1: Microcode is not ready
STAT	**2**	**xxxx**	**Status of test responsed from microcode (bitcoded)** **Bit(s)** / **Description** 15:1: reserved 0: 1: Open circuit detected (no load connected)
ERR1	**2**	**xxxx**	**Content of register err_seq1ch1 part 1 (0x1F0)**
ERR2	**2**	**xxxx**	**Content of register err_seq1ch1 part 2 (0x1F1)**
ERR3	**2**	**xxxx**	**Content of register err_seq1ch1 part 3 (0x1F2)**
CNUM	**1**	**xx**	**Number of started cycles with different slew rates**

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)

Code	Description
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0xF0	**No SPI controller on given ports found.**
0xF2	**DGDI communication failed (fix pattern error)**
0xF3	**DGDI communication failed (word error)**
0xF4	**DGDI communication failed (frame error)**
0xF5	**DGDI communication failed (word error and frame error)**
0xF6	**DGDI communication failed (cksys missing)**
0xF7	**DGDI communication failed (word error and cksys missing)**
0xF8	**DGDI communication failed (frame error and cksys missing)**
0xF9	**DGDI communication failed (word error, frame error and cksys missing)**
0xFA	**confict of DCDC configuration with ADC Channel 5/6**
0xFB	**Microcode not found**

TSW_D5 - Piezo injection with CY373

Test step description

Date: 22.02.2016

Created: 18.01.2017 09:12.

This TSW provides functions for test of injection powerstage for piezo injectors (CY373).

Important: before use, initialize ports, SPI, used adc/dsadc channels and setup clock for CY373.

All physical parameters like period, timeouts etc. use the 16 bit RVVV format, where R (4 bit) is the resolution (see following table) and VVV (12 bit) is the value of the parameter. For example 2ms would be 0xD002 (or 0xC014 or 0xB0C8 or 0xA7D0).

Table 1. Resolution table

Value of R	Resolution
0x0	1.0E+0s = 1s
0x1	1.0E+1s = 10s
0x2	1.0E+2s = 100s
0x3	1.0E+3s = 1000s
...	...
0x8	1.0E-8s = 10ns
0x9	1.0E-7s = 100ns
0xA	1.0E-6s = 1µs
0xB	1.0E-5s = 10µs
0xC	1.0E-4s = 100µs
0xD	1.0E-3s = 1ms
0xE	1.0E-2s = 10ms
0xF	1.0E-1s = 100ms

Table 2. Cylinder mapping table for command 0x10

CYL (Cylinder)	Device	LS driver output GLSxy
01 (0x01)	1	11
02 (0x02)	1	12
03 (0x03)	1	13
04 (0x04)	1	21
05 (0x05)	1	22
06 (0x06)	1	23
07 (0x07)	2	11
08 (0x08)	2	12
09 (0x09)	2	13
10 (0x0A)	2	21

CYL (Cylinder)	Device	LS driver output GLSxy
11 (0x0B)	2	22
12 (0x0C)	2	23

Table 3. Bank mapping table for command 0x10

BNK (Bank)	Device	Bank (1-Bank-Sys)	Bank (2-Bank-Sys)
01 (0x01)	1	1	1
02 (0x02)	1	illegal	2
03 (0x03)	2	1	1
04 (0x04)	2	illegal	2

Table 4. Initialisation values (in hex) for command 0x20

Address	Register	Table 0	Table 1
0x10	PA_CHARGE_B1	0x5D	
0x11	PA_CHG_B1	0x2D	0x48
0x12	PA_TCH_B1	0x13	
0x13	PA_DCH_B1	0x3D	0x62
0x14	PA_TDC_B1	0x0F	
0x16	PA_CHARGE_B2	0x5D	
0x17	PA_CHG_B2	0x2D	0x48
0x18	PA_TCH_B2	0x13	
0x19	PA_DCH_B2	0x3D	0x62
0x1A	PA_TDC_B2	0x0F	
0x1C	PA_UBUFF_B1	0x6C	
0x1D	PA_UBUFF_B2	0x6C	
0x1E	PA_CSLOP_B1	0x00	
0x1F	PA_DSLOP_B1	0x00	
0x20	PA_CSLOP_B2	0x00	
0x21	PA_DSLOP_B2	0x00	
0x2D	PA_T_MIN_ON	0x1F	
0x2E	PA_T_LSE	0x50	
0x2F	PA_T_OFFSET	0x00	
0x30	PA_DCH_ERR	0x38	
0x31	PA_TDC_ERR	0x18	
0x32	PA_T_UT0MIN	0x07	
0x34	PA_DCDC_CLK	0x54	
0x35	PA_T_IDelta	0xC9	
0x37	PA_T_PWM_1	0x3C	
0x38	PA_T_PWM_3	0x1C	
0x39	PA_T_PWM_4	0x16	
0x3A	PA_T_PWM_5	0x14	
0x3B	PA_M_REG	0x20	

Address	Register	Table 0	Table 1
0x3C	PA_T_DCDC_A	0x08	
0x3D	PA_T_DCDC_B	0x1F	
0x3E	PA_T_DCDC_C	0x2A	
0x3F	PA_T_DCDC_D	0x2F	

The table shows the initalisation values which are written to the device by command 0x20. The "Table 1" column shows only these values, which are different to column "Table 0". Normally the Table 0 is used in conjunction with the 20 Milliohm Buffer-Shunt and Table 1 in conjunction with the 33 Milliohm Buffer-Shunt.

Test step commands

.

01 - Define device pins

This command defines four controller pins used for spi communication and eight controller pins used for digital device access. Up to 2 devices could be defined.

Note: The used pins have to be initialized with TSW_04.

Hint: If on one bank systems some io pins of bank 2 are not connected and therefore not used, use the same pins for bank 2 as for bank 1 to avoid erroneous response.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D5	01	01	...

Parameters

parameters to select a SPI controller for the device

5	+0	+1	+2	+3	+4	+5
(1..2*)	**POMT**	**PIMT**	**POMR**	**PIMR**	**POCK**	**PICK**

...	+6	+7
(1..2*)	**POSL**	**PISL**

parameters to select pins for the device access

13	+0	+1	+2	+3	+4	+5
(1..2*)	**PONON1**	**PINON1**	**POSEL10**	**PISEL10**	**POSEL11**	**PISEL11**

...	+6	+7	+8	+9	+10	+11
(1..2*)	**POBSY1**	**PIBSY1**	**PONON2**	**PINON2**	**POSEL20**	**PISEL20**

...	+12	+13	+14	+15
(1..2*)	**POSEL21**	**PISEL21**	**POBSY2**	**PIBSY2**

Details

parameters to select a SPI controller for the device
Port and Pin parameters to select the SPI controller where the device is connected

Name	Length	Fmt	Description
POMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit Slave Receive) signal
PIMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within POMT of MTSR (Master Transmit Slave Receive) (SI) signal
POMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive Slave Transmit) signal
PIMR	**1**	**xx**	**Pin number of MRST signal** Pin number within POMR of MRST (Master Receive Slave Transmit) (SO) signal
POCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PICK	**1**	**xx**	**Pin number of SCK signal** Pin number within POCK of SCK (Serial Clock) signal
POSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PISL	**1**	**xx**	**Pin number of SLSO signal** Pin number within POSL of SLSO (Slave Select) signal

parameters to select pins for the device access
Port and Pin parameters to select the pins for the device access (bank 1 and bank 2)

Name	Length	Fmt	Description
PONON1	**1**	**xx**	**Port number of nON1 signal** Port number of negated ON1 signal
PINON1	**1**	**xx**	**Pin number of nON1 signal** Pin number within POnON1 of negated ON1 signal
POSEL10	**1**	**xx**	**Port number of SEL10 signal** Port number of SEL10 signal
PISEL10	**1**	**xx**	**Pin number of SEL10 signal** Pin number within POSEL10 of SEL10 signal
POSEL11	**1**	**xx**	**Port number of SEL11 signal** Port number of SEL11 signal
PISEL11	**1**	**xx**	**Pin number of SEL11 signal** Pin number within POSEL11 of SEL11 signal
POBSY1	**1**	**xx**	**Port number and TIM of BUSY1 signal** **Bit(s)** / **Description** **7:6** **measurement configuration** **00_B:** HW; First TIM **01_B:** HW; Second TIM

Name	Length	Fmt	Description
			Bit(s) / Description **10_B:** reserved **11_B:** SW **5:0** **Port number of BUSY1 signal**
PIBSY1	**1**	**xx**	**Pin number of BUSY1 signal** Pin number within POBSY1 of BUSY1 signal
PONON2	**1**	**xx**	**Port number of nON2 signal** Port number of negated ON2 signal
PINON2	**1**	**xx**	**Pin number of nON2 signal** Pin number within POnON2 of negated ON2 signal
POSEL20	**1**	**xx**	**Port number of SEL20 signal** Port number of SEL20 signal
PISEL20	**1**	**xx**	**Pin number of SEL20 signal** Pin number within POSEL20 of SEL20 signal
POSEL21	**1**	**xx**	**Port number of SEL21 signal** Port number of SEL21 signal
PISEL21	**1**	**xx**	**Pin number of SEL21 signal** Pin number within POSEL21 of SEL21 signal
POBSY2	**1**	**xx**	**Port number and TIM of BUSY2 signal** **Bit(s) / Description** **7:6** **measurement configuration** **00_B:** HW; First TIM **01_B:** HW; Second TIM **10_B:** reserved **11_B:** SW **5:0** **Port number of BUSY2 signal**
PIBSY2	**1**	**xx**	**Pin number of BUSY2 signal** Pin number within POBSY2 of BUSY2 signal

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D5	xx	xx	01	...

Parameters

Response of define device pins command

6	+0	+1
(1..2*)	**MOD**	**CH**

Details

Response of define device pins command

Name	Length	Fmt	Description
MOD	**1**	**xx**	**Module** used TIM submodule for busy time measurement. 0xFF if no module found.
CH	**1**	**xx**	**Channel** used channel of TIM submodule. 0xFF if no module found.

.

02 - Define adc channels

This command defines the ADC channels for which measurement results will be provided after injection. Description of parameter see start conversion and read result (command 10 of tsw 01)

Note: The used adc channels have to be configured with TSW_01.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D5	02	01	...

Parameters

input pins for adc (repeatable up to eight inputs)

5	+0	+1
(1..n*)	**CHA**	**MUX**

Details

input pins for adc (repeatable up to eight inputs)

Name	Length	Fmt	Description
CHA	**1**	**xx**	**Index of the channel**
MUX	**1**	**xx**	**Channel of the external multiplexer**

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	D5	xx	xx	01

.

03 - Define indication pin

This command .is used to define a pin which is toggled to indicate the begin (high) and end (low) of analog digital conversion during injections. This feature is only used for debugging purposes and is not necessary for normal function.

Note: The used pin have to be initialized with TSW_04.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D5	03	01	...

Parameters

parameters to select pin for the adc indication

5	+0	+1
(1*)	**PORT**	**PIN**

Details

parameters to select pin for the adc indication
Port and Pin parameter to select the pin for the adc indication

Name	Length	Fmt	Description
PORT	**1**	**xx**	**Port number** Port number for indication pin
PIN	**1**	**xx**	**Pin number** Pin number for indication pin within port

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	D5	xx	xx	01

.

04 - Configure ncd function

This command configure the parameter for the ncd (needle closing detection) function. Up to 2 devices could be defined.

Note: The used pins have to be initialized with TSW_04 and the used dsadc channels have to be configured with TSW_07.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D5	04	01	...

Parameters

parameter for all NCD AD measurements

5 (1*)	+0 SAMPLES

parameter for AD measurement and selection pins for the measurement switch access

6	+0	+1	+2	+3	+4	+5
(1..2*)	PIDS1	PONCD1	PINCD1	ALVL1	PIDS2	PONCD2

...	+6	+7
(1..2*)	PINCD2	ALVL2

Details

parameter for all NCD AD measurements

Name	Length	Fmt	Description
SAMPLES	**1**	**xx**	**Number of consecutive NCD AD conversions**

parameter for AD measurement and selection pins for the measurement switch access

Name	Length	Fmt	Description
PIDS1	**1**	**xx**	**DS ADC input pin for bank 1** see TSW_07 Delta sigma ADC for details
PONCD1	**1**	**xx**	**Port number of switch** Port number of ncd switch on bank 1
PINCD1	**1**	**xx**	**Pin number of switch** Pin number within PONCD1 of ncd switch
ALVL1	**1**	**xx**	**Activation level (bank 1)** Activation level for ncd measurement switching **Bit(s)** / **Description** **7:1** / **reserved** **0** / **Level** 0_B: Low level 1_B: High level
PIDS2	**1**	**xx**	**DS ADC input pin for bank 2** see TSW_07 Delta sigma ADC for details
PONCD2	**1**	**xx**	**Port number of switch** Port number of ncd switch on bank 2
PINCD2	**1**	**xx**	**Pin number of switch** Pin number within PONCD2 of ncd switch
ALVL2	**1**	**xx**	**Activation level (bank 2)** Activation level for ncd measurement switching **Bit(s)** / **Description** **7:1** / **reserved** **0** / **Level** 0_B: Low level

Name	Length	Fmt	Description	
			Bit(s)	**Description**
				1_B: High level

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	D5	xx	xx	01

.

10 - Set injection and measurement times

This command defines the injection and measurement times for each cylinder. This is done by a set of events. Each event is processed in the order it appears in the command. The whole set of events is one cycle used in command 0x40. For the mapping of the cylinders see table 2 and for the mapping of the banks see table 3. If NCD function is configured and adc channels are defined in parallel, there is a need to do a new setup of the adc channels between ncd measurement and injection measurement. Therefore a additional time up to 20 microseconds is need and this time should be considered defining the wait times.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D5	10	01	...

Parameters

Parameter set for charge event

5	+0	+1	+3
(1..n*)	**CHARGE**	**WAIT**	**CYL**

Parameter set for discharge event

9	+0	+1	+3
(1..n*)	**DISCHRG**	**WAIT**	**CYL**

Parameter set for NCD on event

13	+0	+1	+3
(1..n*)	**NCDON**	**WAIT**	**BNK**

Parameter set for NCD off event

17	+0	+1	+3
(1..n*)	**NCDOFF**	**WAIT**	**BNK**

Parameter set for start ADC event

21	+0	+1
(1..n*)	**START**	**WAIT**

Parameter set for read ADC result event

24	+0	+1
(1..n*)	**READADC**	**WAIT**

Parameter set for Read NCD ADC event

27	+0	+1	+3
(1..n*)	**NCD_ADC**	**WAIT**	**BNK**

Parameter set for disable WD interrupt

31	+0	+1
(1..n*)	**DIS_INT**	**WAIT**

Parameter set for enable WD interrupt

34	+0	+1
(1..n*)	**ENB_INT**	**WAIT**

Parameter set for reset event table

37	+0
(1*)	**RSTTABL**

Details

Parameter set for charge event
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
CHARGE	**1**	**xx**	**Event type** In this case "0x11" for CHARGE
WAIT	**2**	**xxxx**	**Wait time** Wait time before the start of the charge event. Format: RVVV (see table 1)
CYL	**1**	**xx**	**cylinder number** cylinder number (1-12) for the charge event

Parameter set for discharge event
Multiple parameter sets may be appended to create a complete sequence of events

Note: Discharge is only possible, if the cylinder is charged before.

Name	Length	Fmt	Description
DISCHRG	**1**	**xx**	**Event type** In this case "0x10" for DISCHARGE
WAIT	**2**	**xxxx**	**Wait time** Wait time before the start of the discharge event. Format: RVVV (see table 1)
CYL	**1**	**xx**	**cylinder number** cylinder number (1-12) for the discharge event

Parameter set for NCD on event
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
NCDON	**1**	**xx**	**Event type** In this case "0x21" for NCDON
WAIT	**2**	**xxxx**	**Wait time** Wait time before the start of the ncd on event. Format: RVVV (see table 1)
BNK	**1**	**xx**	**bank number** bank number (1-4) for the ncd on event

Parameter set for NCD off event
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
NCDOFF	**1**	**xx**	**Event type** In this case "0x20" for NCDOFF
WAIT	**2**	**xxxx**	**Wait time** Wait time before the start of the ncd off event. Format: RVVV (see table 1)
BNK	**1**	**xx**	**bank number** bank number (1-4) for the ncd off event

Parameter set for start ADC event
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
START	**1**	**xx**	**Event type** In this case "0x81" for START ADC
WAIT	**2**	**xxxx**	**Wait time** Wait time before the start of the analog digital conversion event. Format: RVVV (see table 1)

Parameter set for read ADC result event
Multiple parameter sets may be appended to create a complete sequence of events

Note: If the conversions are not finished, this event waits until they are.

Name	Length	Fmt	Description
READADC	**1**	**xx**	**Event type** In this case "0x80" for READ ADC
WAIT	**2**	**xxxx**	**Wait time** Wait time before the read of the analog digital conversion result. Format: RVVV (see table 1)

Parameter set for Read NCD ADC event
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
NCD_ADC	**1**	**xx**	**Event type** In this case "0xA0" for READ NCD ADC

Name	Length	Fmt	Description
WAIT	**2**	**xxxx**	**Wait time** Wait time before the start of the ncd adc read event. Format: RVVV (see table 1)
BNK	**1**	**xx**	**bank number** bank number (1-4) for the ncd adc read event

Parameter set for disable WD interrupt
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
DIS_INT	**1**	**xx**	**Event type** In this case "0xF0" for DISABLE INT
WAIT	**2**	**xxxx**	**Wait time** Wait time before the start of the interrupt disable event. Format: RVVV (see table 1)

Parameter set for enable WD interrupt
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
ENB_INT	**1**	**xx**	**Event type** In this case "0xF1" for ENABLE INT
WAIT	**2**	**xxxx**	**Wait time** Wait time before the start of the interrupt enable event. Format: RVVV (see table 1)

Parameter set for reset event table

Name	Length	Fmt	Description
RSTTABL	**1**	**xx**	**Event type** In this case "0xFF" for RESET TABLE

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D5	xx	xx	01	...

Parameters

Response of set injection and measurement times command

6	+0
(1*)	**EVENTS**

Details

Response of set injection and measurement times command

Name	Length	Fmt	Description
EVENTS	**1**	**xx**	**Number of defined events**

Name	Length	Fmt	Description
			Number of totally defined events

.

20 - Init device (device 1)

This command is used to initialise the CY373. For the initialisation values see table 4.

Note: Before the initialiation a RESET_OFF command is sent to the device. After the initialisation, the config register is set according to the parameter of the command and with the init bit set.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D5	20	01	...

Parameters

parameters to configure the device

5	+0	+1	+2	+3
(1*)	**BANKS**	**DCDCS**	**DIATH**	**INI**

Details

parameters to configure the device

Name	Length	Fmt	Description
BANKS	**1**	**xx**	**Bank mode** Bank mode of the device <table><tr><th>Value</th><th>Mode</th></tr><tr><td>0x00</td><td>1-Bank-System</td></tr><tr><td>0x01</td><td>2-Bank-System</td></tr></table>
DCDCS	**1**	**xx**	**Number of DC/DCs** Number of DC/DCs of the device <table><tr><th>Value</th><th>Mode</th></tr><tr><td>0x00</td><td>1 DC/DC (DC/DC #1)</td></tr><tr><td>0x01</td><td>2 DC/DC</td></tr></table>
DIATH	**1**	**xx**	**Diag threshold** Diagnose threshold <table><tr><th>Value</th><th>Mode</th></tr><tr><td>0x00</td><td>Buffer-Shunt = 20m</td></tr><tr><td>0x01</td><td>Buffer-Shunt = 33m</td></tr></table>
INI	**1**	**xx**	**Used table** Used table for the initialisation for the device

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td><table><tr><th>Value</th><th>Table</th></tr><tr><td>0x00</td><td>Table 0 (20mOhm)</td></tr><tr><td>0x01</td><td>Table 1 (33mOhm)</td></tr></table></td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D5	xx	xx	01	...

Parameters

Response of init device command

6	+0	+1
(1*)	**CHIPID**	**CHIPREV**

Details

Response of init device command

Name	Length	Fmt	Description
CHIPID	**1**	**xx**	**Chip identification** Chip identification from ID_REG (0x6E for CY373)
CHIPREV	**1**	**xx**	**Chip revision** Chip revision from REV_REG

.

21 - Init device (device 2)

This command is used to initialise the CY373. It is used for the second configured device. Description see test step command 0x20.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	D5	21	01

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	D5	xx	xx	01

.

30 - Start DC/DC (device 1)

This command starts the DC/DC converters.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D5	30	01	...

Parameters

parameters for command

5	+0
(1*)	**TIMEOUT**

Details

parameters for command

Name	Length	Fmt	Description
TIMEOUT	**2**	**xxxx**	**Buffer charge timeout** If the charging procedure exceeds this time, it will be aborted by transmitting the SPI command DC_OFF. A zero value means no supervision. Format: RVVV where R=Resolution, V=Value [s] (see table 1)

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D5	xx	xx	01	...

Parameters

Response of command

6	+0	+2
(1*)	**TI0**	**TI1**

Details

Response of command
.

Name	Length	Fmt	Description
TI0	**2**	**xxxx**	**Startup time of DCDC0** Format: RVVV where R=Resolution, V=Value [s] (see table 1)
TI1	**2**	**xxxx**	**Startup time of DCDC1** Format: RVVV where R=Resolution, V=Value [s] (see table 1)

.

31 - Start DC/DC (device 2)

This command starts the DC/DC converters. It is used for the second configured device. Description see test step command 0x30.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	D5	31	01

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	D5	xx	xx	01

.

40 - Start injection cycles

This command starts the injection cycles which has to be configured by command 0x10.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D5	40	01	...

Parameters

parameters for command

5	+0
(1*)	**CYCLES**

Details

parameters for command

Name	Length	Fmt	Description
CYCLES	**4**	**xxxxxxxx**	**Number of injection cycles** This parameter sets the number of subsequent injection cycles. A zero value means endless.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D5	xx	xx	01	...

Parameters

Response of command for bank 1 (if CYCLES > 0)

6	+0	+1	+2	+4
(1*)	DIAG1_P	DIAG1_A	BSY1MIN	BSY1MAX

Response of command for bank 2 (if CYCLES > 0)

12	+0	+1	+2	+4
(1*)	DIAG2_P	DIAG2_A	BSY2MIN	BSY2MAX

Response of command for bank 1 for second device (if configured and CYCLES > 0)

18	+0	+1	+2	+4
(1*)	DIAG1_P	DIAG1_A	BSY1MIN	BSY1MAX

Response of command for bank 2 for second device (if configured and CYCLES > 0)

24	+0	+1	+2	+4
(1*)	DIAG2_P	DIAG2_A	BSY2MIN	BSY2MAX

repeatable ADC measurement results (if CYCLES > 0)

...	+0
(0..n*)	ADV

Details

Response of command for bank 1 (if CYCLES > 0)

Name	Length	Fmt	Description
DIAG1_P	1	xx	Diagnostic register of bank 1 before injection
DIAG1_A	1	xx	Diagnostic register of bank 1 after injection
BSY1MIN	2	xxxx	Busy time minimum of bank 1 (Format: RVVV)
BSY1MAX	2	xxxx	Busy time maximum of bank 1 (Format: RVVV)

Response of command for bank 2 (if CYCLES > 0)

Name	Length	Fmt	Description
DIAG2_P	1	xx	Diagnostic register of bank 2 before injection
DIAG2_A	1	xx	Diagnostic register of bank 2 after injection
BSY2MIN	2	xxxx	Busy time minimum of bank 2 (Format: RVVV)
BSY2MAX	2	xxxx	Busy time maximum of bank 2 (Format: RVVV)

Response of command for bank 1 for second device (if configured and CYCLES > 0)

Name	Length	Fmt	Description
DIAG1_P	1	xx	Diagnostic register of bank 1 before injection
DIAG1_A	1	xx	Diagnostic register of bank 1 after injection
BSY1MIN	2	xxxx	Busy time minimum of bank 1 (Format: RVVV)
BSY1MAX	2	xxxx	Busy time maximum of bank 1 (Format: RVVV)

Response of command for bank 2 for second device (if configured and CYCLES > 0)

Name	Length	Fmt	Description
DIAG2_P	1	xx	**Diagnostic register of bank 2 before injection**
DIAG2_A	1	xx	**Diagnostic register of bank 2 after injection**
BSY2MIN	2	xxxx	**Busy time minimum of bank 2 (Format: RVVV)**
BSY2MAX	2	xxxx	**Busy time maximum of bank 2 (Format: RVVV)**

repeatable ADC measurement results (if CYCLES > 0)
For each ReadADCEvent all configured adc channels are reported in the order they appear in the define adc channels command (0x02). This is repeated for all ReadADCEvent. After that, if NCD is configured, all measurements from Read NCD ADC event are reported.

Name	Length	Fmt	Description
ADV	2	xxxx	**ADC measurement result value (if configured)**

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)

Code	Description
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0xF0	**No SPI controller on given ports found.**
0xF1	**HAL parameter error**
0xF2	**SPI communication/protocol failed**
0xF3	**SPI communication/protocol failed (FIX)**
0xF4	**SPI communication/protocol failed (COL)**
0xF5	**SPI communication/protocol failed (CFG)**
0xF6	**some defines missing**
0xF7	**no more events possible**
0xF8	**no more resoures available**
0xF9	**invalid configuration**

TSW_D6 - Piezo injection with CY374

Test step description

Date: 14.12.2016

Created: 18.01.2017 09:12.

This TSW provides functions for test of injection powerstage for piezo injectors (CY374).

Important: before use, initialize ports, SPI, used adc/dsadc channels and setup clock for CY374.

Note: See also TSW_DF.

All physical parameters like period, timeouts etc. use the 16 bit RVVV format, where R (4 bit) is the resolution (see following table) and VVV (12 bit) is the value of the parameter. For example 2ms would be 0xD002 (or 0xC014 or 0xB0C8 or 0xA7D0).

Table 1. Resolution table

Value of R	Resolution
0x0	1.0E+0s = 1s
0x1	1.0E+1s = 10s
0x2	1.0E+2s = 100s
0x3	1.0E+3s = 1000s
0x4	1.0E+4s = 10000s
0x5	1.0E+5s = 100000s
0x6	1.0E+6s = 1000000s
0x7	1.0E-9s = 1ns
0x8	1.0E-8s = 10ns
0x9	1.0E-7s = 100ns
0xA	1.0E-6s = 1µs
0xB	1.0E-5s = 10µs
0xC	1.0E-4s = 100µs
0xD	1.0E-3s = 1ms
0xE	1.0E-2s = 10ms
0xF	1.0E-1s = 100ms

Table 2. Initialisation values for command 0x20/0x21

Address (R/W)	Register	Value (dez)	Physical Value
0x50/0x51	PA_CHG_1_B1	79	5.61A
0x52/0x53	PA_CMIN_OFF_1_B1	6	960ns
0x54/0x55	PA_CMIN_ON_1_B1	6	960ns
0x56/0x57	PA_CHGB_1_B1	31	2.20A
0x58/0x59	PA_CSLOPE_1A_B1	0	0
0x5A/0x5B	PA_SLOPE_B_B1	0	0
0x5C/0x5D	PA_CHARGE_1_B1	125	20µs

Address (R/W)	Register	Value (dez)	Physical Value
0x5E/0x5F	PA_CHG_2_B1	0	0.00A
0x60/0x61	PA_CMIN_OFF_2_B1	0	0
0x62/0x63	PA_CMIN_ON_2_B1	0	0
0x64/0x65	PA_CHGB_2_B1	0	0.00A
0x66/0x67	PA_CSLOPE_2A_B1	0	0
0x68/0x69	PA_CHARGE_2_B1	125	20µs
0x6A/0x6B	PA_DCH_1_B1	64	-4.54A
0x6C/0x6D	PA_DMIN_OFF_1_B1	6	960ns
0x6E/0x6F	PA_DMIN_ON_1_B1	6	960ns
0x70/0x71	PA_DCHB_1_B1	31	2.20A
0x72/0x73	PA_DSLOPE_1_B1	0	0
0x74/0x75	PA_DISCHARGE_1_B1	125	20µs
0x76/0x77	PA_DCH_2_B1	0	-0.00A
0x78/0x79	PA_DMIN_OFF_2_B1	25	-
0x7A/0x7B	PA_DMIN_ON_2_B1	25	-
0x7C/0x7D	PA_DCHB_2_B1	0	0.00A
0x7E/0x7F	PA_DSLOPE_2_B1	0	0
0x80/0x81	PA_CHG_1_B2	80	5.68A
0x82/0x83	PA_CMIN_OFF_1_B2	25	-
0x84/0x85	PA_CMIN_ON_1_B2	0	-
0x86/0x87	PA_CHGB_1_B2	31	2.20A
0x88/0x89	PA_CSLOPE_1A_B2	0	0
0x8A/0x8B	PA_SLOPE_B_B2	0	0
0x8C/0x8D	PA_CHARGE_1_B2	125	20µs
0x8E/0x8F	PA_CHG_2_B2	0	0.00A
0x90/0x91	PA_CMIN_OFF_2_B2	53	-
0x92/0x93	PA_CMIN_ON_2_B2	53	-
0x94/0x95	PA_CHGB_2_B2	0	0.00A
0x96/0x97	PA_CSLOPE_2A_B2	0	0
0x98/0x99	PA_CHARGE_2_B2	125	20µs
0x9A/0x9B	PA_DCH_1_B2	80	-5.68A
0x9C/0x9D	PA_DMIN_OFF_1_B2	10	-
0x9E/0x9F	PA_DMIN_ON_1_B2	121	-
0xA0/0xA1	PA_DCHB_1_B2	31	2.20A
0xA2/0xA3	PA_DSLOPE_1_B2	0	0
0xA4/0xA5	PA_DISCHARGE_1_B2	125	20µs
0xA6/0xA7	PA_DCH_2_B2	0	-0.00A
0xA8/0xA9	PA_DMIN_OFF_2_B2	18	-
0xAA/0xAB	PA_DMIN_ON_2_B2	126	-
0xAC/0xAD	PA_DCHB_2_B2	0	0.00A
0xAE/0xAF	PA_DSLOPE_2_B2	0	0

Address (R/W)	Register	Value (dez)	Physical Value
0xB0/0xB1	PA_T_CYL_TIMING	0	0µs
0xB2/0xB3	PA_T_EDGE_B1	255	not active
0xB4/0xB5	PA_T_EDGE_B2	255	not active
0xB6/0xB7	PA_LIMIT_LOW	12	0.85A
0xB8/0xB9	PA_LIMIT_HIGH	255	18.11A
0xBA/0xBB	PA_FAK_I_BAND	200	200
0xBC/0xBD	PA_FAK_HSL-GK_H	53	-
0xBE/0xBF	PA_FAK_HSL-GK_L	36	-
0xC0/0xC1	PA_FAK_LSE-GK_H	44	-
0xC2/0xC3	PA_FAK_LSE-GK_L	30	-
0xC4/0xC5	PA_FAK_WECHSEL	25	-
0xC6/0xC7	PA_FAK_MIN_OFF_1	37	-
0xC8/0xC9	PA_FAK_MIN_OFF_2	16	-
0xCA/0xCB	PA_FAK_MIN_OFF_3	63	-
0xCC/0xCD	PA_T_MIN_OFF_OFFSET	122	-
0xCE/0xCF	PA_FAK_UTEILER	144	-
0xD0/0xD1	PA_DCH_ERR	92	6.53A
0xD2/0xD3	PA_TDC_ERR	35	5.60µs
0xD4/0xD5	PA_T_STOP	16	2.56µs
0xD6/0xD7	PA_T_GLS	2	1.28µs
0xD8/0xD9	PA_T_UT0min	56	8.96µs
0xDA/0xDB	PA_U_UMIN	10	10.0V
0xDC/0xDD	PA_U_UDISCH	30	30.0V
0xDE/0xDF	PA_U_UT0MIN	60	60.0V
0xE0/0xE1	PA_U_UT0MAX	30	30.0V
0xE2/0xE3	PA_I_IDelta	169	IDelta1: 7.27A IDelta2: 6.55A
0xE4/0xE5	PA_I_ILSE	103	7.31A
0xE6/0xE7	PA_T_IPIEZO_MAX	10	1.6µs
0xE8/0xE9	PA_UBAT_TH	126	17.55V
0xEA/0xEB	PA_UBUFF_1	26	29.4V
0xEC/0xED	PA_UBUFF_2	56	63.3V
0xEE/0xEF	PA_UBUFF_3	89	100.6V
0xF0/0xF1	PA_UBUFF	230	259.9V
0xF2/0xF3	PA_DCDC_CLK	122	102.46kHz
0xF4/0xF5	PA_DCDC_PARA	92	14.6A
0xF6/0xF7	PA_T_PWM_ON_MAX	169	6.76µs
0xF8/0xF9	PA_T_DCDC_A	25	1.00µs
0xFA/0xFB	PA_T_DCDC_B	58	2.32µs
0xFC/0xFD	PA_T_DCDC_C	106	4.24µs
0xFE/0xFF	PA_T_DCDC_D	114	4.56µs

The table shows the initalisation values which are written to the device by command 0x20.

Test step commands

.

01 - Define device pins

This command defines four controller pins used for spi communication and four controller pins used for digital device access. Up to 2 devices could be defined.

Note: The used pins have to be initialized with TSW_04.

Hint: If on one bank systems some io pins of bank 2 are not connected and therefore not used, use the same pins for bank 2 as for bank 1 to avoid erroneous response.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D6	01	01	...

Parameters

parameters to select a SPI controller for the device

5 (1..2*)	+0	+1	+2	+3
	POMT	**PIMT**	**POMR**	**PIMR**

... (1..2*)	+4	+5	+6	+7
	POCK	**PICK**	**POSL**	**PISL**

parameters to select pins for the device access

13 (1..2*)	+0	+1	+2	+3
	PO_NON1	**PI_NON1**	**PO_BSY1**	**PI_BSY1**

... (1..2*)	+4	+5	+6	+7
	PO_NON2	**PI_NON2**	**PO_BSY2**	**PI_BSY2**

Details

parameters to select a SPI controller for the device
Port and Pin parameters to select the SPI controller where the device is connected

Name	Length	Fmt	Description
POMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit Slave Receive) signal
PIMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within POMT of MTSR (Master Transmit Slave Receive) (SI) signal
POMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive Slave Transmit) signal

Name	Length	Fmt	Description
PIMR	**1**	**xx**	**Pin number of MRST signal** Pin number within POMR of MRST (Master Receive Slave Transmit) (SO) signal
POCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PICK	**1**	**xx**	**Pin number of SCK signal** Pin number within POCK of SCK (Serial Clock) signal
POSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PISL	**1**	**xx**	**Pin number of SLSO signal** Pin number within POSL of SLSO (Slave Select) signal

parameters to select pins for the device access
Port and Pin parameters to select the pins for the device access (bank 1 and bank 2)

Name	Length	Fmt	Description
PO_NON1	**1**	**xx**	**Port number of nON1 signal** Port number of negated ON1 signal
PI_NON1	**1**	**xx**	**Pin number of nON1 signal** Pin number within POnON1 of negated ON1 signal
PO_BSY1	**1**	**xx**	**Port number and TIM of BUSY1 signal** **Bit(s)** – **Description** **7:6** – **measurement configuration** **00_B:** HW; First TIM **01_B:** HW; Second TIM **10_B:** reserved **11_B:** SW **5:0** – **Port number of BUSY1 signal**
PI_BSY1	**1**	**xx**	**Pin number of BUSY1 signal** Pin number within POBSY1 of BUSY1 signal
PO_NON2	**1**	**xx**	**Port number of nON2 signal** Port number of negated ON2 signal
PI_NON2	**1**	**xx**	**Pin number of nON2 signal** Pin number within POnON2 of negated ON2 signal
PO_BSY2	**1**	**xx**	**Port number and TIM of BUSY2 signal** **Bit(s)** – **Description** **7:6** – **measurement configuration** **00_B:** HW; First TIM **01_B:** HW; Second TIM **10_B:** reserved **11_B:** SW **5:0** – **Port number of BUSY2 signal**

Name	Length	Fmt	Description
PI_BSY2	**1**	**xx**	**Pin number of BUSY2 signal** Pin number within POBSY2 of BUSY2 signal

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D6	xx	xx	01	...

Parameters

response of define device pins command

6	+0	+1
(1..2*)	**MOD**	**CH**

Details

response of define device pins command

Name	Length	Fmt	Description
MOD	**1**	**xx**	**Module** used TIM submodule for busy time measurement. 0xFF if no module found.
CH	**1**	**xx**	**Channel** used channel of TIM submodule. 0xFF if no module found.

.

02 - Define adc channels

This command defines the ADC channels for which measurement results will be provided after injection. Description of parameter see start conversion and read result (command 10 of tsw 01)

Note: This command is obselete and will be removed in future. Use TSW_DF instead!

Note: The used adc channels have to be configured with TSW_01.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D6	02	01	...

Parameters

input pins for adc (repeatable up to eight inputs)

5	+0	+1
(1..8*)	**CHA**	**MUX**

Details

input pins for adc (repeatable up to eight inputs)

Name	Length	Fmt	Description
CHA	**1**	**xx**	**Index of the channel**
MUX	**1**	**xx**	**Channel of the external multiplexer**

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	D6	xx	xx	01

.

03 - Define indication pin

This command is used to define a pin which is toggled to indicate the begin (high) and end (low) of analog digital conversion during injections. This feature is only used for debugging purposes and is not necessary for normal function.

Note: This command is obselete and will be removed in future. Use TSW_DF instead!

Note: The used pin have to be initialized with TSW_04.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D6	03	01	...

Parameters

parameters to select pin for the adc indication

5	+0	+1
(1*)	**PORT**	**PIN**

Details

parameters to select pin for the adc indication
Port and Pin parameter to select the pin for the adc indication

Name	Length	Fmt	Description
PORT	**1**	**xx**	**Port number** Port number for indication pin
PIN	**1**	**xx**	**Pin number** Pin number for indication pin within port

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	D6	xx	xx	01

.

05 - Define sync pulse time

This command is used to define the time for the sync pulse. This time is predefined with 800ns. This command is only necessary if another time is needed.

Note: This command is obselete and will be removed in future. Use TSW_DF instead!

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D6	05	01	...

Parameters

parameters to define the sync pulse time

...	+0
(1*)	**LOWTIME**

Details

parameters to define the sync pulse time

Name	Length	Fmt	Description
LOWTIME	**2**	**xxxx**	**Low time** Low time for the sync pulse. Format: RVVV (see table 1)

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D6	xx	xx	01	...

Parameters

response of define sync pulse time command

6	+0	+4
(1*)	**TICKS_P**	**TICKS_N**

Details

response of define sync pulse time command

Name	Length	Fmt	Description
TICKS_P	**4**	**xxxxxxxx**	**old time** previously time for sync pulse width (in ticks)
TICKS_N	**4**	**xxxxxxxx**	**new time** new time for sync puls width (in ticks)

.

06 - Define spi data

This command defines the spi data used in event SPI_CMD. The SPI command (Byte 1, BURST_XN) is added atomatically by the event, so only the data (Byte 2 until Byte n) has to be defined

Note: This command is obselete and will be removed in future. Use TSW_DF instead!

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D6	06	01	...

Parameters

parameters to define the spi data

...	+0
(1*)	**BURST**

parameters to define the spi data for burst A

...	+0
(1*)	**DATA**

parameters to define the spi data for burst B

...	+0
(1*)	**DATA**

parameters to define the spi data for burst C

...	+0
(1*)	**DATA**

parameters to define the spi data for burst D

...	+0
(1*)	**DATA**

Details

parameters to define the spi data

Name	Length	Fmt	Description
BURST	**1**	**xx**	**Burst type**

Burst type	**BURST**
SPI BURST A	0x00
SPI BURST B	0x01
SPI BURST C	0x02

Name	Length	Fmt	Description	
			Burst type	**BURST**
			SPI BURST D	0x03

parameters to define the spi data for burst A

Name	Length	Fmt	Description
DATA	**1**	**xx..**	**Burst A data (9)**

parameters to define the spi data for burst B

Name	Length	Fmt	Description
DATA	**1**	**xx..**	**Burst B data (9)**

parameters to define the spi data for burst C

Name	Length	Fmt	Description
DATA	**1**	**xx..**	**Burst C data (12)**

parameters to define the spi data for burst D

Name	Length	Fmt	Description
DATA	**1**	**xx..**	**Burst D data (15)**

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	D6	xx	xx	01

.

10 - Set injection and measurement times

This command defines the injection and measurement times for each cylinder. This is done by a set of events. Each event is processed in the order it appears in the command. The whole set of events is one cycle used in command 0x40.

Note: This command is obselete and will be removed in future. Use TSW_DF instead!

Hint: To address device 2 (if configured) simply add 0x40 to the event type.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D6	10	01	...

Parameters

Parameter set for event

5	+0	+1
(1..n*)	**SPI_CMD**	**WAIT**

Parameter set for event

8 (1..n*)	+0	+1
	S_PULSE	**WAIT**

Parameter set for event

11 (1..n*)	+0	+1	+3
	L_PULSE	**WAIT**	**LOW**

Parameter set for start ADC event

16 (1..n*)	+0	+1
	ST_ADC	**WAIT**

Parameter set for read ADC result event

19 (1..n*)	+0	+1
	RD_ADC	**WAIT**

Parameter set for disable WD interrupt

22 (1..n*)	+0	+1
	DIS_INT	**WAIT**

Parameter set for enable WD interrupt

25 (1..n*)	+0	+1
	ENA_INT	**WAIT**

Parameter set for event

28 (1..n*)	+0	+1
	SET_RST	**WAIT**

Parameter set for reset event table

... (1*)	+0
	RST_TAB

Details

Parameter set for event
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
SPI_CMD	**1**	**xx**	**Event type**

Event	**Event type**
SPI BURST A1, A2, ..., A6	0x00 - 0x05
SPI BURST B1, B2, ..., B6	0x08 - 0x0D
SPI BURST C1, C2, ..., C6	0x10 - 0x15

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td><table><tr><th>Event</th><th>Event type</th></tr><tr><td>SPI BURST D1, D2, ..., D6</td><td>0x18 - 0x1D</td></tr><tr><td>SPI NCD_MEAS_1</td><td>0x0E</td></tr><tr><td>SPI NCD_MEAS_2</td><td>0x0F</td></tr></table></td></tr>
<tr><td>WAIT</td><td>2</td><td>xxxx</td><td>Wait time
Wait time before the start of the event Format: RVVV (see table 1)</td></tr>
</table>

Parameter set for event
Multiple parameter sets may be appended to create a complete sequence of events

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>S_PULSE</td><td>1</td><td>xx</td><td>Event type<table><tr><th>Event</th><th>Event type</th></tr><tr><td>Sync Pulse ON1</td><td>0x06</td></tr><tr><td>Sync Pulse ON2</td><td>0x07</td></tr></table></td></tr>
<tr><td>WAIT</td><td>2</td><td>xxxx</td><td>Wait time
Wait time before the start of the event Format: RVVV (see table 1)</td></tr>
</table>

Parameter set for event
Multiple parameter sets may be appended to create a complete sequence of events

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>L_PULSE</td><td>1</td><td>xx</td><td>Event type<table><tr><th>Event</th><th>Event type</th></tr><tr><td>Low Pulse ON1</td><td>0x16</td></tr><tr><td>Low Pulse ON2</td><td>0x17</td></tr></table></td></tr>
<tr><td>WAIT</td><td>2</td><td>xxxx</td><td>Wait time
Wait time before the start of the event Format: RVVV (see table 1)</td></tr>
<tr><td>LOW</td><td>2</td><td>xxxx</td><td>Low time
Low time for the pulse Format: RVVV (see table 1)</td></tr>
</table>

Parameter set for start ADC event
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
ST_ADC	**1**	**xx**	**Event type** In this case "0x31" for ST_ADC
WAIT	**2**	**xxxx**	**Wait time** Wait time before the start of the analog digital conversion event Format: RVVV (see table 1)

Parameter set for read ADC result event
Multiple parameter sets may be appended to create a complete sequence of events

Note: If the conversions are not finished, this event waits until they are.

Name	Length	Fmt	Description
RD_ADC	**1**	**xx**	**Event type** In this case "0x30" for RD_ADC
WAIT	**2**	**xxxx**	**Wait time** Wait time before the read of the analog digital conversion result Format: RVVV (see table 1)

Parameter set for disable WD interrupt
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
DIS_INT	**1**	**xx**	**Event type** In this case "0x20" for DIS_INT
WAIT	**2**	**xxxx**	**Wait time** Wait time before the start of the disable WD interrupt event Format: RVVV (see table 1)

Parameter set for enable WD interrupt
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
ENA_INT	**1**	**xx**	**Event type** In this case "0x21" for ENA_INT
WAIT	**2**	**xxxx**	**Wait time** Wait time before the start of the enable WD interrupt event Format: RVVV (see table 1)

Parameter set for event
Multiple parameter sets may be appended to create a complete sequence of events

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>SET_RST</td><td>1</td><td>xx</td><td>Event type
<table>
<tr><th>Event</th><th>Event type</th></tr>
<tr><td>Set Rst Pin Low</td><td>0x1E</td></tr>
<tr><td>Set Rst Pin High</td><td>0x1F</td></tr>
</table></td></tr>
<tr><td>WAIT</td><td>2</td><td>xxxx</td><td>Wait time
Wait time before the start of the event Format: RVVV (see table 1)</td></tr>
</table>

Parameter set for reset event table

Name	Length	Fmt	Description
RST_TAB	**1**	**xx**	**Event type** In this case "0xFF" for RST_TAB to reset event table

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D6	xx	xx	01	...

Parameters

response of set injection and measurement times command

6 (1*)	+0 **USED**	+1 **FREE**

Details

response of set injection and measurement times command

Name	Length	Fmt	Description
USED	**1**	**xx**	**Number of defined events** Number of totally defined events
FREE	**1**	**xx**	**Number of free events** Number of free events

.

20 - Init device (device 1)

This command is used to initialise the device. For the initialisation values see table 2.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D6	20	01	...

Parameters

parameters to configure the device

5 (1*)	+0 **CONFIG1**	+1 **CONFIG2**	+2 **CONFIG3**	+3 **MEAS**

optional parameter of init device command

9 (0..n*)	+0 **ADR**	+1 **VAL**

Details

parameters to configure the device

Name	Length	Fmt	Description
CONFIG1	**1**	**xx**	**Config register 1** Content of config register 1. If necessary, Bit 4 (cfg_par) is corrected by command.
CONFIG2	**1**	**xx**	**Config register 2** Content of config register 2. If necessary, Bit 4 (cfg_par) is corrected by command.
CONFIG3	**1**	**xx**	**Config register 3** Content of config register 3.

Name	Length	Fmt	Description
MEAS	**1**	**xx**	**Meas register** Content of meas register.

optional parameter of init device command

Name	Length	Fmt	Description
ADR	**1**	**xx**	**Register address** **Bit(s)** | **Description** **7:0** | **Register address**
VAL	**1**	**xx**	**Register value** **Bit(s)** | **Description** **7:0** | **Register value**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D6	xx	xx	01	...

Parameters

response of init device command

6	+0	+1
(1*)	**CHIPID**	**CHIPREV**

additional response of command

...	+0
(0..n*)	**VAL**

Details

response of init device command

Name	Length	Fmt	Description
CHIPID	**1**	**xx**	**Chip identification** Chip identification from ID_REG (0x7C for CY374)
CHIPREV	**1**	**xx**	**Chip revision** Chip revision from REV_REG

additional response of command

Name	Length	Fmt	Description
VAL	**1**	**xx**	**Register value** **Bit(s)** | **Description** **7:0** | **Register value**

.

21 - Init device (device 2)

This command is used to initialise the CY374. It is used for the second configured device. Description see test step command 0x20.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	D6	21	01

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	D6	xx	xx	01

.

28 - Set/get register value(s) (device 1)

This command is used to transfer data to or from the device.

Example: To write the value xy to the Meas-Register, use the sequence 11xy for ADR/VAL.

Example: To read out Status-Register, Flag-Register, Diagnose-Register 1 and Diagnose-Register 2, use the sequence 0400 1200 0600 0800 for ADR/VAL.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D6	28	01	...

Parameters

parameter for command

5	+0	+1
(0..n*)	**ADR**	**VAL**

Details

parameter for command

Name	Length	Fmt	Description
ADR	**1**	**xx**	**Register address** **Bit(s)** / **Description** **7:0** / **Register address**

Name	Length	Fmt	Description
VAL	**1**	**xx**	**Register value** **Bit(s)** | **Description** **7:0** | **Register value**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D6	xx	xx	01	...

Parameters

response of command

...	+0
(0..n*)	**VAL**

Details

response of command

Name	Length	Fmt	Description
VAL	**1**	**xx**	**Register value** **Bit(s)** | **Description** **7:0** | **Register value**

.

29 - Set/get register value(s) (device 2)

This command is used to transfer data to or from the device. It is used for the second configured device. Description see test step command 0x28.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	D6	29	01

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	D6	xx	xx	01

.

30 - Start DC/DC (device 1)

This command starts the DC/DC converters.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D6	30	01	...

Parameters

parameters for command

...	+0
(1*)	**TIMEOUT**

Details

parameters for command

Name	Length	Fmt	Description
TIMEOUT	**2**	**xxxx**	**Buffer charge timeout** If the charging procedure exceeds this time, it will be aborted by transmitting the SPI command DC_OFF. A zero value means no supervision. Format: RVVV where R=Resolution, V=Value [s] (see table 1)

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D6	xx	xx	01	...

Parameters

response of command

...	+0
(1*)	**TI**

Details

response of command
.

Name	Length	Fmt	Description
TI	**2**	**xxxx**	**Startup time of DCDC** Format: RVVV where R=Resolution, V=Value [s] (see table 1).

.

31 - Start DC/DC (device 2)

This command starts the DC/DC converters. It is used for the second configured device. Description see test step command 0x30.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	D6	31	01

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	D6	xx	xx	01

.

40 - Start injection cycles

This command starts the injection cycles which has to be configured by command 0x10.

Note: This command is obselete and will be removed in future. Use TSW_DF instead!

Note: Due to resource limitation, the measurement of busy signals is not working. Use TSW_DF instead!

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D6	40	01	...

Parameters

parameters for command

...	+0
(1*)	**CYCLES**

Details

parameters for command

Name	Length	Fmt	Description
CYCLES	**4**	**xxxxxxxx**	**Number of injection cycles** This parameter sets the number of subsequent injection cycles. A zero value means endless.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D6	xx	xx	01	...

Parameters

response of command (if CYCLES > 0)

6	+0	+1	+2	+3
(1*)	**STAT_P**	**STAT_A**	**FLAGS_P**	**FLAGS_A**

response of command for bank 1 (if CYCLES > 0)

10	+0	+1	+2	+4
(1*)	**DIAG1_P**	**DIAG1_A**	**BSY1MIN**	**BSY1MAX**

response of command for bank 2 (if CYCLES > 0)

16	+0	+1	+2	+4
(1*)	**DIAG2_P**	**DIAG2_A**	**BSY2MIN**	**BSY2MAX**

response of command for second device (if configured and CYCLES > 0)

22	+0	+1	+2	+3
(1*)	**STAT_P**	**STAT_A**	**FLAGS_P**	**FLAGS_A**

response of command for bank 1 for second device (if configured and CYCLES > 0)

26	+0	+1	+2	+4
(1*)	**DIAG1_P**	**DIAG1_A**	**BSY1MIN**	**BSY1MAX**

response of command for bank 2 for second device (if configured and CYCLES > 0)

32	+0	+1	+2	+4
(1*)	**DIAG2_P**	**DIAG2_A**	**BSY2MIN**	**BSY2MAX**

repeatable ADC measurement results (if CYCLES > 0)

...	+0
(0..n*)	**ADV**

Details

response of command (if CYCLES > 0)

Name	Length	Fmt	Description
STAT_P	1	xx	Status register before injection
STAT_A	1	xx	Status register after injection
FLAGS_P	1	xx	FLAGS register before injection
FLAGS_A	1	xx	FLAGS register after injection

response of command for bank 1 (if CYCLES > 0)

Name	Length	Fmt	Description
DIAG1_P	1	xx	Diagnostic register of bank 1 before injection
DIAG1_A	1	xx	Diagnostic register of bank 1 after injection

Name	Length	Fmt	Description
BSY1MIN	**2**	**xxxx**	**Busy time minimum of bank 1 (Format: RVVV)**
BSY1MAX	**2**	**xxxx**	**Busy time maximum of bank 1 (Format: RVVV)**

response of command for bank 2 (if CYCLES > 0)

Name	Length	Fmt	Description
DIAG2_P	**1**	**xx**	**Diagnostic register of bank 2 before injection**
DIAG2_A	**1**	**xx**	**Diagnostic register of bank 2 after injection**
BSY2MIN	**2**	**xxxx**	**Busy time minimum of bank 2 (Format: RVVV)**
BSY2MAX	**2**	**xxxx**	**Busy time maximum of bank 2 (Format: RVVV)**

response of command for second device (if configured and CYCLES > 0)

Name	Length	Fmt	Description
STAT_P	**1**	**xx**	**Status register before injection**
STAT_A	**1**	**xx**	**Status register after injection**
FLAGS_P	**1**	**xx**	**FLAGS register before injection**
FLAGS_A	**1**	**xx**	**FLAGS register after injection**

response of command for bank 1 for second device (if configured and CYCLES > 0)

Name	Length	Fmt	Description
DIAG1_P	**1**	**xx**	**Diagnostic register of bank 1 before injection**
DIAG1_A	**1**	**xx**	**Diagnostic register of bank 1 after injection**
BSY1MIN	**2**	**xxxx**	**Busy time minimum of bank 1 (Format: RVVV)**
BSY1MAX	**2**	**xxxx**	**Busy time maximum of bank 1 (Format: RVVV)**

response of command for bank 2 for second device (if configured and CYCLES > 0)

Name	Length	Fmt	Description
DIAG2_P	**1**	**xx**	**Diagnostic register of bank 2 before injection**
DIAG2_A	**1**	**xx**	**Diagnostic register of bank 2 after injection**
BSY2MIN	**2**	**xxxx**	**Busy time minimum of bank 2 (Format: RVVV)**
BSY2MAX	**2**	**xxxx**	**Busy time maximum of bank 2 (Format: RVVV)**

repeatable ADC measurement results (if CYCLES > 0)
For each ReadADCEvent all configured adc channels are reported in the order they appear in the define adc channels command (0x02). This is repeated for all ReadADCEvent. After that, if NCD is configured, all measurements from Read NCD ADC event are reported.

Name	Length	Fmt	Description
ADV	**2**	**xxxx**	**ADC measurement result value (if configured)**

.

41 - Start injection cycles (launch mode)

This command starts the injection cycles which has to be configured by command 0x10.

Note: This command is obselete and will be removed in future. Use TSW_DF instead!

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D6	41	01	...

Parameters

parameters for command

...	+0
(1*)	**CYCLES**

Details

parameters for command

Name	Length	Fmt	Description
CYCLES	**4**	**xxxxxxxx**	**Number of injection cycles** This parameter sets the number of subsequent injection cycles. A zero value means endless.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D6	xx	xx	01	...

Parameters

response of command (if CYCLES > 0)

...	+0
(1..n*)	**BDATA**

Details

response of command (if CYCLES > 0)

Name	Length	Fmt	Description
BDATA	**1**	**xx..**	**answer of last spi burst command [10|13|16]**

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**

Code	Description
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0xF0	**No SPI controller on given ports found.**
0xF1	**HAL parameter error**
0xF2	**SPI communication/protocol failed**
0xF3	**SPI communication/protocol failed (FIX)**
0xF4	**SPI communication/protocol failed (TRANS_F)**
0xF6	**some defines missing**
0xF7	**no more events possible**
0xF8	**no more resoures available**
0xF9	**invalid configuration**

TSW_D8 - Piezo injection with CJ871

Test step description

Date: 09.03.2016

Created: 18.01.2017 09:12.

This TSW provides functions for test of injection powerstage for piezo injectors (CJ871).

Important: before use, initialize ports, SPI and setup a 2MHZ clock for CJ871.

All physical parameters like period, timeouts etc. use the 16 bit RVVV format, where R (4 bit) is the resolution (see following table) and VVV (12 bit) is the value of the parameter. For example 2ms would be 0xD002 (or 0xC014 or 0xB0C8 or 0xA7D0).

Table 1. Resolution table

Value of R	Resolution
0x0	1.0E+0s = 1s
0x1	1.0E+1s = 10s
0x2	1.0E+2s = 100s
0x3	1.0E+3s = 1000s
...	...
0x8	1.0E-8s = 10ns
0x9	1.0E-7s = 100ns
0xA	1.0E-6s = 1µs
0xB	1.0E-5s = 10µs
0xC	1.0E-4s = 100µs
0xD	1.0E-3s = 1ms
0xE	1.0E-2s = 10ms
0xF	1.0E-1s = 100ms

Test step commands

.

01 - Define device pins

This command defines four controller pins used for spi communication and eleven controller pins used for digital device access. Up to 2 devices could be defined.

Note: the used pins have to be initialized with TSW_04.

Note: Because the triggered AD measurement for the JDP is not working in cut 1.x (e5607: SARADC: External trigger on SARADC module are not usable), a common workaround with interrupt is used. Therefore a timer resource from the GTM (parameter TIMIDX) is needed. Because some pins are connectable to more than one TIM, the TIMIDX parameter defines which one is used. This is also necessary to avoid conflict with other loaded teststeps. For example the port P10.8/PH[12] in JDP_DEV3 can use TIM0_5 and TIM1_5. If TIM0_5 should be used, choose 0 for TIMIDX, if TIM1_5 should be used, choose 1.

Hint: if some pins of a bank are not connected and therefore not used, use the same pins for this bank as for a used bank to avoid erroneous response.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D8	01	01	...

Parameters

parameters to select a SPI controller for the device

5 (1..n*)	+0	+1	+2	+3	+4	+5
	POMT	**PIMT**	**POMR**	**PIMR**	**POCK**	**PICK**

... (1..n*)	+6	+7
	POSL	**PISL**

parameters to select pins for the device access (bank 1)

13 (1..n*)	+0	+1	+2	+3	+4	+5
	POBLE1	**PIBLE1**	**POELE1**	**PIELE1**	**POZS1**	**PIZS1**

parameters to select pins for the device access (bank 2)

19 (1..n*)	+0	+1	+2	+3	+4	+5
	POBLE2	**PIBLE2**	**POELE2**	**PIELE2**	**POZS2**	**PIZS2**

parameters to select pins for the device access (bank 3)

25 (1..n*)	+0	+1	+2	+3	+4	+5
	POBLE3	**PIBLE3**	**POELE3**	**PIELE3**	**POZS3**	**PIZS3**

parameters to select pins for the device access

31 (1..n*)	+0	+1	+2	+3	+4
	POTRADC	**PITRADC**	**TIMIDX**	**PONRST**	**PINRST**

Details

parameters to select a SPI controller for the device
Port and Pin parameters to select the SPI controller where the device is connected

Name	Length	Fmt	Description
POMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit Slave Receive) signal
PIMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within POMT of MTSR (Master Transmit Slave Receive) (SI) signal

Name	Length	Fmt	Description
POMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive Slave Transmit) signal
PIMR	**1**	**xx**	**Pin number of MRST signal** Pin number within POMR of MRST (Master Receive Slave Transmit) (SO) signal
POCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PICK	**1**	**xx**	**Pin number of SCK signal** Pin number within POCK of SCK (Serial Clock) signal
POSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PISL	**1**	**xx**	**Pin number of SLSO signal** Pin number within POSL of SLSO (Slave Select) signal

parameters to select pins for the device access (bank 1)
Port and Pin parameters to select the pins for the device access (bank 1)

Name	Length	Fmt	Description
POBLE1	**1**	**xx**	**Port number of BLE1 signal** Port number of BLE1 signal
PIBLE1	**1**	**xx**	**Pin number of BLE1 signal** Pin number within POBLE1 of BLE1 signal
POELE1	**1**	**xx**	**Port number of ELE1 signal** Port number of ELE1 signal
PIELE1	**1**	**xx**	**Pin number of ELE1 signal** Pin number within POELE1 of ELE1 signal
POZS1	**1**	**xx**	**Port number of ZS1 signal** Port number of ZS1 signal
PIZS1	**1**	**xx**	**Pin number of ZS1 signal** Pin number within POZS1 of ZS1 signal

parameters to select pins for the device access (bank 2)
Port and Pin parameters to select the pins for the device access (bank 2)

Name	Length	Fmt	Description
POBLE2	**1**	**xx**	**Port number of BLE2 signal** Port number of BLE2 signal
PIBLE2	**1**	**xx**	**Pin number of BLE2 signal** Pin number within POBLE2 of BLE2 signal
POELE2	**1**	**xx**	**Port number of ELE2 signal** Port number of ELE2 signal
PIELE2	**1**	**xx**	**Pin number of ELE2 signal** Pin number within POELE2 of ELE2 signal
POZS2	**1**	**xx**	**Port number of ZS2 signal** Port number of ZS2 signal
PIZS2	**1**	**xx**	**Pin number of ZS2 signal** Pin number within POZS2 of ZS2 signal

parameters to select pins for the device access (bank 3)
Port and Pin parameters to select the pins for the device access (bank 3)

Name	Length	Fmt	Description
POBLE3	**1**	**xx**	**Port number of BLE3 signal** Port number of BLE3 signal
PIBLE3	**1**	**xx**	**Pin number of BLE3 signal** Pin number within POBLE3 of BLE3 signal
POELE3	**1**	**xx**	**Port number of ELE3 signal** Port number of ELE3 signal
PIELE3	**1**	**xx**	**Pin number of ELE3 signal** Pin number within POELE3 of ELE3 signal
POZS3	**1**	**xx**	**Port number of ZS3 signal** Port number of ZS3 signal
PIZS3	**1**	**xx**	**Pin number of ZS3 signal** Pin number within POZS3 of ZS3 signal

parameters to select pins for the device access
Port and Pin parameters to select the pins for the device access

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>POTRADC</td><td>1</td><td>xx</td><td>Port number of TRIG-ADC signal
Port number of TRIG-ADC signal</td></tr>
<tr><td>PITRADC</td><td>1</td><td>xx</td><td>Pin number of TRIG-ADC signal
Pin number within POTRADC of TRIG-ADC signal</td></tr>
<tr><td>TIMIDX</td><td>1</td><td>xx</td><td>Index TIMx_y of GTM
GTM Timer Input Module number which is connected to the pin connected to TRIG-ADC.
0xFF if triggered ADC is not used.

Table 2.
<table>
<tr><th>Value</th><th>TIM #</th></tr>
<tr><td>0x00</td><td>First TIM</td></tr>
<tr><td>0x01</td><td>Second TIM</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>0xFF</td><td>triggered ADC is not used</td></tr>
</table></td></tr>
<tr><td>PONRST</td><td>1</td><td>xx</td><td>Port number of NRST signal
Port number of NRST signal</td></tr>
<tr><td>PINRST</td><td>1</td><td>xx</td><td>Pin number of NRST signal
Pin number within PONRST of NRST signal</td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	D8	xx	xx	01

.

02 - Define adc channels

This command defines the ADC channels for which measurement results will be provided after injection. Description of parameter see start conversion and read result (command 10 of tsw 01)

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D8	02	01	...

Parameters

repeatable parameter

5	+0	+1
(1..n*)	**CHA**	**MUX**

Details

repeatable parameter

Name	Length	Fmt	Description
CHA	**1**	**xx**	**Index of the channel**
MUX	**1**	**xx**	**Channel of the external multiplexer**

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	D8	xx	xx	01

.

03 - Define indication pin

This command .is used to define a pin which is toggled to indicate the begin (high) and end (low) of analog digital conversion during injections. This feature is only used for debugging purposes and is not necessary for normal function.

Note: the used pin have to be initialized with TSW_04.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D8	03	01	...

Parameters

parameters to select pin for the adc indication

5	+0	+1
(1*)	**RORT**	**PIN**

Details

parameters to select pin for the adc indication
Port and Pin parameter to select the pin for the adc indication

Name	Length	Fmt	Description
RORT	**1**	**xx**	**Port number** Port number for indication pin
PIN	**1**	**xx**	**Pin number** Pin number for indication pin within port

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	D8	xx	xx	01

.

10 - Set injection and measurement times

This command defines the injection and measurement times for each cylinder. This is done by a set of events. Each event is processed in the order it appears in the command. The whole set of events is one cycle used in command 0x40. For the mapping of the cylinders see table 3.

Table 3. Cylinder mapping table

cylinder number	device	bank	cylinder
01 (0x01)	1	1	1
02 (0x02)	1	1	2
03 (0x03)	1	2	1
04 (0x04)	1	2	2
05 (0x05)	1	3	1
06 (0x06)	1	3	2
07 (0x07)	2	1	1
08 (0x08)	2	1	2
09 (0x09)	2	2	1
10 (0x0A)	2	2	2
11 (0x0B)	2	3	1
12 (0x0C)	2	3	2

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D8	10	01	...

Parameters

Parameter set for start charge event

5	+0	+1	+3
(1..n*)	STRTCHR	WAIT	CYL

Parameter set for stop charge event

9	+0	+1	+3
(1..n*)	STOPCHR	WAIT	CYL

Parameter set for start discharge event

13	+0	+1	+3
(1..n*)	STRTDIS	WAIT	CYL

Parameter set for stop discharge event

17	+0	+1	+3
(1..n*)	STOPDIS	WAIT	CYL

Parameter set for start ADC event

21	+0	+1
(1..n*)	STRTADC	WAIT

Parameter set for read ADC result event

24	+0	+1
(1..n*)	READADC	WAIT

Parameter set for set spi register event table

27	+0	+1	+2	+3
(1*)	SETSPI	DEVICE	REGADDR	REGVAL

Parameter set for set reset pin event

31	+0	+1	+3
(1..n*)	RSTPIN	WAIT	DEVLEV

Parameter set for reset event table

35	+0
(1*)	RSTTABL

Details

Parameter set for start charge event
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
STRTCHR	**1**	**xx**	**Event type** In this case "0x11" for START CHARGE
WAIT	**2**	**xxxx**	**Wait time**

Name	Length	Fmt	Description
			Wait time before the start of the start charge event Format: RVVV (see table 1)
CYL	**1**	**xx**	**cylinder number** cylinder number (1-12, mapping see table 3) for the start charge event

Parameter set for stop charge event
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
STOPCHR	**1**	**xx**	**Event type** In this case "0x10" for STOP CHARGE
WAIT	**2**	**xxxx**	**Wait time** Wait time before the start of the stop charge event Format: RVVV (see table 1)
CYL	**1**	**xx**	**cylinder number** cylinder number (1-12, mapping see table 3) for the stop charge event

Parameter set for start discharge event
Multiple parameter sets may be appended to create a complete sequence of events

Note: Discharge is only possible, if the cylinder is charged before.

Name	Length	Fmt	Description
STRTDIS	**1**	**xx**	**Event type** In this case "0x01" for START DISCHARGE
WAIT	**2**	**xxxx**	**Wait time** Wait time before the start of the start discharge event Format: RVVV (see table 1)
CYL	**1**	**xx**	**cylinder number** cylinder number (1-12, mapping see table 3) for the start discharge event

Parameter set for stop discharge event
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
STOPDIS	**1**	**xx**	**Event type** In this case "0x00" for STOP DISCHARGE
WAIT	**2**	**xxxx**	**Wait time** Wait time before the start of the stop discharge event Format: RVVV (see table 1)
CYL	**1**	**xx**	**cylinder number** cylinder number (1-12, mapping see table 3) for the stop discharge event

Parameter set for start ADC event
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
STRTADC	**1**	**xx**	**Event type**

Name	Length	Fmt	Description
			In this case "0x81" for START ADC
WAIT	**2**	**xxxx**	**Wait time** Wait time before the start of the analog digital conversion event Format: RVVV (see table 1)

Parameter set for read ADC result event
Multiple parameter sets may be appended to create a complete sequence of events

Note: If the conversions are not finished, this event does not wait until they are.

Name	Length	Fmt	Description
READADC	**1**	**xx**	**Event type** In this case "0x80" for READ ADC
WAIT	**2**	**xxxx**	**Wait time** Wait time before the read of the analog digital conversion result Format: RVVV (see table 1)

Parameter set for set spi register event table
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
SETSPI	**1**	**xx**	**Event type** In this case "0x40" for SET SPI REGISTER
DEVICE	**1**	**xx**	**device [0|1]** device for the write or read
REGADDR	**1**	**xx**	**register address** register address to write or read shifted left one bit (+1 for read)
REGVAL	**1**	**xx**	**register value** register value to write (ignored for read)

Parameter set for set reset pin event
Multiple parameter sets may be appended to create a complete sequence of events

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>RSTPIN</td><td>1</td><td>xx</td><td>Event type
In this case "0x20" for SET RESET PIN</td></tr>
<tr><td>WAIT</td><td>2</td><td>xxxx</td><td>Wait time
Wait time before the start of the set reset pin event Format: RVVV (see table 1)</td></tr>
<tr><td>DEVLEV</td><td>1</td><td>xx</td><td>device and level
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:5</td><td>reserved</td></tr>
<tr><td>4</td><td>Device
0_B: Device 1
1_B: Device 2</td></tr>
<tr><td>3:1</td><td>reserved</td></tr>
<tr><td>0</td><td>Level</td></tr>
</table>
</td></tr>
</table>

Name	Length	Fmt	Description	
			Bit(s)	**Description**
				0_B: Low level
				1_B: High level

Parameter set for reset event table

Name	Length	Fmt	Description
RSTTABL	**1**	**xx**	**Event type** In this case "0xFF" for RESET TABLE

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D8	xx	xx	01	...

Parameters

Response of set injection and measurement times command

6	+0
(1*)	**EVENTS**

Details

Response of set injection and measurement times command

Name	Length	Fmt	Description
EVENTS	**1**	**xx**	**Number of defined events** Number of totally defined events

.

20 - Init device

This command is used to initialise the CJ871. For the initialisation values see table 4.

Note: Before the initialiation a SPI-RESET command is sent to the device. This is a write of 0xAD to the register RESET (0x7F). After the initialisation, the register CONF_STA (0x40) is set according to the parameter of the command.

Table 4. Initialisation values (in hex) for command 0x20

Address	Register	Value
0x02	T_MAX	0xC3
0x03	T_TRDLY	0x04
0x04	T_HSL	0x07
0x05	T_ANSMIN	0x14
0x06	T_ANSDIF	0xFF
0x07	T_NULLST	0x0A
0x08	T_OFFU	0x96

Address	Register	Value
0x09	T_DIAGKSUB	0x78
0x0A	T_IPSEL	0x03
0x0B	T_PIEZO	0x07
0x0C	T_DIAGWT1	0x3C
0x0D	T_KSUB	0x0B
0x0E	T_KSMLA	0x06
0x12	UBOT	0x07
0x13	ULOW	0x07
0x14	UMIN	0x12
0x15	IPSEL	0x5F
0x16	IKSUB	0x66
0x1C-0x2D	CHG_xxx	0x48
0x2E-0x3F	DCH_xxx	0x53
0x41	CONF_DYN	0x82
0x4A	DIAGR_MSK	0x00
0x4B	T_FILGSTOP	0x64
0x4C	T_LRMIN	0x00
0x4D	T_LRMAX	0x7F
0x4E	T_TRIG	0x0F
0x52	CHG_B	0x5C
0x53	DCH_B	0x63
0x54	IPMAX	0xAA
0x55	IDCDC	0x66
0x56	UBUFF	0x64
0x57	UTOP	0x3D
0x58	MTRIG	0x00
0x59	ATRIG	0x00
0x5B	IKSMLA	0x0F
0x40	CONF_STA	0X010XX1 bin

The table shows the initalisation values which are written to the device by command 0x20.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D8	20	01	...

Parameters

parameters to configure the device

5	+0	+1	+2
(1..n*)	**DVCN**	**BANKS**	**DCSYNC**

Details

parameters to configure the device

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>DVCN</td><td>1</td><td>xx</td><td>Device number (0-1)

Table 5.
<table>
<tr><th>Value</th><th>Mode</th></tr>
<tr><td>0x00</td><td>Device #1</td></tr>
<tr><td>0x01</td><td>Device #2</td></tr>
</table></td></tr>
<tr><td>BANKS</td><td>1</td><td>xx</td><td>Bank mode of the device

Table 6.
<table>
<tr><th>Value</th><th>Mode</th></tr>
<tr><td>0x00</td><td>3-Bank-System (1,2,3)</td></tr>
<tr><td>0x01</td><td>2-Bank-System (1,2)</td></tr>
</table></td></tr>
<tr><td>DCSYNC</td><td>1</td><td>xx</td><td>DCSYNC pin direction

Table 7.
<table>
<tr><th>Value</th><th>Mode</th></tr>
<tr><td>0x00</td><td>DCSYNC pin is input (slave device)</td></tr>
<tr><td>0x01</td><td>DCSYNC pin is output (master device)</td></tr>
</table>
Note: If there is only one device, it is the master device and DCSYNC must be output.</td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D8	xx	xx	01	...

Parameters

Response of init device command

6	+0	+1
(1..n*)	**CHIPID**	**CHIPREV**

Details

Response of init device command

Name	Length	Fmt	Description
CHIPID	**1**	**xx**	**Chip identification from ID_REG (0xF0 for CJ871/CJ870)**

Name	Length	Fmt	Description
CHIPREV	**1**	**xx**	**Chip revision from REV_REG**

.

30 - Start DC/DC

This command starts the DC/DC converters.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D8	30	01	...

Parameters

parameters for command

5	+0	+1
(1..n*)	**DVCN**	**TIMEOUT**

Details

parameters for command

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>DVCN</td><td>1</td><td>xx</td><td>Device number (0-1)

Table 8.<table><tr><th>Value</th><th>Mode</th></tr><tr><td>0x00</td><td>Device #1</td></tr><tr><td>0x01</td><td>Device #2</td></tr></table></td></tr>
<tr><td>TIMEOUT</td><td>2</td><td>xxxx</td><td>Buffer charge timeout
If the charging procedure exceeds this time, it will be aborted by transmitting the SPI-RESET command (write of 0xAD to the register RESET (0x7F)). A zero value means no supervision.

Format: RVVV where R=Resolution, V=Value [s] (see table 1)</td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D8	xx	xx	01	...

Parameters

Response of command

6	+0
(1..n*)	**TIME**

Details

Response of command

.

Name	Length	Fmt	Description
TIME	**2**	**xxxx**	**Startup time of DCDC** Format: RVVV where R=Resolution, V=Value [s] (see table 1)

.

40 - Start integrator adjustment

This command starts the integrator adjustment and reads back the results of AD conversions. This is done by first writing 0x00 into the MTRIG register to all configured devices to set the Q/U outputs to high-impedance and than writing 0x01 (ABG_ENA) into the ATRIG register of the addressed device to start the adjustment. If triggered ADC is used the results of the conversions are fetched as soon as the conversions are finished. A timeout time of (T_EICHWT + T_EICH + T_TRIG) * 2 is set in this case. If triggered ADC is not used the ADC is started approximately after (T_EICHWT + T_EICH). The AD channel which is connected to the Q/U output of the CJ871 has to be defined with the command 0x02 (define adc channels).

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D8	40	01	...

Parameters

parameters for command

5	+0	+1
(1*)	**DVCN**	**ADJTIME**

Details

parameters for command

Name	Length	Fmt	Description
DVCN	**1**	**xx**	**Device number (0-1)** **Table 9.** Value / Mode 0x00 / Device #1 0x01 / Device #2
ADJTIME	**1**	**xx**	**Adjustment time**

Name	Length	Fmt	Description
			Adjustment time in microseconds. The maximum are 63µs (=0x3Fµs). This value is written to register T_EICH.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D8	xx	xx	01	...

Parameters

repeatable ADC measurement results

6	+0
(0..n*)	**ADV**

Details

repeatable ADC measurement results
.

Name	Length	Fmt	Description
ADV	**2**	**xxxx**	**ADC measurement result value (if configured)**

.

50 - Start injection cycles

This command starts the injection cycles which has to be configured by command 0x10.

Note: The status and diagnostic register set in the response includes in this order the register **STATUS, STA_BANK1, STA_BANK2, STA_BANK3, DIAG_GL, DIAG_B1, DIAG_B2** and **DIAG_B3**.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	D8	50	01	...

Parameters

parameters for command

5	+0
(1*)	**CYCLES**

Details

parameters for command

Name	Length	Fmt	Description
CYCLES	**4**	**xxxxxxxx**	**Number of injection cycles**

Name	Length	Fmt	Description
			This parameter sets the number of subsequent injection cycles. A zero value means endless.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	D8	xx	xx	01	...

Parameters

Response of command for device 1 (if CYCLES > 0)

6	+0	+1
(1*)	**SDRS1_P**	**SDRS1_A**

Response of command for device 2 (if CYCLES > 0)

8	+0	+1
(1*)	**SDRS2_P**	**SDRS2_A**

repeatable ADC measurement results (if CYCLES > 0)

10	+0
(0..n*)	**ADV**

Details

Response of command for device 1 (if CYCLES > 0)

Name	Length	Fmt	Description
SDRS1_P	**1**	**xx..**	**Status and diagnostic register set of device 1 before injection**
SDRS1_A	**1**	**xx..**	**Status and diagnostic register set of device 1 after injection**

Response of command for device 2 (if CYCLES > 0)

Name	Length	Fmt	Description
SDRS2_P	**1**	**xx..**	**Status and diagnostic register set of device 2 before injection**
SDRS2_A	**1**	**xx..**	**Status and diagnostic register set of device 2 after injection**

repeatable ADC measurement results (if CYCLES > 0)

Name	Length	Fmt	Description
ADV	**2**	**xxxx**	**ADC measurement result value (if configured)**

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0xF0	**No SPI controller/ADC module on given ports found.**
0xF1	**HAL parameter error**
0xF2	**SPI communication/protocol failed**
0xF3	**SPI communication/protocol failed (FIX)**
0xF4	**SPI communication/protocol failed (ERR)**
0xF6	**some defines missing**
0xF7	**no more events possible**

Code	Description
0xF8	**no more resoures available**
0xF9	**invalid configuration**
0xFF	**HAL busy**

TSW_DA - Solenoid injection with CY335

Test step description

Date: 04.12.2015

Created: 18.01.2017 09:12.

This TSW provides functions for test of injection powerstage for solenoid injectors (CY335).

Important: before use, initialize SPI and setup a 8MHZ clock for CY335.

Test step commands

.

20 - Init device

This command is used to initialise the CY335. For the initialisation values see table 1. If this values are inappropriate, the optional parameters can be used to set user defined values for the internal registers. This is done by setting pairs of register address and wished register value.

Table 1. Initialisation values (in hex) for command 0x20

Address	Register	Value
0x1F	RELEASE RESET	0x66
0x03	MDAC	0x33
0x02	MODE	0x01
0x01	CFG1	0x01
0x09	DMSK	0xF4
0x2E	MFBOC	0x9F
0x30	DUCMAX	0x65
0x31	DUCMIN	0x48
0x32	DIMAX	0x6F
0x33	DUC90	0x5B
0x34	DUC80	0x52
0x50	TERR1	0x3A
0x51	TERR2	0x28
0x52	TSL	0x2E
0x53	TSKIP_SL0	0x08
0x54	TSKIP_SL1	0x05
0x55	TWAIT	0x03
0x56	TONMAX	0x8F
0x59	TSLD	0x0C
0x5C	IH1	0x4B

Address	Register	Value
0x60	IH2	0x53
0x64	IH3	0x36
0x68	IL1	0x4B
0x6C	IL2	0x4C
0x70	IL3	0x31
0x04	DCC	0xD4
0x40	PRGM00	0xA5
0x41	PRGM01	0xBB
0x42	PRGM02	0xF0
0x43	PRGM03	0xF0
0x44	PRGM04	0xF0
0x45	PRGM05	0xF0
0x46	PRGM06	0xF0
0x47	PRGM07	0xF0
0x48	PRGM08	0xF0
0x49	PRGM09	0xF0
0x4A	PRGM10	0xF0
0x4B	PRGM11	0xF0
0x4C	PRGM12	0xF0
0x4D	PRGM13	0xF0
0x4E	PRGM14	0xCC
0x4F	PRGM15	0x71
0x03	MDAC	0x33
0x02	MODE	command parameter MODE

The table shows the initalisation values which are written to the device by command 0x20.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	DA	20	01	...

Parameters

parameters to select a SPI controller to the device

5	+0	+1	+2	+3	+4	+5	+6	+7
(1*)	**POMT**	**PIMT**	**POMR**	**PIMR**	**POCK**	**PICK**	**POSL**	**PISL**

parameters to configure the device

13	+0	+1
(1*)	**CFG**	**MODE**

optional parameter of initialise device command

15	+0	+1
(0..n*)	ADR	VAL

Details

parameters to select a SPI controller to the device
Port and Pin parameters to select the SPI controller where the device is connected

Name	Length	Fmt	Description
POMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit Slave Receive) signal
PIMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within POMT of MTSR (Master Transmit Slave Receive) (SI) signal
POMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive Slave Transmit) signal
PIMR	**1**	**xx**	**Pin number of MRST signal** Pin number within POMR of MRST (Master Receive Slave Transmit) (SO) signal
POCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PICK	**1**	**xx**	**Pin number of SCK signal** Pin number within POCK of SCK (Serial Clock) signal
POSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PISL	**1**	**xx**	**Pin number of SLSO signal** Pin number within POSL of SLSO (Slave Select) signal

parameters to configure the device

Name	Length	Fmt	Description
CFG	**1**	**xx**	**Device configuration (for future use)**
MODE	**1**	**xx**	**Value for register MODE (0x02)**

Bank mode	use Value
single bank mode / 4 cylinder	0x40
two bank mode / 1 HS-FET	0x00
two bank mode / 4 cylinder / one YSEL per bank	0x82
two bank mode / 6 cylinder	0x02

optional parameter of initialise device command

Name	Length	Fmt	Description
ADR	**1**	**xx**	**Register address**

Bit(s)	Description
7	**reserved, must be zero**

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td><table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>6:0</td><td>Register address</td></tr></table></td></tr>
<tr><td>VAL</td><td>1</td><td>xx</td><td>Register value<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>7:0</td><td>Register value</td></tr></table></td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	DA	xx	xx	01	...

Parameters

Response of init device command

6 (1*)	+0	+1
	CHIPID	**CHIPREV**

Details

Response of init device command

Name	Length	Fmt	Description
CHIPID	**1**	**xx**	**Chip identification** Chip identification from DEV_ID (0x3B for CY335)
CHIPREV	**1**	**xx**	**Chip revision** Chip revision from REV_ID

.

28 - Get status

This command is used to read out the error and diagnosis register from the CY335.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	DA	28	01	...

Parameters

parameters to select a SPI controller to the device

5 (1*)	+0	+1	+2	+3	+4	+5	+6	+7
	POMT	**PIMT**	**POMR**	**PIMR**	**POCK**	**PICK**	**POSL**	**PISL**

Details

parameters to select a SPI controller to the device

Port and Pin parameters to select the SPI controller where the device is connected

Name	Length	Fmt	Description
POMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit Slave Receive) signal
PIMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within POMT of MTSR (Master Transmit Slave Receive) (SI) signal
POMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive Slave Transmit) signal
PIMR	**1**	**xx**	**Pin number of MRST signal** Pin number within POMR of MRST (Master Receive Slave Transmit) (SO) signal
POCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PICK	**1**	**xx**	**Pin number of SCK signal** Pin number within POCK of SCK (Serial Clock) signal
POSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PISL	**1**	**xx**	**Pin number of SLSO signal** Pin number within POSL of SLSO (Slave Select) signal

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	DA	xx	xx	01	...

Parameters

Response of get status command

6	+0	+1	+2	+3	+4
(1*)	**ERR**	**DIAG**	**DIA_B1**	**DIA_B2**	**DIA_Z**

Details

Response of get status command

Name	Length	Fmt	Description
ERR	**1**	**xx**	**Value of register ERR**
DIAG	**1**	**xx**	**Value of register DIAG**
DIA_B1	**1**	**xx**	**Value of register DIA_BNK1**
DIA_B2	**1**	**xx**	**Value of register DIA_BNK2**
DIA_Z	**6**	**xx..**	**Value of register DIA_Zn**

.

30 - DC/DC control

This command activates or deactivates the DC/DC converter.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	DA	30	01	...

Parameters

parameters to select a SPI controller to the device

5	+0	+1	+2	+3	+4	+5	+6	+7
(1*)	**POMT**	**PIMT**	**POMR**	**PIMR**	**POCK**	**PICK**	**POSL**	**PISL**

parameters to configure the device

13	+0	+1
(1*)	**CTRL**	**WTIME**

Details

parameters to select a SPI controller to the device
Port and Pin parameters to select the SPI controller where the device is connected

Name	Length	Fmt	Description
POMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit Slave Receive) signal
PIMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within POMT of MTSR (Master Transmit Slave Receive) (SI) signal
POMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive Slave Transmit) signal
PIMR	**1**	**xx**	**Pin number of MRST signal** Pin number within POMR of MRST (Master Receive Slave Transmit) (SO) signal
POCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PICK	**1**	**xx**	**Pin number of SCK signal** Pin number within POCK of SCK (Serial Clock) signal
POSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PISL	**1**	**xx**	**Pin number of SLSO signal** Pin number within POSL of SLSO (Slave Select) signal

parameters to configure the device

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>CTRL</td><td>1</td><td>xx</td><td>Control DC/DC converter
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>15:1</td><td>not used, must be set to 0</td></tr>
<tr><td>0</td><td>Select status of DC/DC converter
0_B: Off
1_B: On</td></tr>
</table>
</td></tr>
<tr><td>WTIME</td><td>2</td><td>xxxx</td><td>Wait time after DC/DC activation/deactivation (Format: RVVV)</td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	DA	xx	xx	01

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)

Code	Description
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0xF0	**No SPI controller on given ports found.**
0xF2	**SPI communication/protocol failed**
0xF3	**SPI communication/protocol failed (FIX)**

TSW_DF - Event Control

Test step description

Date: 22.02.2016

Created: 18.01.2017 09:12.

This TSW provides a generic event handling control, mainly used for injection devices.

Test step commands

.

01 - Define controller output pins

This command defines up to 18 gpio output pins (controller view) used to control a device via direct setting.

Note: The used pins have to be initialized with TSW_04 (GPIO, Output).

Note: An ascending number is assigned to each defined pin, starting from 0. This number is used for parameter PIN in command 10.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	DF	01	01	...

Parameters

Parameter set for define controller output pins

5	+0	+1
(1..n*)	**PO**	**PI**

Details

Parameter set for define controller output pins

Name	Length	Fmt	Description
PO	**1**	**xx**	**Port number**
PI	**1**	**xx**	**Pin number inside port**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	DF	xx	xx	01	...

Parameters

Response set for define controller output pins

... (1..n*)	+0
	NPINS

Details

Response set for define controller output pins

Name	Length	Fmt	Description
NPINS	**1**	**xx**	**number of defined pins**

.

03 - Define controller pulse output pins

This command defines up to 4 output pins (controller view) used to control a device via pulse output.

Note: The used pins have to be initialized with TSW_04 (PWM, Output).

Note: An ascending number is assigned to each defined pin, starting from 0. This number is used for parameter PIN in command 10.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	DF	03	01	...

Parameters

Parameter set for define controller pulse output pins

5 (1..n*)	+0	+1	+2	+3	+4
	PO	**PI**	**CFG**	**LVL**	**MAXTIME**

Details

Parameter set for define controller pulse output pins

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>PO</td><td>1</td><td>xx</td><td>Port number</td></tr>
<tr><td>PI</td><td>1</td><td>xx</td><td>Pin number inside port</td></tr>
<tr><td>CFG</td><td>1</td><td>xx</td><td>Configuration
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:5</td><td>not used, must be set to 0</td></tr>
<tr><td>4</td><td>Select GTM TOM or ATOM
0_B: use assigned TOM
1_B: use assigned ATOM</td></tr>
<tr><td>3:1</td><td>not used, must be set to 0</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td><table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>0</td><td>Select GTM TOM/ATOM
0_B: first assigned TOM/ATOM
1_B: second assigned TOM/ATOM</td></tr></table></td></tr>
<tr><td>LVL</td><td>1</td><td>xx</td><td>Level of pulse<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>7:1</td><td>not used, must be set to 0</td></tr><tr><td>0</td><td>Select level of pulse
0_B: Low
1_B: high</td></tr></table></td></tr>
<tr><td>MAXTIME</td><td>2</td><td>xxxx</td><td>Pulse time of longest pulse
Time for the longest low/high pulse. Format: RVVV (see table 1). This value is only used to calculate the resolution.</td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	DF	xx	xx	01	...

Parameters

Response set for define controller output pins

6 (1..n*)	+0	+1
	MOD	**CH**

Details

Response set for define controller output pins

Name	Length	Fmt	Description
MOD	**1**	**xx**	**Module** used TOM/ATOM submodule for pulse output. 0xFF if no module found.
CH	**1**	**xx**	**Channel** used channel of TOM submodule. 0xFF if no module found.

.

04 - Define controller pulse input pins

This command defines up to 4 input pins (controller view) used to measure a device pulse.

Note: The used pins have to be initialized with TSW_04 (PWM, Input).

Note: An ascending number is assigned to each defined pin, starting from 0. This number is used for parameter PIN in command 10.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	DF	04	01	...

Parameters

Parameter set for define controller pulse input pins

5	+0	+1	+2	+3
(1..n*)	**PO**	**PI**	**CFG**	**LVL**

Details

Parameter set for define controller pulse input pins

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>PO</td><td>1</td><td>xx</td><td>Port number</td></tr>
<tr><td>PI</td><td>1</td><td>xx</td><td>Pin number inside port</td></tr>
<tr><td>CFG</td><td>1</td><td>xx</td><td>Configuration
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:1</td><td>not used, must be set to 0</td></tr>
<tr><td>0</td><td>Select GTM TIM

0_B: first assigned TIM
1_B: second assigned TIM</td></tr>
</table></td></tr>
<tr><td>LVL</td><td>1</td><td>xx</td><td>Level of pulse
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:1</td><td>not used, must be set to 0</td></tr>
<tr><td>0</td><td>Select level of pulse

0_B: Low
1_B: high</td></tr>
</table></td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	DF	xx	xx	01	...

Parameters

Response set for define controller input pins

6 (1..n*)	+0	+1
	MOD	**CH**

Details

Response set for define controller input pins

Name	Length	Fmt	Description
MOD	**1**	**xx**	**Module** used TIM submodule for busy time measurement. 0xFF if no module found.
CH	**1**	**xx**	**Channel** used channel of TIM submodule. 0xFF if no module found.

.

05 - Define device spi pins

This command defines 4 controller pins used for spi communication. Up to 2 devices could be defined.

Note: The used pins have to be initialized with TSW_04.

Note: An ascending number is assigned to each defined device, starting from 0. This number is used for parameter DEV in command 10.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	DF	05	01	...

Parameters

parameters to select a SPI controller for the device

5 (1..2*)	+0	+1	+2	+3
	POMT	**PIMT**	**POMR**	**PIMR**

... (1..2*)	+4	+5	+6	+7
	POCK	**PICK**	**POSL**	**PISL**

Details

parameters to select a SPI controller for the device
Port and Pin parameters to select the SPI controller where the device is connected

Name	Length	Fmt	Description
POMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit Slave Receive) signal
PIMT	**1**	**xx**	**Pin number of MTSR signal**

Name	Length	Fmt	Description
			Pin number within POMT of MTSR (Master Transmit Slave Receive) (SI) signal
POMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive Slave Transmit) signal
PIMR	**1**	**xx**	**Pin number of MRST signal** Pin number within POMR of MRST (Master Receive Slave Transmit) (SO) signal
POCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PICK	**1**	**xx**	**Pin number of SCK signal** Pin number within POCK of SCK (Serial Clock) signal
POSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PISL	**1**	**xx**	**Pin number of SLSO signal** Pin number within POSL of SLSO (Slave Select) signal

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	DF	xx	xx	01

.

06 - Define spi data

This command defines the spi data used in event SPI_CMD.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	DF	06	01	...

Parameters

parameters to define the spi data

5 (1*)	+0	+1
	NUM	**DATA**

Details

parameters to define the spi data

Name	Length	Fmt	Description
NUM	**1**	**xx**	**Burst number used in command 10 [0,15]**
DATA	**1**	**xx..**	**Burst data (up to 16)**

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	DF	xx	xx	01

.

07 - Define adc channels

This command defines the ADC channels for which measurement results will be provided after sequence. Description of parameter see start conversion and read result (command 10 of tsw 01)

Note: The used adc channels have to be configured with TSW_01. If this is not done, this command could cause a trap!

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	DF	07	01	...

Parameters

input pins for adc (repeatable up to eight inputs)

5	+0	+1
(1..n*)	**CHA**	**MUX**

Details

input pins for adc (repeatable up to eight inputs)

Name	Length	Fmt	Description
CHA	**1**	**xx**	**Index of the channel**
MUX	**1**	**xx**	**Channel of the external multiplexer**

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	DF	xx	xx	01

.

10 - Define sequence

This command defines a sequence of events, mainly used for the injection teststeps (D5, D6 and D8). Each event is processed in the order it appears in the command. The whole set of events is one cycle used in command 0x40.

Hint: the parameter PIN used in events SETPIN, PULSE, ACTPM und RDPM is the number given in commands 01, 03 and 04.

Hint: The event PULSE (0x50) is done by hardware. The sequencer is only triggering the start and will continue to process the next event. If this is not the wished behavior the WAIT (0x00) event could be used directly after the PULSE event.

Hint: It is not necessary to setup all events in one command line. If more lines a used, the following events are appended to the existing events. If this is not the wished behaviour the event RSTTABL (0xFF) could be used to clear the event queue.

Table 1. Event summary

Event	Id	Parameter
WAIT	0x00	TIME
SETPIN (low)	0x40	PIN
SETPIN (high)	0x41	PIN
PULSE	0x50	PIN, TIME
ACTPM	0x61	PIN
RDPM	0x60	PIN
SPI_CMD	0x70	DEV, CMD_NUM
TRGADC	0x81	-
READADC	0x80	-
MXADC	0x82	TIME
DIS_INT	0xF0	-
ENB_INT	0xF1	-
RSTTABL	0xFF	-

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	DF	10	01	...

Parameters

Parameter set for Wait event

5	+0	+1
(1..n*)	**WAIT**	**TIME**

Parameter set for set output pin event

8	+0	+1
(1..n*)	**SETPIN**	**PIN**

Parameter set for set pulse event

10	+0	+1	+2
(1..n*)	**PULSE**	**PIN**	**TIME**

Parameter set for activate pulse measurement event

14	+0	+1
(1..n*)	**ACTPM**	**PIN**

Parameter set for read pulse measurement event

16	+0	+1
(1..n*)	**RDPM**	**PIN**

Parameter set for spi cmd event

18	+0	+1	+2
(1..n*)	**SPI_CMD**	**DEV**	**CMD_NUM**

Parameter set for trigger ADC event

21	+0
(1..n*)	**TRGADC**

Parameter set for read ADC result event

22	+0
(1..n*)	**READADC**

Parameter set for Min/Max ADC event

23	+0	+1
(1..n*)	**MXADC**	**TIME**

Parameter set for disable WD interrupt

26	+0
(1..n*)	**DIS_INT**

Parameter set for enable WD interrupt

27	+0
(1..n*)	**ENB_INT**

Parameter set for reset event table

28	+0
(1*)	**RSTTABL**

Details

Parameter set for Wait event
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
WAIT	**1**	**xx**	**Event type wait** In this case "0x00" for WAIT
TIME	**2**	**xxxx**	**Wait time**

Name	Length	Fmt	Description
			Wait time before the start of the next event. Format: RVVV (see table 1)

Parameter set for set output pin event
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
SETPIN	**1**	**xx**	**Event type set output pin** In this case "0x40"/"0x41" for SETPIN low/high
PIN	**1**	**xx**	**pin number** pin number of the output pin to set.

Parameter set for set pulse event
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
PULSE	**1**	**xx**	**Event type set pulse** In this case "0x50" for PULSE
PIN	**1**	**xx**	**pin number** pin number of the output pin to set.
TIME	**2**	**xxxx**	**Pulse time** Time for the low/high pulse. Format: RVVV (see table 1)

Parameter set for activate pulse measurement event
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
ACTPM	**1**	**xx**	**Event type activate pulse measurement** In this case "0x61" for ACTPM
PIN	**1**	**xx**	**pin number** pin number of the input pin to measure.

Parameter set for read pulse measurement event
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
RDPM	**1**	**xx**	**Event type read pulse measurement** In this case "0x60" for RDPM
PIN	**1**	**xx**	**pin number** pin number of the input pin to measure.

Parameter set for spi cmd event
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
SPI_CMD	**1**	**xx**	**Event type spi cmd** In this case "0x70" for SPI_CMD
DEV	**1**	**xx**	**device number** device [0,1] number of the device.
CMD_NUM	**1**	**xx**	**command number** command [0,15] number of the command.

Parameter set for trigger ADC event
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
TRGADC	**1**	**xx**	**Event type trigger adc** In this case "0x81" for TRGADC (start ADC)

Parameter set for read ADC result event
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
READADC	**1**	**xx**	**Event type read adc** In this case "0x80" for READADC

Parameter set for Min/Max ADC event
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
MXADC	**1**	**xx**	**Event type min/max adc** In this case "0x82" for MXADC
TIME	**2**	**xxxx**	**Measurement time** Measurement time for min/max ADC. Format: RVVV (see table 1)

Parameter set for disable WD interrupt
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
DIS_INT	**1**	**xx**	**Event type disable WD trigger** In this case "0xF0" for DISABLE INT

Parameter set for enable WD interrupt
Multiple parameter sets may be appended to create a complete sequence of events

Name	Length	Fmt	Description
ENB_INT	**1**	**xx**	**Event type enable WD trigger** In this case "0xF1" for ENABLE INT

Parameter set for reset event table

Name	Length	Fmt	Description
RSTTABL	**1**	**xx**	**Event type reset sequence table** In this case "0xFF" for RESET TABLE

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	DF	xx	xx	01	...

Parameters

Response of command

...	+0
(1*)	**EVENTS**

Details

Response of command

Name	Length	Fmt	Description
EVENTS	**1**	**xx**	**Number of defined events** Number of totally defined events

.

40 - Run sequence cycles

This command runs the sequence cycles which has to be configured by command 0x10.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	DF	40	01	...

Parameters

parameters for command

...	+0
(1*)	**CYCLES**

Details

parameters for command

Name	Length	Fmt	Description
CYCLES	**4**	**xxxxxxxx**	**Number of injection cycles** This parameter sets the number of subsequent sequence cycles. A zero value means endless.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	DF	xx	xx	01	...

Parameters

repeatable pulse length measurement respones (if CYCLES > 0)

...	+0
(0..n*)	**PLMIN**

...	+2
(0..n*)	**PLMAX**

repeatable ADC measurement results (if CYCLES > 0)

...	+0
(0..n*)	**ADV**

repeatable SPI respones (if CYCLES > 0)

...	+0
(0..n*)	**BDATA**

Details

repeatable pulse length measurement respones (if CYCLES > 0)

Name	Length	Fmt	Description
PLMIN	**2**	**xxxx**	**minimal measured pulse length (if configured)**
PLMAX	**2**	**xxxx**	**maximal measured pulse length (if configured)**

repeatable ADC measurement results (if CYCLES > 0)
For each READADC event all configured adc channels are reported in the order they appear in the define adc channels command (0x07). This is repeated for all READADC events.

Name	Length	Fmt	Description
ADV	**2**	**xxxx**	**ADC measurement result value (if configured and used)**

repeatable SPI respones (if CYCLES > 0)

Name	Length	Fmt	Description
BDATA	**1**	**xx..**	**answer of last spi burst command (if configured and used)**

.

80 - Read sequence results

This command is used to read out results after a run with command 0x40.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	DF	80	01	...

Parameters

Parameter set for Read sequence results

...	+0
(1*)	**CFG**

Details

Parameter set for Read sequence results

Name	Length	Fmt	Description
CFG	**2**	**xxxx**	**General configuration** **Bit(s)** / **Description** **15:0** / **Reserved for future, must be zero**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	DF	xx	xx	01	...

Parameters

Response set for Read sequence results

6 (1*)	+0	+2
	MPNTS	**NCHN**

Repeatable ADC min/max measurement results:

9 (0..n*)	+0	+2
	MIN	**MAX**

Details

Response set for Read sequence results

Name	Length	Fmt	Description
MPNTS	**2**	**xxxx**	**number of ADC min/max meaurements done**
NCHN	**1**	**xx**	**number of defined ADC channels**

Repeatable ADC min/max measurement results:
If ADCMinMaxEvent is used, for each configured ADC channel the minimal and maximal values are reported.

Name	Length	Fmt	Description
MIN	**2**	**xxxx**	**ADC measurement min value**
MAX	**2**	**xxxx**	**ADC measurement max value**

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**

Code	Description
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0xF0	**No SPI controller on given ports found.**
0xF1	**HAL parameter error**
0xF2	**SPI communication/protocol failed**
0xF6	**some defines missing**
0xF7	**no more events possible**
0xF8	**no more resoures available**
0xF9	**invalid configuration**

TSW_E3 - InfoBlock check

Test step description

This test step checks the software consistency of the ASW in flash (checksums,...) and can be called with several modes.

Test step commands

.

01 - Check infoblocks

This command check the infoblocks.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	E3	01	01

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	E3	xx	xx	01	...

Parameters

Response set for check infoblocks request

6	+0	+4	+5	+6
(1*)	**BSC**	**BCNT**	**TTNR**	**SWID**

Details

Response set for check infoblocks request

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>BSC</td><td>4</td><td>xxxxxxxx</td><td>Block status code

Note
If all infoblocks and checksums are OK, blockstatus code is 0x00000000
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>0</td><td>1: Check failed: if the LinkChain is in an endless loop</td></tr>
<tr><td>1</td><td>1: Check failed: value of Prolog CRC is equal to saved value from HexModX</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>2</td><td>1: Check failed: value of valid pattern</td></tr>
<tr><td>3</td><td>1: Check failed: value of all checksum blocks are equal to saved values from HexModX</td></tr>
<tr><td>4</td><td>1: Check failed: compatibility values are equal between linked blocks</td></tr>
<tr><td>5</td><td>1: Check failed: address in compatibility link is pointing to existing flash address</td></tr>
<tr><td>6</td><td>1: Check failed: addresses in checksum blocks are pointing to existing flash addresses</td></tr>
<tr><td>7</td><td>1: Check failed: address in link chain is pointing to existing flash address</td></tr>
<tr><td>8</td><td>1: Check failed: number of checksum entries is always bigger then 0</td></tr>
<tr><td>9</td><td>1: Check failed: configured value for checksum algorithm is supported</td></tr>
<tr><td>10</td><td>1: Check failed: area of prolog lays in existing flash address used by valid pattern check</td></tr>
<tr><td>11</td><td>1: Check failed: area of prolog lays in existing flash address used by CRC</td></tr>
<tr><td>12</td><td>1: Check failed: calculated end of prolog is smaller as start address of corresponding epilog</td></tr>
<tr><td>13</td><td>1: Check failed: ECC for Prolog and Epilog from a section</td></tr>
<tr><td>28:14</td><td>not used, always set to 0</td></tr>
<tr><td>29</td><td>1: Check failed: SWID not available at least in one block</td></tr>
<tr><td>30</td><td>1: Check failed: SWID is different at least in one block</td></tr>
<tr><td>31</td><td>1: Check failed: TTNR is different at least in one block</td></tr>
</table>
</td></tr>
<tr><td>BCNT</td><td>1</td><td>xx</td><td>Blockcount
Number of found and valid blocks.</td></tr>
<tr><td>TTNR</td><td>1</td><td>xx..</td><td>TTNR [10 Bytes]</td></tr>
<tr><td>SWID</td><td>1</td><td>xx..</td><td>SW ID [10 Bytes]

Note
If no block contains a valid SWID, 0x00 will be reported for all Bytes</td></tr>
</table>

.

02 - Read infoblock data

This command reads data from infoblock.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	E3	02	01	...

Parameters

Parameter set for read data from infoblock request

5	+0
(1*)	**BNR**

Details

Parameter set for read data from infoblock request

Name	Length	Fmt	Description
BNR	**1**	**xx**	**Block number**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	E3	xx	xx	01	...

Parameters

Response set for read data from infoblock request

6	+0	+4
(1*)	**BSC**	**BCNT**

Response set for read data from infoblock request

11	+0
(1*)	**BDATA**

Details

Response set for read data from infoblock request

Name	Length	Fmt	Description
BSC	**4**	**xxxxxxxx**	**Block status code** Detailed description see response description of command 0x01
BCNT	**1**	**xx**	**Blockcount** Number of found and valid blocks.

Response set for read data from infoblock request

Name	Length	Fmt	Description
BDATA	**80**	**xx..**	**Block Data** Infoblock prolog data WITHOUT first 256 Byte (ValidPattern)

.

03 - Read checksum table data

This command reads data from checksum table.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	E3	03	01	...

Parameters

Parameter set for read data from checksum table request

5	+0	+1
(1*)	**BNR**	**CSNR**

Details

Parameter set for read data from checksum table request

Name	Length	Fmt	Description
BNR	**1**	**xx**	**Block number**
CSNR	**1**	**xx**	**Checksum table number**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	E3	xx	xx	01	...

Parameters

Response set for read data from checksum table request

6	+0	+4
(1*)	**BSC**	**BCNT**

Response set for read data from checksum table request

11	+0
(1*)	**CSDATA**

Details

Response set for read data from checksum table request

Name	Length	Fmt	Description
BSC	**4**	**xxxxxxxx**	**Block status code** Detailed description see response description of command 0x01
BCNT	**1**	**xx**	**Blockcount** Number of found and valid blocks.

Response set for read data from checksum table request

Name	Length	Fmt	Description
CSDATA	**16**	**xx..**	**Checksum table Data**

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)

Code	Description
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**TSW: Wrong Version of BootCtrl (must be >= 1.02.01)**
0x02	**TSW: Invalid BootModeHeader**
0x03	**TSW: BootCtrl in copypage (aborted programming)**
0x04	**TSW: Length of SWID longer than specified**
0x01	**HAL: Memlay error**
0xFF	**HAL: Memlay invalid parameters**
0xFE	**HAL: Memlay function not implemented**
0xFD	**HAL: Memlay info table block not found**
0xFA	**HAL: Memlay info table block address invalid**

TSW_E4 - Checksum and CRC calculation

Test step description

Date: 14.10.2013

This tsw calculates the checksum/CRC over a selectable area.

Test step commands

.

01 - Calculate CRC4 SMP48x

This algorithm calculates a CRC4 value for SMP48x.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	E4	01	01	...

Parameters

Parameters for CRC4 SMP48x calculation

5	+0	+1
(1*)	**BITS**	**DATA**

Details

Parameters for CRC4 SMP48x calculation

Name	Length	Fmt	Description
BITS	**1**	**xx**	**Number of bits** Number of Bits of the Value that should be used for calculation. 0x00 - 0x20 allowed.
DATA	**4**	**xxxxxxxx**	**Data for the CRC calculation** Value over which the CRC is calculated (right aligned).

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	E4	xx	xx	01	...

Parameters

Response of calculation

6	+0
(1*)	**CRC**

Details

Response of calculation

Name	Length	Fmt	Description
CRC	**4**	**xxxxxxxx**	**Calculated CRC4** Calculated CRC4. Use the last 4 Bits

.

02 - Calculate CRC16

This algorithm calculates a CRC16 over the selected range and responds the result as 32 bit value. The startvalue is always 0xFFFF.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	E4	02	01	...

Parameters

Parameters for CRC16 calculation

5	+0	+4	+8
(1*)	**GENPOL**	**STRADD**	**ENDADD**

Details

Parameters for CRC16 calculation

Name	Length	Fmt	Description
GENPOL	**4**	**xxxxxxxx**	**Generator polynom**
STRADD	**4**	**xxxxxxxx**	**Startaddress for the CRC calculation**
ENDADD	**4**	**xxxxxxxx**	**Endaddress for the CRC calculation**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	E4	xx	xx	01	...

Parameters

Response of calculation

6	+0
(1*)	**CRC**

Details

Response of calculation

Name	Length	Fmt	Description
CRC	4	xxxxxxxx	Calculated CRC16

.

03 - Calculate CRC16 CCITT

This algorithm calculates a CRC16 over the selected range and responds the result as 32 bit value. The startvalue is always 0x0.

Command

1	2	3	4	5
SVC	TSW	CMD	VER	...
80	E4	03	01	...

Parameters

Parameters for CRC16 CCITT calculation

5	+0	+4	+8
(1*)	GENPOL	STRADD	ENDADD

Details

Parameters for CRC16 CCITT calculation

Name	Length	Fmt	Description
GENPOL	4	xxxxxxxx	Generator polynom
STRADD	4	xxxxxxxx	Startaddress for the CRC calculation
ENDADD	4	xxxxxxxx	Endaddress for the CRC calculation

Response

1	2	3	4	5	6
SVC	TSW	ESVC	ETSW	VER	...
C0	E4	xx	xx	01	...

Parameters

Response of calculation

6	+0
(1*)	CRC

Details

Response of calculation

Name	Length	Fmt	Description
CRC	**4**	**xxxxxxxx**	**Calculated CRC16 CCITT**

.

04 - Calculate CRC16 RC30

This algorithm calculates a CRC16 over the selected range and responds the result as 32 bit value. Generatorpolynom: 0x11021.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	E4	04	01	...

Parameters

Parameters for CRC16 RC30 calculation

5	+0	+4	+8
(1*)	**STRVAL**	**STRADD**	**ENDADD**

Details

Parameters for CRC16 RC30 calculation

Name	Length	Fmt	Description
STRVAL	**4**	**xxxxxxxx**	**Startvalue**
STRADD	**4**	**xxxxxxxx**	**Startaddress for the CRC calculation**
ENDADD	**4**	**xxxxxxxx**	**Endaddress for the CRC calculation**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	E4	xx	xx	01	...

Parameters

Response of calculation

6	+0
(1*)	**CRC**

Details

Response of calculation

Name	Length	Fmt	Description
CRC	**4**	**xxxxxxxx**	**Calculated CRC16 RC30**

.

05 - Calculate CRC32

This algorithm calculates a CRC32 over the selected range and responds the result as 32 bit value. The default start value is always 0xFFFFFFFF. Generator polynom is 0xEDB88320.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	E4	05	01	...

Parameters

Parameters for CRC32 calculation

5	+0	+4	+8
(1*)	**STRVAL**	**STRADD**	**ENDADD**

Details

Parameters for CRC32 calculation

Name	Length	Fmt	Description
STRVAL	**4**	**xxxxxxxx**	**Startvalue**
STRADD	**4**	**xxxxxxxx**	**Startaddress for the CRC calculation**
ENDADD	**4**	**xxxxxxxx**	**Endaddress for the CRC calculation**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	E4	xx	xx	01	...

Parameters

Response of calculation

6	+0
(1*)	**CRC**

Details

Response of calculation

Name	Length	Fmt	Description
CRC	**4**	**xxxxxxxx**	**Calculated CRC32**

.

06 - Calculate CRC32REVH04C11DB7

This algorithm calculates a CRC32 over the selected range and responds the result as 32 bit value. Generator polynom is 0xEDB88320. Startvalue is 0xFFFFFFFF. Data need not to be aligned. The used algorithm is equivalent with the FSW CRC32REVH04C11DB7 algorithm.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	E4	06	01	...

Parameters

Parameters for CRC32 calculation

5	+0	+4	+8
(1*)	**STRVAL**	**STRADD**	**ENDADD**

Details

Parameters for CRC32 calculation

Name	Length	Fmt	Description
STRVAL	**4**	**xxxxxxxx**	**Startvalue** Not used!
STRADD	**4**	**xxxxxxxx**	**Startaddress for the CRC calculation**
ENDADD	**4**	**xxxxxxxx**	**Endaddress for the CRC calculation**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	E4	xx	xx	01	...

Parameters

Response of calculation

6	+0
(1*)	**CRC**

Details

Response of calculation

Name	Length	Fmt	Description
CRC	**4**	**xxxxxxxx**	**Calculated CRC32**

.

10 - Calculate 8bit checksum

This algorithm adds byte by byte over the selected range and responds the result as 32 bit value.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	E4	10	01	...

Parameters

Parameters for 8bit checksum calculation

5	+0	+4	+8
(1*)	**STRVAL**	**STRADD**	**ENDADD**

Details

Parameters for 8bit checksum calculation

Name	Length	Fmt	Description
STRVAL	**4**	**xxxxxxxx**	**Startvalue**
STRADD	**4**	**xxxxxxxx**	**Startaddress for the checksum calculation**
ENDADD	**4**	**xxxxxxxx**	**Endaddress for the checksum calculation**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	E4	xx	xx	01	...

Parameters

Response of calculation

6	+0
(1*)	**CHKSUM**

Details

Response of calculation

Name	Length	Fmt	Description
CHKSUM	**4**	**xxxxxxxx**	**Calculated 8bit checksum**

.

11 - Calculate 16bit checksum

This algorithm adds 16 bit values over the selected range and responds the result as 32 bit value.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	E4	11	01	...

Parameters

Parameters for 16bit checksum calculation

5	+0	+4	+8
(1*)	STRVAL	STRADD	ENDADD

Details

Parameters for 16bit checksum calculation

Name	Length	Fmt	Description
STRVAL	4	xxxxxxxx	Startvalue
STRADD	4	xxxxxxxx	Startaddress for the checksum calculation
ENDADD	4	xxxxxxxx	Endaddress for the checksum calculation

Response

1	2	3	4	5	6
SVC	TSW	ESVC	ETSW	VER	...
C0	E4	xx	xx	01	...

Parameters

Response of calculation

6	+0
(1*)	CHKSUM

Details

Response of calculation

Name	Length	Fmt	Description
CHKSUM	4	xxxxxxxx	Calculated 16bit checksum

.

12 - Calculate 32bit checksum

This algorithm adds 32 bit values over the selected range and responds the result as 32 bit value.

Command

1	2	3	4	5
SVC	TSW	CMD	VER	...
80	E4	12	01	...

Parameters

Parameters for 32bit checksum calculation

5	+0	+4	+8
(1*)	**STRVAL**	**STRADD**	**ENDADD**

Details

Parameters for 32bit checksum calculation

Name	Length	Fmt	Description
STRVAL	**4**	**xxxxxxxx**	**Startvalue**
STRADD	**4**	**xxxxxxxx**	**Startaddress for the checksum calculation**
ENDADD	**4**	**xxxxxxxx**	**Endaddress for the checksum calculation**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	E4	xx	xx	01	...

Parameters

Response of calculation

6	+0
(1*)	**CHKSUM**

Details

Response of calculation

Name	Length	Fmt	Description
CHKSUM	**4**	**xxxxxxxx**	**Calculated 32bit checksum**

.

20 - Blankcheck

This function checks, wheather the selected area is erased and responds 0xFFFFFFFF if all OK or the address of the first occurence, where the read value was not equal to blank pattern. Addresses have to be 4-byte aligned.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	E4	20	01	...

Parameters

Parameters for blankcheck

5	+0	+4
(1*)	**STRADD**	**ENDADD**

Details

Parameters for blankcheck

Name	Length	Fmt	Description
STRADD	**4**	**xxxxxxxx**	**Startaddress for the blankcheck calculation**
ENDADD	**4**	**xxxxxxxx**	**Endaddress for the blankcheck calculation**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	E4	xx	xx	01	...

Parameters

Response of check

6	+0
(1*)	**BLANK**

Details

Response of check

Name	Length	Fmt	Description
BLANK	**4**	**xxxxxxxx**	**Blankcheck** 0xFFFFFFFF in case of memory area is blank, else error address appears.

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)

Code	Description
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**TSW: address not 4-byte aligned**

TSW_E5 - Read SeedKey name

Test step description

Date: 29.08.2014

This teststep is used to read out the used SeedKey name from the ECU.

Note: not used for FSL target.

Test step commands

.

01 - Read SeedKey name

This command read out the SeedKey name in ASCII.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	E5	01	01

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	E5	xx	xx	01	...

Parameters

Response set for SeedKey name request

6	+0	+1
(1*)	**LEN**	**KEY**

Details

Response set for SeedKey name request

Name	Length	Fmt	Description
LEN	**1**	**xx**	**Length of SeedKey name request**
KEY	**1**	**xx..**	**SeedKey name** Name in ASCII characters. The chain of characters terminates with 0x00.

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**Infoblock chain interrupted**
0x02	**ECUSECUSRV not found within infoblock chain**
0x03	**TSW SeedKey name not found in ECUSECUSRV**

TSW_E6 - Read device specific information

Test step description

Date: 29.11.2016

This teststep is used to read out some specific information from the target system.

Test step commands

.

01 - Read device specific information

This command read out chip infrormation like identification, frequency, temperature.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	E6	01	01

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	E6	xx	xx	01	...

Parameters

Specific Chip information

6	+0	+2	+3	+4	+5	+6
(1*)	**ID**	**REV**	**MAN**	**FAM**	**TYP**	**DFRQ**

...	+8	+10	+12
(1*)	**PFRQ**	**TEMP**	**CAL**

Details

Specific Chip information

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>ID</td><td>2</td><td>xxxx</td><td>Chip Identification Number
The ID request response contains the value of the CHIP ID and device package type.
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>15:8</td><td>Not used</td></tr>
<tr><td>7:0</td><td>Device type and package
70_H: TC27x LQFP176 - Device 3</td></tr>
</table>
</td></tr>
</table>

Name	Length	Fmt	Description
			Bit(s) / **Description** **71_H:** TC27x BGA292 - Device 3 **90_H:** TC29x BGA292 - Device 4 **91_H:** TC29x BGA416 - Device 4 **92_H:** TC29x BGA516 - Device 4 **75_H:** TC37x LQFP176 - Device 3+ **77_H:** TC37x BGA292 - Device 3+ **87_H:** TC38x BGA292 - Device 4+ **89_H:** TC38x BGA516 - Device 4+ **97_H:** TC39x BGA292 - Device 5+ **99_H:** TC39x BGA516 - Device 5+
REV	**1**	**xx**	**Chip Revision Number** **Bit(s)** / **Description** **7:4** — **0_H:** Axx-Step; **1_H:** Bxx-Step; **2_H:** Cxx-Step **3:0** — **0_H:** x11-Step; **1_H:** x12-Step
MAN	**1**	**xx**	**Chip Manufacturer** **$C1_H$:** Infineon
FAM	**1**	**xx**	**Chip Family** **01_H:** Aurix Gen1 (TC2xx) **02_H:** Aurix Gen2 (TC3xx) ASCII character in MCU Part Number
TYP	**1**	**xx**	**Device Type** **0_H:** Production device **1_H:** Emulation device
DFRQ	**2**	**xxxx**	**Device Clock Frequency** Device clock frequency in MHz.
PFRQ	**2**	**xxxx**	**PLL Clock Frequency** PLL clock frequency in MHz.
TEMP	**2**	**xxxx**	**Device Temperature** Value of device on-chip temperature sensor in °C.
CAL	**1**	**xx**	**Sensor Calibration** not used.

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0xEF	**HAL_FLS: Locking or unlocking of flash failed**

TSW_E7 - Reprog signature file validity check

Test step description

Test step commands

.

01 - Download signature

Can only be used for IFX and JDP.

This command is used to download a signature file from a FSW container.

For example (CFITerm):

LOADFILE signature.hex,,,255,80E70101,C0E7000001,00,last

downloads the whole file signature.hex.

Note

This command should NOT be used to load more than one set of signatures! In this case only the first loaded signature will be used.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	E7	01	01	...

Parameters

Signature data

5 (1*)	+0 **SDATA**

Details

Signature data

Name	Length	Fmt	Description
SDATA	**1**	**xx**	**Signature data**

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	E7	xx	xx	01

.

02 - Download certificate

Can only be used for IFX and JDP.

This command is used to download a certificate file prior to flash programming. This is mandatory, if tuning protection is activated and the certificate is availble at FSW container.

Each time a certificate is loaded, the one before is erased. For next signature check (finish) the new certificate is used.

Note

Certificates bigger than 1 KB will be rejected.

Certificate usage:

If no certificate is (successfully) loaded, existing certificate from Epilog of ECUSEcuSrv will be used.

For example (CFITerm):

LOADFILE certificate.hex,,,255,80E70201,C0E7000001,00,last

downloads the whole file certificate.hex.

Errors are reported only once. Download procedure is reset automatical.

Without data the command returns state of certificate loading.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	E7	02	01	...

Parameters

Certificate

5	+0
(1*)	**SDATA**

Details

Certificate

Name	Length	Fmt	Description
SDATA	**1**	**xx..**	**Certificate data**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	E7	xx	xx	01	...

Parameters

Response of command "Download certificate"

6	+0
(1*)	**DLCEST**

Details

Response of command "Download certificate"

Name	Length	Fmt	Description
DLCEST	**1**	**xx**	**download state of certificate** **00_H:** No certificate loaded **01_H:** Loading **02_H:** Certificate completely loaded, ready to use

.

03 - Validity check of loaded reprog signatures

Can only be used for IFX and JDP.

This command is used to do a validity check of loaded reprog signatures.

Note

If ECU contains OTP areas and a reprogramming was done, validity check may fail at position of first OTP area in signature block!

Note

If an error occurs (SSTAT != 0x1) after execution of this command, further executions will not lead to new results!

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	E7	03	01

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	E7	xx	xx	01	...

Parameters

Response set for check infoblocks request

6	+0	+1	+2	+3	+7
(1*)	**TP**	**SSTAT**	**SFIND**	**SADR**	**EADR**

Details

Response set for check infoblocks request

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>TP</td><td>1</td><td>xx</td><td>TP status

0x00 - no tuningprotection / HSM not active
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:5</td><td>not used</td></tr>
<tr><td>4</td><td>1: EcuSecu is activ</td></tr>
<tr><td>3</td><td>1: HSM flash protections active</td></tr>
<tr><td>2</td><td>1: HSM is activ</td></tr>
<tr><td>1</td><td>1: PFlash protections active</td></tr>
<tr><td>0</td><td>1: RBSN set</td></tr>
</table></td></tr>
<tr><td>SSTAT</td><td>1</td><td>xx</td><td>Signatur verification state

01_H: EcuSecu Ok
10_H: needed EcuSecu function not available
40_H: HSM state is not active
41_H: EcuSecu state is not active
42_H: SSM projects: address information for PAV area not available
50_H: PKCS11: init error
51_H: PKCS11: open session error
52_H: PKCS11: create obj. error (certificate)
53_H: PKCS11: verify init error
54_H: PKCS11: verify error
55_H: PKCS11: digest init
56_H: PKCS11: digest update
57_H: PKCS11: digest final
58_H: PKCS11: signature invalid
FF_H: EcuSecuSrv not found or application ECU (SSM)

Note for PKCS11: create obj. error
Used certificate can cause an "create obj. error" if it not belong to the project.</td></tr>
<tr><td>SFIND</td><td>1</td><td>xx</td><td>Index of failed signature block
Signature block counting starts at 0.

Note
If no error occured (SSTAT = 0x1) total amount of checked signature blocks will be reported.</td></tr>
<tr><td>SADR</td><td>4</td><td>xxxxxxxx</td><td>Start address of failed HASH range</td></tr>
</table>

Name	Length	Fmt	Description
			Note If no error occured (SSTAT = 0x1) 0x00000000 will be reported.
EADR	**4**	**xxxxxxxx**	**End address of failed HASH range** **Note** If no error occured (SSTAT = 0x1) 0x00000000 will be reported.

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**TSW: Wrong Version of BootCtrl (must be >= 1.02.01)**
0x02	**TSW: Invalid signature length**
0x03	**TSW: ECUSECU or HSM is not running or not found**
0x04	**TSW: Certificate not fully loaded**
0x05	**TSW: Certificate invalid. Internal ID not correct**
0x06	**TSW: Certificate to long. Internal length greater 1 KB**
0x07	**TSW: HEX file of certificate to long. Length greater than internal length information or buffer limit reached**
0x08	**TSW: HEX file of certificate with unknown length information. Existing certificate from Epilog of ECUSEcuSrv will be used.**

TSW_EA - Eval. serial EEPROM

Test step description

Purpose: This TSW provides evaluation of serial EEPROM device.

Important: before using, initializing of some other hardware resources is required.

Required test steps

tsw_02: configuration of SPI communication.

tsw_04: configuration of port/pin SPI signals.

This TSW is used to read from/write to a serial EEPROM which is connected by SPI. Only EEPROMs with 16bit addressfield in the write or read command are supported.

The serial EEPROM is connected to the system via several hardware lines for clock, reset and communication lines to the controller via SPI/SSC bus connection. All data values and setup data are transfered through this SPI/SSC bus.

For a detailed description of serial EEPROM, please refer to M95064 (ST) component documentation/data sheet.

The serial EEPROM provides a set of SPI read and write commands accessing to data of the device. A table containing all SPI instructions is shown below

Table 1. SPI instructions to access EEPROM data and status register

SPI instruction	Instruction code	Description
WRITE_STATUS	01h	Write status register
WRITE_DATA	02h	Write data to device
READ_DATA	03h	Read data from device
WRITE_DISABLE	04h	Disable write access to device
READ_STATUS	05h	Read status register
WRITE_ENABLE	06h	Enable write access to device

Test step commands

.

01 - Read EEPROM data

This command supports reading a number of data bytes from a given start address from the serial EEPROM device

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	EA	01	01	...

Parameters

Read EEPROM data via SPI bus

5	+0	+1	+2	+3	+4	+5
(1*)	**PORTMT**	**PINMT**	**PORTMR**	**PINMR**	**PORTCK**	**PINCK**

...	+6	+7
(1*)	**PORTSL**	**PINSL**

Start address of EEPROM and number of data bytes to be read from

13	+0	+2
(1*)	**EEADR**	**EENUM**

Details

Read EEPROM data via SPI bus
Port and Pin parameters to select the SPI controller where serial EEPROM is connected

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

Start address of EEPROM and number of data bytes to be read from

Name	Length	Fmt	Description
EEADR	**2**	**xxxx**	**Address EEPROM** First address to be read from EEPROM **Note:** The range of EEPROM address is theoretically 0000..FFFFh. The typical max. address for used device is 1FFFh
EENUM	**1**	**xx**	**Number of data bytes** Number of bytes to be read from EEPROM

Name	Length	Fmt	Description
			Note: The range for number of data bytes is 1..250dez (01..FAh)

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	EA	xx	xx	01	...

Parameters

Response of read EEPROM data

6 (1..n*)	+0
	EEDAT

Details

Response of read EEPROM data

Name	Length	Fmt	Description
EEDAT	**1**	**xx**	**1st data byte from EEPROM** The range for responded number of data bytes is 1..250dez (01..FAh)If operation was successful, number of data bytes from EEPROM responded...

.

02 - Write EEPROM data

This command supports writing a number of data bytes from a given start address to the serial EEPROM device

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	EA	02	01	...

Parameters

Write EEPROM data via SPI bus

5 (1*)	+0	+1	+2	+3	+4	+5
	PORTMT	**PINMT**	**PORTMR**	**PINMR**	**PORTCK**	**PINCK**

... (1*)	+6	+7
	PORTSL	**PINSL**

Bloc protection option and start address of EEPROM to be write to

13 (1*)	+0	+1
	BLOCP	**EEADR**

Data bytes

16	+0
(1..n*)	**EEDAT**

Details

Write EEPROM data via SPI bus
Port and Pin parameters to select the SPI controller where serial EEPROM is connected

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

Bloc protection option and start address of EEPROM to be write to

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>BLOCP</td><td>1</td><td>xx</td><td>Bloc protection
Removing of bloc protection automatically

Note
Please refer to data sheet of the device, to understand how does block protection working on the EEPROM

<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:0</td><td>Automatic removing of block protection

00H: without change of block protection
XXH: block protection will be removed before writing to EEPROM</td></tr>
</table>
</td></tr>
</table>

Name	Length	Fmt	Description
EEADR	**2**	**xxxx**	**Address EEPROM** First address to be write to EEPROM

Data bytes

Name	Length	Fmt	Description
EEDAT	**1**	**xx**	**1st data byte to be write..** ..further data bytes to be written to EEPROM **Note** The maximum number of data bytes is 250dez (FAh)

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	EA	xx	xx	01	...

Parameters

Response of write EEPROM data

6	+0	+2	+3
(1*)	**RADDR**	**WRDAT**	**RDDAT**

Details

Response of write EEPROM data

Name	Length	Fmt	Description
RADDR	**2**	**xxxx**	**Write/read verify error address** Value shows the EEPROM address where a write error occured. Value is FFFFh if no error has been detected.
WRDAT	**1**	**xx**	**Value to be written** In case of a write verify error, byte reflects the value which should be written, otherwise it is FFh.
RDDAT	**1**	**xx**	**Read value** In case of a write verify error, byte reflects the value which is read from error address, otherwise it is FFh.

.

03 - Check EEPROM device

This command performs a pattern check. Each EEPROM cell is written once and read back. If written value is not equal to the pattern, an error is reported. The test pattern looks like FBh,FAh,...,01h,FBh,FAh,F9h,... etc. which avoids same values on every 2n boundery.

Note

Because whole EEPROM is written the EEPROM content is destroyed!

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	EA	03	01	...

Parameters

Check EEPROM via SPI bus

5 (1*)	+0	+1	+2	+3	+4	+5
	PORTMT	**PINMT**	**PORTMR**	**PINMR**	**PORTCK**	**PINCK**

... (1*)	+6	+7
	PORTSL	**PINSL**

Bloc protection option and size of EEPROM to be check

13 (1*)	+0	+1
	BLOCP	**SIZE**

Details

Check EEPROM via SPI bus
Port and Pin parameters to select the SPI controller where serial EEPROM is connected

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

Bloc protection option and size of EEPROM to be check

Name	Length	Fmt	Description
BLOCP	**1**	**xx**	**Bloc protection** Removing of bloc protection automatically **Bit(s)** – **Description** **7:0** – **Automatic removing of block protection** **00_H:** without change of block protection **XX_H:** block protection will be removed before writing to EEPROM
SIZE	**2**	**xxxx**	**Size of EEPROM** Size of EEPROM to be checked **Note** The range of EEPROM size is theoretically 0001h..FFFFh. The typical value for size for used device is 2000h

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	EA	xx	xx	01	...

Parameters

Response of check EEPROM data

6	+0	+2	+3
(1*)	**RADDR**	**WRDAT**	**RDDAT**

Details

Response of check EEPROM data

Name	Length	Fmt	Description
RADDR	**2**	**xxxx**	**Write/read verify error address** Value shows the EEPROM address where a write error occured. Value is FFFFh if no error has been detected.
WRDAT	**1**	**xx**	**Value to be written** In case of a write verify error, byte reflects the value which should be written, otherwise it is FFh.
RDDAT	**1**	**xx**	**Read value** In case of a write verify error, byte reflects the value which is read from error address, otherwise it is FFh.

.

04 - Size of EEPROM device

This command evaluates the EEPROM size by writing address FFFFh with a check pattern an reading the last byte of all pages with smaller address. The EEPROM size is then the address of the page with different check byte value + 32 byte. If the responded EEPROM size is 0000h, it indicates that an EEPROM with 8bit address only is connected which is not supported.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	EA	04	01	...

Parameters

Get size of EEPROM via SPI bus

5 (1*)	+0	+1	+2	+3	+4	+5
	PORTMT	**PINMT**	**PORTMR**	**PINMR**	**PORTCK**	**PINCK**

... (1*)	+6	+7
	PORTSL	**PINSL**

Bloc protection option

13 (1*)	+0
	BLOCP

Details

Get size of EEPROM via SPI bus
Port and Pin parameters to select the SPI controller where serial EEPROM is connected

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal**

Name	Length	Fmt	Description
			Port number of SLSO (Slave Select) signal
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

Bloc protection option

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>BLOCP</td><td>1</td><td>xx</td><td>Remove bloc protection
Removing of bloc protection automatically
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:0</td><td>Automatic removing of block protection
00_H: without change of block protection
XX_H: block protection will be removed before writing to EEPROM</td></tr>
</table>
</td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	EA	xx	xx	01	...

Parameters

Size of EEPROM

6	+0
(1*)	**SIZE**

Details

Size of EEPROM

Name	Length	Fmt	Description
SIZE	**2**	**xxxx**	**Get size of EEPROM** If operation was successful, size of detected EEPROM responded...The range of EEPROM sizes could be 0400h..8000h (1..32kB). The typical value is 2000h (8kB). If the responded value=0000h, no EEPROM was detected

.

05 - Fill EEPROM device

This command allows filling data pattern into serial EEPROM device.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	EA	05	01	...

Parameters

Fill EEPROM via SPI bus

5	+0	+1	+2	+3	+4	+5
(1*)	**PORTMT**	**PINMT**	**PORTMR**	**PINMR**	**PORTCK**	**PINCK**

...	+6	+7
(1*)	**PORTSL**	**PINSL**

Bloc protection option, EEPROM size, fill pattern

13	+0	+1	+3
(1*)	**BLOCP**	**SIZE**	**DATA**

Details

Fill EEPROM via SPI bus
Port and Pin parameters to select the SPI controller where serial EEPROM is connected

Name	Length	Fmt	Description
PORTMT	**1**	**xx**	**Port number of MTSR signal** Port number of MTSR (Master Transmit) signal
PINMT	**1**	**xx**	**Pin number of MTSR signal** Pin number within PORTMT of MTSR (Master Transmit) signal
PORTMR	**1**	**xx**	**Port number of MRST signal** Port number of MRST (Master Receive) signal
PINMR	**1**	**xx**	**Pin number of MRST signal** Pin number withing PORTMR of MRST (Master Receive) signal
PORTCK	**1**	**xx**	**Port number of SCK signal** Port number of SCK (Serial Clock) signal
PINCK	**1**	**xx**	**Pin number of SCK signal** Pin number within PORTCK of SCK (Serial Clock) signal
PORTSL	**1**	**xx**	**Port number of SLSO signal** Port number of SLSO (Slave Select) signal
PINSL	**1**	**xx**	**Pin number of SLSO signal** Pin number within PORTSL of SLSO (Slave Select) signal

Bloc protection option, EEPROM size, fill pattern

Name	Length	Fmt	Description
BLOCP	**1**	**xx**	**Bloc protection**

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>Removing of bloc protection automatically
<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>7:0</td><td>Automatic removing of block protection
00_H: without change of block protection
XX_H: block protection will be removed before writing to EEPROM</td></tr></table></td></tr>
<tr><td>SIZE</td><td>2</td><td>xxxx</td><td>Size of EEPROM
Size of EEPROM to be filled
Note
The range of EEPROM size is theoretically 0001h..FFFFh. The typical value to fill the whole EEPROM for used device is 2000h</td></tr>
<tr><td>DATA</td><td>1</td><td>xx</td><td>Fill pattern
Data (00h..FFh) EEPROM to be filled with</td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	EA	xx	xx	01

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)

Code	Description
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**No SPI controller on given ports found.**
0x02	**HAL parameter error**
0x03	**SPI communication failed**
0x04	**Timeout EEPROM write enable(WEL) or write in progress(WIP)**
0xFF	**HAL busy**
0xFE	**Generic HAL error**
0xFD	**HAL not compatible to current CPU**
0xFC	**Invalid params**
0xFB	**HAL module not implemented**
0xFA	**HAL function not implemented**
0xEF	**GPIO port not found**
0xEE	**GPIO pin not found**
0xED	**GPIO function not supported**
0xEC	**GPIO port.pin / module not connectable**

TSW_ED - EEPROM Emulation V2.0

Test step description

Date: 17.08.2016

Created: 18.01.2017 09:12.

This TSW provides functions to access the EEPROM emulation. This TSW is similar to TSW_EE but uses memory layout for FEE version 2.0.

Notes for MDG1 emulation

The use of HSMDFLASH and DFLASH is completely independent from each other. So each of them have to be initialized before usage. The HSMDFLASH is not available on all devices. See documentation to TSW_99.

Test step commands

.

01 - Erase and init emulation

This command erases the complete emulated EEPROM and writes the initialization values.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	ED	01	01	...

Parameters

Parameter set for emulation initialization

5	+0	+1	+2
(1..n*)	**CFG**	**CS**	**CRYPT**

Details

Parameter set for emulation initialization

Name	Length	Fmt	Description
CFG	**1**	**xx**	**Configuration Data** **Bit(s)** / **Description** **7** – **Force flash erase** **0_B:** no erase if flash is erased **1_B:** force flash erase **6** – **Skip erased check** **0_B:** do erased (blank) check

Name	Length	Fmt	Description	
			Bit(s)	**Description**
				1_B: skip erased (blank) check
			5	**Destination** **0_B:** DFLASH (Data Flash) **1_B:** HSMDFLASH (only device 3 and 4)
			4	**Skip initialization** **0_B:** erase and initialize flash **1_B:** only erase
			3:0	**reserved**
CS	**1**	**xx**	**Chip Select (not used)**	
CRYPT	**1**	**xx**	**EEPROM encryption (not used)** **0_H:** Encryption disabled **XX_H:** Encryption enabled with specific version	

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	ED	xx	xx	01	...

Parameters

Response set for emulation initialization

...	+0
(1..n*)	**ASTAT**

Details

Response set for emulation initialization

Name	Length	Fmt	Description
ASTAT	**1**	**xx**	**Additional Status of Initialization** For every set a additional status is responded **Value** / **Meaning** 0x00 / no additional info 0xC8 / Function not possible (HSM)

.

02 - Write Data To Block

This command writes bytes into a block of the EEPROM Emulation. The desired block is defined by its FEE index.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	ED	02	01	...

Parameters

Parameter set for WriteBlock

5	+0	+1	+3	+5	+7	+8	+9
(1*)	**CFG**	**IDX**	**OFFS**	**BLEN**	**CS**	**CRYPT**	**DATA**

Details

Parameter set for WriteBlock

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>CFG</td><td>1</td><td>xx</td><td>Configuration Data
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7</td><td>force fill (only MDG1 DFLASH)

0_B: return error, if partial block and no previous block is available
1_B: fill unknown data with 0x00, if no previous block is available</td></tr>
<tr><td>6</td><td>force write (only MDG1 DFLASH)

0_B: no new write, if data, blocklength, security level and survival attribut are equal
1_B: write, even if data, blocklength, security level and survival attribut are equal</td></tr>
<tr><td>5</td><td>Destination

0_B: DFLASH (Data Flash)
1_B: HSMDFLASH (only device 3 and 4)</td></tr>
<tr><td>4</td><td>survival attribut (only MDG1 DFLASH)

0_B: no survival
1_B: survival</td></tr>
<tr><td>3</td><td>extended IDX (only used for MDG1 DFLASH in production)

0_B: normal (FSW) block
1_B: PAV exclusive block</td></tr>
<tr><td>2:1</td><td>reserved</td></tr>
</table>
</td></tr>
</table>

Name	Length	Fmt	Description
			Bit(s) / **Description** **0** — **security level (only MDG1 DFLASH)** **0_B:** Security Level "Single" **1_B:** Security Level "Double"
IDX	**2**	**xxxx**	**Block index** FEE index of the desired block
OFFS	**2**	**xxxx**	**Block offset** Offset address (in bytes) within the selected block. If the block is written the first time, the offset has to be zero and all data of the whole block should be available. (See also bit 7 in CFG).
BLEN	**2**	**xxxx**	**Total block length** Total length of the block (in bytes).
CS	**1**	**xx**	**Chip Select (not used)**
CRYPT	**1**	**xx**	**EEPROM encryption (only MDG1 DFLASH)** **0_H:** Encryption disabled **1_H:** Encryption with XTEA (AUDI)
DATA	**1**	**xx..**	**Data to write** 0 to 242 bytes to write into the selected eeprom block.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	ED	xx	xx	01	...

Parameters

Response of WriteDataToBlock

6	+0	+1	+5
(1*)	**BSB**	**STAT**	**DATA**

Details

Response of WriteDataToBlock

Name	Length	Fmt	Description
BSB	**1**	**xx**	**Block Status Byte** Current security level and survival attribut of the block. **Bit(s)** / **Description** **7:5** — **reserved** **4** — **survival attribut (only MDG1 DFLASH)**

Name	Length	Fmt	Description
			Bit(s) / **Description** **0_B:** no survival **1_B:** survival **3:1** — **reserved** **0** — **security level (only MDG1 DFLASH)** **0_B:** Security Level "Single" **1_B:** Security Level "Double"
STAT	**4**	**xxxxxxxx**	**Block status** Status information of the returned block **Bit(s)** / **Description** **31:16** — **reserved** **15:0** — **Block version counter** Number of updates of this block (not used and always zero).
DATA	**1**	**xx..**	**Uploaded block data bytes [0...242]**

.

03 - Read Data From Block

This command reads bytes from a block of the EEPROM Emulation. The desired block is defined by its FEE index.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	ED	03	01	...

Parameters

Parameter set for ReadDataFromBlock

5	+0	+1	+3	+5	+6	+7
(1..n*)	**CFG**	**IDX**	**OFFS**	**CNT**	**CS**	**CRYPT**

Details

Parameter set for ReadDataFromBlock

Name	Length	Fmt	Description
CFG	**1**	**xx**	**Configuration Data** **Bit(s)** / **Description** **7:6** — **reserved** **5** — **Destination** **0_B:** DFLASH (Data Flash)

Name	Length	Fmt	Description
			Bit(s) | **Description** | **1_B:** HSMDFLASH (only device 3 and 4) **4** | **reserved** **3** | **extended IDX (only used for MDG1 DFLASH in production)** **0_B:** normal (FSW) block **1_B:** PAV exclusive block **2:0** | **reserved**
IDX	**2**	**xxxx**	**Block index** FEE index of the desired block
OFFS	**2**	**xxxx**	**Block offset** Offset address (in bytes) within the selected block.
CNT	**1**	**xx**	**Number of bytes to read** If there are less bytes in the block than requested, the available bytes will be returned, but error "FEE: Not enough data" will be set.
CS	**1**	**xx**	**Chip Select (not used)**
CRYPT	**1**	**xx**	**EEPROM decryption** **0_H:** Decryption disabled **1_H:** Decryption with XTEA (AUDI)

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	ED	xx	xx	01	...

Parameters

Response of ReadDataFromBlock

6	+0	+1	+5	+6
(1..n*)	**BSB**	**STAT**	**ATTR**	**DATA**

Details

Response of ReadDataFromBlock

Name	Length	Fmt	Description
BSB	**1**	**xx**	**Block Status Byte** Current security level and survival attribut of the block **Bit(s)** | **Description** **7:5** | **reserved**

Name	Length	Fmt	Description
			<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>4</td><td>**survival attribut (only MDG1 DFLASH)** **0_B:** no survival **1_B:** survival</td></tr><tr><td>3:1</td><td>**reserved**</td></tr><tr><td>0</td><td>**security level (only MDG1 DFLASH)** **0_B:** Security Level "Single" **1_B:** Security Level "Double"</td></tr></table>
STAT	**4**	**xxxxxxxx**	**Block status** Status information of the returned block<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>31:16</td><td>**reserved**</td></tr><tr><td>15:0</td><td>**Block version counter** Number of updates of this block (not used and always zero).</td></tr></table>
ATTR	**1**	**xx**	**Block attributes (not used)**
DATA	**1**	**xx..**	**Uploaded block data bytes [0...244]**

.

23 - Read Data From Block (with BLEN)

This command is the same command as cmd 03 but with additional parameter BLEN. In case of HSMDFLASH the BLEN includes the CMAC data. So the BLEN is alway greater than 9.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	ED	23	01	...

Parameters

Parameter set for ReadDataFromBlock

5	+0	+1	+3	+5	+7	+8	+9
(1..n*)	**CFG**	**IDX**	**OFFS**	**BLEN**	**CNT**	**CS**	**CRYPT**

Details

Parameter set for ReadDataFromBlock

Name	Length	Fmt	Description
CFG	**1**	**xx**	**Configuration Data**<table><tr><th>Bit(s)</th><th>Description</th></tr><tr><td>7:6</td><td>**reserved**</td></tr></table>

Name	Length	Fmt	Description
			Bit(s) / **Description** **5** **Destination** **0_B:** DFLASH (Data Flash) **1_B:** HSMDFLASH (only device 3 and 4) **4** **reserved** **3** **extended IDX (only used for MDG1 DFLASH in production)** **0_B:** normal (FSW) block **1_B:** PAV exclusive block **2:0** **reserved**
IDX	**2**	**xxxx**	**Block index** FEE index of the desired block
OFFS	**2**	**xxxx**	**Block offset** Offset address (in bytes) within the selected block.
BLEN	**2**	**xxxx**	**Total block length** Total length of the block (in bytes).
CNT	**1**	**xx**	**Number of bytes to read** If there are less bytes in the block than requested, the available bytes will be returned, but error "FEE: Not enough data" will be set.
CS	**1**	**xx**	**Chip Select (not used)**
CRYPT	**1**	**xx**	**EEPROM decryption** **0_H:** Decryption disabled **1_H:** Decryption with XTEA (AUDI)

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	ED	xx	xx	01	...

Parameters

Response of ReadDataFromBlock

6	+0	+1	+5	+6
(1..n*)	**BSB**	**STAT**	**ATTR**	**DATA**

Details

Response of ReadDataFromBlock

Name	Length	Fmt	Description
BSB	**1**	**xx**	**Block Status Byte**

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>Current security level and survival attribut of the block
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:5</td><td>reserved</td></tr>
<tr><td>4</td><td>survival attribut (only MDG1 DFLASH)
0_B: no survival
1_B: survival</td></tr>
<tr><td>3:1</td><td>reserved</td></tr>
<tr><td>0</td><td>security level (only MDG1 DFLASH)
0_B: Security Level "Single"
1_B: Security Level "Double"</td></tr>
</table></td></tr>
<tr><td>STAT</td><td>4</td><td>xxxxxxxx</td><td>Block status
Status information of the returned block
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>31:16</td><td>reserved</td></tr>
<tr><td>15:0</td><td>Block version counter
Number of updates of this block (not used and always zero).</td></tr>
</table></td></tr>
<tr><td>ATTR</td><td>1</td><td>xx</td><td>Block attributes (not used)</td></tr>
<tr><td>DATA</td><td>1</td><td>xx..</td><td>Uploaded block data bytes [0...244]</td></tr>
</table>

.

04 - Delete Block

This command deletes a single EEPROM block.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	ED	04	01	...

Parameters

Parameter set for DeleteBlock

5 (1*)	+0	+1	+3
	CFG	**IDX**	**CS**

Details

Parameter set for DeleteBlock

Name	Length	Fmt	Description
CFG	**1**	**xx**	**Configuration Data**

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th colspan="2">Description</th></tr>
<tr><td rowspan="7"></td><td rowspan="7"></td><td rowspan="7"></td><td>Bit(s)</td><td>Description</td></tr>
<tr><td>7:6</td><td>reserved</td></tr>
<tr><td>5</td><td>Destination
0_B: DFLASH (Data Flash)
1_B: HSMDFLASH (only device 3 and 4)</td></tr>
<tr><td>4</td><td>reserved</td></tr>
<tr><td>3</td><td>extended IDX (only used for MDG1 DFLASH in production)
0_B: normal (FSW) block
1_B: PAV exclusive block</td></tr>
<tr><td>2:1</td><td>reserved</td></tr>
<tr><td>0</td><td>security level (only MDG1)
0_B: Security Level "Single"
1_B: Security Level "Double"</td></tr>
<tr><td>IDX</td><td>2</td><td>xxxx</td><td colspan="2">Block index
FEE index of the block to delete.</td></tr>
<tr><td>CS</td><td>1</td><td>xx</td><td colspan="2">Chip Select (not used)</td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	ED	xx	xx	01

.

06 - Get Emulation Info

Get status information of the Flash EEPROM Emulation.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	ED	06	01	...

Parameters

Parameter set for Get Emulation Info

...	+0
(1..n*)	**CFG**

Details

Parameter set for Get Emulation Info

Name	Length	Fmt	Description
CFG	**1**	**xx**	**Configuration Data** **Bit(s)** / **Description** **7:6** — **reserved** **5** — **Destination** **0_B:** DFLASH (Data Flash) **1_B:** HSMDFLASH (only device 3 and 4) **4:0** — **reserved**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	ED	xx	xx	01	...

Parameters

Response of GetEmulationInfo

6	+0	+4
(1*)	**SECTCNT**	**LVER**

Details

Response of GetEmulationInfo

Name	Length	Fmt	Description
SECTCNT	**4**	**xxxxxxxx**	**Sector change counter** Actual count of the 'Sector change counter'
LVER	**1**	**xx**	**Layout version** Version of the used FEE layout **Value** / **Meaning** 0x00 — No proper emulation found 0x01 — Emulation Version 1 0x02 — Emulation Version 2 0xFF — No information available

.

E0 - Get last HSM function return code

This command is only a placeholder and will always return an error response.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	ED	E0	01

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	ED	xx	xx	01

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0xEF	**HAL_FLS: Locking or unlocking of flash failed**
0xEE	**HAL_FLS: Desired flash area was not erased**
0xED	**HAL_FLS: Programming failed**
0xEC	**HAL_FLS: Erasing failed**
0xEB	**HAL_FLS: Start address not aligned on page size**
0xEA	**HAL_FLS: Desired range is too big for flash**
0xE9	**HAL_FLS: Compare failed**
0xE8	**HAL_FLS: Flash block is not allowed to program/erase**
0xCF	**FEE: Not enough data**
0xCE	**FEE: Block not found**
0xCD	**FEE: Sector error (e.g. emulation not initialized)**
0xCC	**FEE: Function/command not supported**
0xCB	**FEE: Illegal parameter**
0xCA	**FEE: Block CRC Error**
0xC9	**FEE: Not enough free memory in flash**
0xC8	**FEE: HSM function not possible**
0xC7	**FEE: Index for HSM unknown or no access**
0xC6	**FEE: HSM function returns error code**
0xC5	**FEE: Block is not supported**
0xC4	**FEE: Block is claimed as erased but is not blank**
0xC3	**FEE: something wrong with FEE (e.g. more than one FAT sector)**

TSW_EE - EEPROM Emulation V1.0

Test step description

Date: 17.08.2016

Created: 18.01.2017 09:12.

This TSW provides functions to access the EEPROM emulation.

Notes for E84 emulation

1) The E84 emulation has a concept requested by the customer (GM) and is different from the MDG1 CUBAS FEE concept.

2) Sector changes can be explicitly forced (see command 07) and have to be done at the end of the test to sort the FEE blocks and put them at the addresses requested by GM.

3) The following FEE indices are used:

```
0xCAFE: "BOSCH Factory Block""
0xDEAD: "GM Static Block"
0xBEEF: "GM Dynamic Block" (not supported)
```

Notes for MDG1 emulation

The use of HSMDFLASH and DFLASH is completely independent from each other. So each of them have to be initialized before usage. The HSMDFLASH is not available on all devices. See documentation to TSW_99.

If HSM is active only restricted access is possible. For valid FEE Blocks and their access rights see table 1.

Table 1.

FEE Block Index	FEE Block Name	Access
0x4100	rba_CryObj_SecData_256_1_1_e	R/W
0x4101	rba_CryObj_SecData_256_1_2_e	R/W
0x4102	rba_CryObj_SecData_256_1_3_e	R/W
0x4108	rba_CryObj_SecData_128_1_1_e	R/W
0x4109	rba_CryObj_SecData_128_1_2_e	R/W
0x410A	rba_CryObj_SecData_128_1_3_e	R/W
0x4110	rba_CryObj_SecData_64_1_1_e	R/W
0x4111	rba_CryObj_SecData_64_1_2_e	R/W
0x4112	rba_CryObj_SecData_64_1_3_e	R/W
0x411A	rba_CryObj_SecData_32_1_1_e	R/W
0x411B	rba_CryObj_SecData_32_1_2_e	R/W
0x411C	rba_CryObj_SecData_32_1_3_e	R/W
0x4122	rba_CryObj_SecData_256_2_1_e	R/W

FEE Block Index	FEE Block Name	Access
0x4123	rba_CryObj_SecData_256_2_2_e	R/W
0x412A	rba_CryObj_SecData_128_2_1_e	R/W
0x412B	rba_CryObj_SecData_128_2_2_e	R/W
0x4131	rba_CryObj_SecData_64_2_1_e	R/W
0x4132	rba_CryObj_SecData_64_2_2_e	R/W
0x4138	rba_CryObj_SecData_32_2_1_e	R/W
0x4139	rba_CryObj_SecData_32_2_2_e	R/W
0x41A1	rba_CryObj_PAVFlag_e	R

Test step commands

.

01 - Erase and init emulation

This command erases the complete emulated EEPROM and writes the initialization values.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	EE	01	01	...

Parameters

Parameter set for emulation initialization

5	+0	+1	+2
(1..n*)	**CFG**	**CS**	**CRYPT**

Details

Parameter set for emulation initialization

Name	Length	Fmt	Description
CFG	**1**	**xx**	**Configuration Data** **Bit(s)** / **Description** **7** — **Force flash erase (not used in E84)** **0_B:** no erase if flash is erased **1_B:** force flash erase **6** — **Skip erased check (not used in E84)** **0_B:** do erased (blank) check **1_B:** skip erased (blank) check **5** — **Destination** **0_B:** DFLASH (Data Flash)

Name	Length	Fmt	Description
			Bit(s) / **Description** **1_B:** HSMDFLASH (only device 3 and 4) **4** — **Skip initialization (not used in E84)** **0_B:** erase and initialize flash **1_B:** only erase **3:0** — **reserved**
CS	**1**	**xx**	**Chip Select (not used)**
CRYPT	**1**	**xx**	**EEPROM encryption (not used)** **0_H:** Encryption disabled **XX_H:** Encryption enabled with specific version

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	EE	xx	xx	01	...

Parameters

Response set for emulation initialization

6	+0
(1..n*)	**ASTAT**

Details

Response set for emulation initialization

Name	Length	Fmt	Description
ASTAT	**1**	**xx**	**Additional Status of Initialization** For every set a additional status is responded **Value** / **Meaning** 0x00 — no additional info 0xC8 — Function not possible (HSM)

.

02 - Write Data To Block

This command writes bytes into a block of the EEPROM Emulation. The desired block is defined by its FEE index.

Notes for E84 emulation

The Customer static block (block ID: 0xDEAD) at E84 EEPROM layout has a block length of 0x610 bytes (including header and footer). This block length is supported by

TSW EE subcommand 02, but for writing data only 242 bytes can be written at once. The total block length (BLEN) at the TSW command should be set equal 0x0000 because block length of static block is fix.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	EE	02	01	...

Parameters

Parameter set for WriteBlock

5	+0	+1	+3	+5	+7	+8
(1*)	**CFG**	**IDX**	**OFFS**	**BLEN**	**CS**	**CRYPT**

...	+9
(1*)	**DATA**

Details

Parameter set for WriteBlock

Name	Length	Fmt	Description
CFG	**1**	**xx**	**Configuration Data** **Bit(s)** / **Description** **7** **force fill (only MDG1 DFLASH)** **0_B:** return error, if partial block and no previous block is available **1_B:** fill unknown data with 0x00, if no previous block is available **6** **force write (only MDG1 DFLASH)** **0_B:** no new write, if data, blocklength, security level and survival attribut are equal **1_B:** write, even if data, blocklength, security level and survival attribut are equal **5** **Destination** **0_B:** DFLASH (Data Flash) **1_B:** HSMDFLASH (only device 3 and 4) **4** **survival attribut (only MDG1 DFLASH)** **0_B:** no survival

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td></td><td>1_B: survival</td></tr>
<tr><td>3</td><td>extended IDX (only used for MDG1 DFLASH in production)
0_B: normal (FSW) block
1_B: PAV exclusive block</td></tr>
<tr><td>2:1</td><td>reserved</td></tr>
<tr><td>0</td><td>security level (only MDG1 DFLASH)
0_B: Security Level "Single"
1_B: Security Level "Double"</td></tr>
</table>
</td></tr>
<tr><td>IDX</td><td>2</td><td>xxxx</td><td>Block index
FEE index of the desired block</td></tr>
<tr><td>OFFS</td><td>2</td><td>xxxx</td><td>Block offset
Offset address (in bytes) within the selected block. If the block is written the first time, the offset has to be zero and all data of the whole block should be available. (See also bit 7 in CFG).</td></tr>
<tr><td>BLEN</td><td>2</td><td>xxxx</td><td>Total block length
Total length of the block (in bytes). If set to 0x0000, TSW calculates the length automatically from the 'block offset' and the length of the previous version of this block (if present).
Notes for E84
Must be set equal 0x0000</td></tr>
<tr><td>CS</td><td>1</td><td>xx</td><td>Chip Select (not used)</td></tr>
<tr><td>CRYPT</td><td>1</td><td>xx</td><td>EEPROM encryption (only MDG1 DFLASH)
0_H: Encryption disabled
1_H: Encryption with XTEA (AUDI)</td></tr>
<tr><td>DATA</td><td>1</td><td>xx..</td><td>Data to write
0 to 242 bytes to write into the selected eeprom block.</td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	EE	xx	xx	01	...

Parameters

Response of WriteDataToBlock

6	+0	+1	+5
(1*)	**BSB**	**STAT**	**DATA**

Details

Response of WriteDataToBlock

Name	Length	Fmt	Description
BSB	**1**	**xx**	**Block Status Byte** Current security level and survival attribut of the block. **Bit(s)** | **Description** **7:5** | **reserved** **4** | **survival attribut (only MDG1 DFLASH)** **0_B:** no survival **1_B:** survival **3:1** | **reserved** **0** | **security level (only MDG1 DFLASH)** **0_B:** Security Level "Single" **1_B:** Security Level "Double"
STAT	**4**	**xxxxxxxx**	**Block status** Status information of the returned block **Bit(s)** | **Description** **31:16** | **reserved** **15:0** | **Block version counter** Number of updates of this block.
DATA	**1**	**xx..**	**Uploaded block data bytes [0...242]**

.

03 - Read Data From Block

This command reads bytes from a block of the EEPROM Emulation. The desired block is defined by its FEE index.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	EE	03	01	...

Parameters

Parameter set for ReadDataFromBlock

5	+0	+1	+3	+5	+6	+7
(1..n*)	**CFG**	**IDX**	**OFFS**	**CNT**	**CS**	**CRYPT**

Details

Parameter set for ReadDataFromBlock

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>CFG</td><td>1</td><td>xx</td><td>Configuration Data
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:6</td><td>reserved</td></tr>
<tr><td>5</td><td>Destination
0_B: DFLASH (Data Flash)
1_B: HSMDFLASH (only device 3 and 4)</td></tr>
<tr><td>4</td><td>reserved</td></tr>
<tr><td>3</td><td>extended IDX (only used for MDG1 DFLASH in production)
0_B: normal (FSW) block
1_B: PAV exclusive block</td></tr>
<tr><td>2:0</td><td>reserved</td></tr>
</table></td></tr>
<tr><td>IDX</td><td>2</td><td>xxxx</td><td>Block index
FEE index of the desired block</td></tr>
<tr><td>OFFS</td><td>2</td><td>xxxx</td><td>Block offset
Offset address (in bytes) within the selected block.</td></tr>
<tr><td>CNT</td><td>1</td><td>xx</td><td>Number of bytes to read
If there are less bytes in the block than requested, the available bytes will be returned, but error "FEE: Not enough data" will be set.</td></tr>
<tr><td>CS</td><td>1</td><td>xx</td><td>Chip Select (not used)</td></tr>
<tr><td>CRYPT</td><td>1</td><td>xx</td><td>EEPROM decryption
0_H: Decryption disabled
1_H: Decryption with XTEA (AUDI)</td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	EE	xx	xx	01	...

Parameters

Response of ReadDataFromBlock

6	+0	+1	+5	+6
(1..n*)	**BSB**	**STAT**	**ATTR**	**DATA**

Details

Response of ReadDataFromBlock

Name	Length	Fmt	Description
BSB	**1**	**xx**	**Block Status Byte**

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>Current security level and survival attribut of the block
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:5</td><td>reserved</td></tr>
<tr><td>4</td><td>survival attribut (only MDG1)
0_B: no survival
1_B: survival</td></tr>
<tr><td>3:1</td><td>reserved</td></tr>
<tr><td>0</td><td>security level (only MDG1)
0_B: Security Level "Single"
1_B: Security Level "Double"</td></tr>
</table></td></tr>
<tr><td>STAT</td><td>4</td><td>xxxxxxxx</td><td>Block status
Status information of the returned block
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>31:16</td><td>reserved</td></tr>
<tr><td>15:0</td><td>Block version counter
Number of updates of this block.</td></tr>
</table></td></tr>
<tr><td>ATTR</td><td>1</td><td>xx</td><td>Block attributes (not used)</td></tr>
<tr><td>DATA</td><td>1</td><td>xx..</td><td>Uploaded block data bytes [0...244]</td></tr>
</table>

.

04 - Delete Block

This command deletes a single EEPROM block. (Not available for E84 emulation)

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	EE	04	01	...

Parameters

Parameter set for DeleteBlock

5	+0	+1	+3
(1*)	**CFG**	**IDX**	**CS**

Details

Parameter set for DeleteBlock

Name	Length	Fmt	Description
CFG	**1**	**xx**	**Configuration Data**

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td></td><td>Bit(s)</td><td>Description</td></tr>
<tr><td></td><td></td><td></td><td>7:6</td><td>reserved</td></tr>
<tr><td></td><td></td><td></td><td>5</td><td>Destination
0_B: DFLASH (Data Flash)
1_B: HSMDFLASH (only device 3 and 4)</td></tr>
<tr><td></td><td></td><td></td><td>4</td><td>reserved</td></tr>
<tr><td></td><td></td><td></td><td>3</td><td>extended IDX (only used for MDG1 DFLASH in production)
0_B: normal (FSW) block
1_B: PAV exclusive block</td></tr>
<tr><td></td><td></td><td></td><td>2:1</td><td>reserved</td></tr>
<tr><td></td><td></td><td></td><td>0</td><td>security level (only MDG1)
0_B: Security Level "Single"
1_B: Security Level "Double"</td></tr>
<tr><td>IDX</td><td>2</td><td>xxxx</td><td colspan="2">Block index
FEE index of the block to delete.</td></tr>
<tr><td>CS</td><td>1</td><td>xx</td><td colspan="2">Chip Select (not used)</td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	EE	xx	xx	01

.

05 - Check for ASW pages

This command is used to check for ASW pages. If any other block than the 'predefined blocks for factory programming' is in the active sector of the emulation, the status bit 5 will be set.

not implemented, yet!

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	EE	05	01

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	EE	xx	xx	01

.

06 - Get Emulation Info

Get status information of the Flash EEPROM Emulation.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	EE	06	01	...

Parameters

Parameter set for Get Emulation Info

... (1..n*)	+0
	CFG

Details

Parameter set for Get Emulation Info

Name	Length	Fmt	Description
CFG	**1**	**xx**	**Configuration Data** **Bit(s)** – **Description** **7:6** – **reserved** **5** – **Destination** **0**$_B$**:** DFLASH (Data Flash) **1**$_B$**:** HSMDFLASH (only device 3 and 4) **4:0** – **reserved**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	EE	xx	xx	01	...

Parameters

Response of GetEmulationInfo

6 (1*)	+0	+4
	SECTCNT	**LVER**

Details

Response of GetEmulationInfo

Name	Length	Fmt	Description
SECTCNT	**4**	**xxxxxxxx**	**Sector change counter**

Name	Length	Fmt	Description
			Actual count of the 'Sector change counter'
LVER	**1**	**xx**	**Layout version** Version of the used FEE layout **Value** / **Meaning** 0xFF / No information available

.

07 - Force sector change

The next sector will be erased and the latest versions of all FEE blocks are copied into this new sector. When the last sector is reached, the TSW starts from the first sector, again. All block properties (security levels etc.) are copied as well.

E84: The 'block version counter' (see CMD_03 parameter STAT) will be reset to 1. 'GM Static Blocks' are not copied.

JDP/IFX: command not implemented, yet!

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	EE	07	01	...

Parameters

Parameter set for SectorChange

5	+0	+1	+2
(1*)	**CFG**	**CS**	**CRYPT**

Details

Parameter set for SectorChange

Name	Length	Fmt	Description
CFG	**1**	**xx**	**Configuration Data (reserved)**
CS	**1**	**xx**	**Chip Select (not used)**
CRYPT	**1**	**xx**	**EEPROM encryption (not used)** **0_H:** Encryption disabled **XX_H:** Encryption enabled with specific version

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	EE	xx	xx	01

.

08 - Set flash module

This command is only available for MDG1 emulation. The flash module after load is set to DFLASH (01).

Note

This command is obsolete, because of destination bit (bit number 5) in commands. Don't use it any more!

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	EE	08	01	...

Parameters

parameters for command

5	+0
(1*)	**MODULE**

Details

parameters for command

Name	Length	Fmt	Description
MODULE	**1**	**xx**	**Flash Module for EEPROM Emulation** see tables 1 - 6 of teststep tsw_99 for the addressmapping **01_H:** DFLASH (Data Flash) **05_H:** HSMDFLASH (HSM Data Flash, only device 3 and 4)

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	EE	xx	xx	01	...

Parameters

Response of set flash module

6	+0
(1*)	**ASTAT**

Details

Response of set flash module

Name	Length	Fmt	Description
ASTAT	**1**	**xx**	**Additional Status**

Name	Length	Fmt	Description
			A additional status is responded **Table 2.** Value / Meaning: 0x00 – switch to DFLASH or switch to HSMDFLASH and HSM is not active 0x01 – switch to HSMDFLASH and HSM is active

.

E0 - Get last HSM function return code

This command is used to get the last HSM function return code. Additional it use HSM function EcuSecuSrv_GetEcuMode() to get more info about HSM and EcuSecuSrv states. It is only used for debugging purpose. (Only MDG1 emulation).

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	EE	E0	01

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	EE	xx	xx	01	...

Parameters

Response of get last return code

6 (1*)	+0	+1	+5	+9
	RF	**RC**	**RV**	**HSMST**

... (1*)	+10	+11	+12	+13
	ESSST	**EXTIST**	**RSV**	**UPTIME**

Details

Response of get last return code

Name	Length	Fmt	Description
RF	**1**	**xx**	**used HSM function**
RC	**4**	**xxxxxxxx**	**return code of used HSM function**
RV	**4**	**xxxxxxxx**	**return value from GetEcuMode()**
HSMST	**1**	**xx**	**HSM state from GetEcuMode()**
ESSST	**1**	**xx**	**EcuSecuSrv state from GetEcuMode()**
EXTIST	**1**	**xx**	**ExtInfo state from GetEcuMode()**
RSV	**1**	**xx**	**Reserved info from GetEcuMode()**

Name	Length	Fmt	Description
UPTIME	**4**	**xxxxxxxx**	**HSM uptime from GetEcuMode()**

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**

Code	Description
0xEF	HAL_FLS: Locking or unlocking of flash failed
0xEE	HAL_FLS: Desired flash area was not erased
0xED	HAL_FLS: Programming failed
0xEC	HAL_FLS: Erasing failed
0xEB	HAL_FLS: Start address not aligned on page size
0xEA	HAL_FLS: Desired range is too big for flash
0xE9	HAL_FLS: Compare failed
0xE8	HAL_FLS: Flash block is not allowed to program/erase
0xCF	FEE: Not enough data
0xCE	FEE: Block not found
0xCD	FEE: Sector error (e.g. emulation not initialized)
0xCC	FEE: Function/command not supported
0xCB	FEE: Illegal parameter
0xCA	FEE: Block CRC Error
0xC9	FEE: Not enough free memory in flash
0xC8	FEE: HSM function not possible (only for MDG1)
0xC7	FEE: Index for HSM unknown or no access (only for MDG1)
0xC6	FEE: HSM function returns error code (only for MDG1)
0xC5	FEE: Block is not supported (only for E84)
0xC4	FEE: Block is claimed as erased but is not blank (only MDG1)

TSW_EF - FBS4 Initialisation

Test step description

Date: 03.11.2016

Created: 18.01.2017 09:12.

This TSW provides functions to initialise the FBS4 (Fahrberechtigungssystem Stufe 4) from Daimler.

Note: This teststep is available for JDP and IFX.

Test step commands

.

01 - Initialise FBS4

This command initialise the FBS4. This is done in the following way: The command overwrites the EEP_DATA5 with the random number and clears every other Bit in the Block. EEP_DATA4 will be created as blank Block. Then the initialisation is forecd by transmitting EEP_DATA1-4 als blank blocks (not the real content of EEP_DATAx is transmitted). Then the Daimler-SW is triggered. In the next step the FBS4 Initialisationdata is transmitted to the Daimler-SW. The the Daimler-SW is triggered again and will responed the MDC to the TSW. The MDC is checked and EEP_DATA5 will be filled with the FBS4 serialnumber if the MDC is correct and every block in the data flash has been written correctly.

Hint: Before use of this teststep, the eeprom emulation have to be initialized once with teststep TSW_EE/TSW_ED.

Hint: Because TSW_EE/TSW_ED is used to write data, the teststep TSW_EE/TSW_ED have to be loaded too.

Table 1. predefined FEE index mapping

Block name	FEE index
EEPData1	0xFC22
EEPData2	0xFF22
EEPData3	0xFE22
EEPData4	0xF922
EEPData5	0xF822

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	EF	01	01	...

Parameters

Parameter set for FBS4 initialisation

5	+0	+1	+2
(1*)	**INI**	**RAND**	**NUM**

Details

Parameter set for FBS4 initialisation

Name	Length	Fmt	Description
INI	**1**	**xx..**	**FBS4 initial data (97 Bytes)**
RAND	**1**	**xx..**	**FBS4 random data (48 Bytes)**
NUM	**1**	**xx..**	**FBS4 number in ascii (24 Bytes)**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	EF	xx	xx	01	...

Parameters

Response of InitFBS4

...	+0
(1*)	**MDC**

Details

Response of InitFBS4

Name	Length	Fmt	Description
MDC	**1**	**xx..**	**Modification Detection Code (24 Bytes)** Byte #5 and #6 (with the first byte called #1) of the MDC represent an error code: **00_H:** OK **01_H:** ECU was initialized before **03_H:** ECU was activated before **04_H:** Checksum failure **$0C_H$:** EEPROM access failure **$0F_H$:** unspecific failure

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**

Code	Description
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0xFA	**HAL: Function not implemented (FEE teststep not loaded)**
0xEF	**HAL: Locking or unlocking of flash failed**
0xEE	**HAL: Desired flash area was not erased**
0xED	**HAL: Programming failed**
0xEC	**HAL: Erasing failed**
0xEB	**HAL: Start address not aligned on page size**
0xEA	**HAL: Desired range is too big for flash**
0xE9	**HAL: Compare failed**
0xCF	**FEE: Not enough data**
0xCE	**FEE: Block not found**

Code	Description
0xCD	**FEE: Sector error (e.g. emulation not initialized)**
0xCC	**FEE: Function/command not supported**
0xCB	**FEE: Illegal parameter**
0xCA	**FEE: Block CRC Error**
0xC9	**FEE: Not enough free memory in flash**

TSW_F9 - Flash test

Test step description

Date: 17.08.2016

Created: 18.01.2017 09:12.

This TSW is used to get information about the Flash and to trigger test on Flash.

Test step commands

.

01 - Read erase counter of PFlash

This command reads out how many times the PFlash has been deleted.

Device 3, 4: PFlash is distributed onto banks. All banks are based on the same sectorization: a 2MByte bank implements logical sector S0 to S26, a 1.5 MByte bank logical sectors S0 to S24 and a 1 MByte bank logical sectors S0 to S22. Each sector has its own erase counter. Counters saturate after 64 times erase.

Device 5: An erase counter is a dedicated 16 Kbyte physical sector within each PFp bank (p = 0..5). Each sector is capable of holding 512 erase entries.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	F9	01	01

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	F9	xx	xx	01	...

Parameters

Response set for erase counter (device 3, 4)

...	+0
(1..n*)	**CNT**

Response set for erase counter (device 5)

...	+0
(1..n*)	**CNT**

Details

Response set for erase counter (device 3, 4)

Name	Length	Fmt	Description
CNT	**1**	**xx**	**Content of erase counters** Content of the erase counters distributed on logical sectors. 27 counters for each bank.

Response set for erase counter (device 5)

Name	Length	Fmt	Description
CNT	**2**	**xxxx**	**Content of erase counters** Number of entries in dedicated sector.

.

04 - Check contents of flash area

This command is used to check if a flash area (defined by a startaddress and a length) is erased or is programmed (contains data). This is more or less a blankcheck.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	F9	04	01	...

Parameters

parameters for command

5	+0	+1	+5
(1..n*)	**MODULE**	**ADR**	**NUM**

Details

parameters for command

Name	Length	Fmt	Description
MODULE	**1**	**xx**	**Flash module for check** see tables 1 - 6 of teststep tsw_99 for the addressmapping **00_H:** PFLASH (Program Flash) **01_H:** DFLASH (Data Flash) **04_H:** HSMPFLASH (HSM Program Flash) **05_H:** HSMDFLASH (HSM Data Flash)
ADR	**4**	**xxxxxxxx**	**Startaddress of area** Absolute startaddress of area to check
NUM	**4**	**xxxxxxxx**	**Number of bytes** Number of bytes in area to check

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	F9	xx	xx	01	...

Parameters

Response of command

6	+0	+1
(1..n*)	**RSLT**	**ADDR**

Details

Response of command

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>RSLT</td><td>1</td><td>xx</td><td>Result of the check
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7:1</td><td>reserved, always zero</td></tr>
<tr><td>0</td><td>State

0_B: Area is totally erased (can be programmed)
1_B: Area is not totally erased (contains already programmed regions)</td></tr>
</table></td></tr>
<tr><td>ADDR</td><td>4</td><td>xxxxxxxx</td><td>Address of first programmed page
Address of the first programmed page, if area is not totally erased, else 0.</td></tr>
</table>

.

10 - Perform check margin test

This command is used to run a check margin test (also known as read margin test)

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	F9	10	01	...

Parameters

parameters for command

5	+0	+1
(1*)	**MODULE**	**CONFIG**

Details

parameters for command

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>MODULE</td><td>1</td><td>xx</td><td>Flash module for test
see tables 1 - 6 of teststep tsw_99 for the addressmapping

00_H: PFLASH (Program Flash)
01_H: DFLASH (Data Flash)
04_H: HSMPFLASH (HSM Program Flash)
05_H: HSMDFLASH (HSM Data Flash)</td></tr>
<tr><td>CONFIG</td><td>1</td><td>xx</td><td>Configuration for test
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>7</td><td>respond error addresses

0_B: disabled
1_B: enabled</td></tr>
<tr><td>6:1</td><td>reserved</td></tr>
<tr><td>0</td><td>Margin level for test

0_B: LOW level
1_B: HIGH level</td></tr>
</table></td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	F9	xx	xx	01	...

Parameters

Response of command

6	+0	+4	+8	+12
(1*)	**ERR1**	**ERR2**	**ERR3**	**ERR4**

Additional response of command (if respond addresses enabled)

22	+0	+1	+2	+3	+4
(1*)	**NR1**	**NR2**	**NR3**	**NR4**	**ADDR**

Details

Response of command

Name	Length	Fmt	Description
ERR1	**4**	**xxxxxxxx**	**total number of single bit errors**
ERR2	**4**	**xxxxxxxx**	**total number of double bit errors**

Name	Length	Fmt	Description
ERR3	**4**	**xxxxxxxx**	**total number of 3-bit errors**
ERR4	**4**	**xxxxxxxx**	**total number of 4-bit errors**

Additional response of command (if respond addresses enabled)

Name	Length	Fmt	Description
NR1	**1**	**xx**	**Number of responded 1-bit error addresses**
NR2	**1**	**xx**	**Number of responded 2-bit error addresses**
NR3	**1**	**xx**	**Number of responded 3-bit error addresses**
NR4	**1**	**xx**	**Number of responded 4-bit error addresses**
ADDR	**4**	**xxxxxxxx..**	**Addresses of faults** **Note** The responded fault addresses points to the start of a page. For IFX PFLASH a page includes 32 and for IFX DFLASH 8 bytes.

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)

Code	Description
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**Function not implemented**
0xE0	**BootCtl to old**
0xE7	**HAL-FLS: Flash area not available**
0xEA	**HAL-FLS: Desired range is too big for flash**
0xEB	**HAL-FLS: Start address not aligned on page size**
0xFA	**HAL-FLS: Function not implemented**
0xF0	**HAL-ECUSECU: HSM jump to RAM not successful**

TSW_FA - Flash programming

Test step description

Purpose

TSW-FA is intended for Flash erasing and programming of a complete software image for application and serial ECUs.

Important

Always execute a hardware reset after programming has finished with TSW-FA !!!

This TSW is intended to be used for erasing and programming whole program flashes which are either completely erased (first programming) or already programmed (reprogramming). It contains a simple KWP2000 communication handler with a reduced number of services which all run out of RAM.

MDG1: DFlash (EEPROM) is not supported.

MDG1: Parallel programming not implemented. Internal flash has only one flash controller. An external flash is not supported.

For series ECUs, the TSW-FA not supports the RB tuningprotection algorithms.

The TSW-FA is not recommended to be used to program only single pages, because after issueing command 0x02 the selected address range will be erased. Since a flash can always be erased only one sector / block only, the whole sector containing the page will be erased. Sector sizes differ between 16 kbyte up to 256 kbyte. To program small data blocks (eg. logistic data etc.) use TSW-FE instead. (Flash part naming: IFX - sector / JDP - block)

The recommended programming flow is described in the document: to be done.

Table 1. List of supported KP2000 services

Service (SID)	Subservice (SSID)	Description
0x3B	..	Write data by local identifier
"	0x00	number of ID's
"	0x01	set baud rate
0x80	..	Escape code
"	FA..	only TSW_FA commands
0x82	-	Stop Communication

detailed description: see "Implemented KWP2000 Services"

Test step commands

.

01 - Select flash types

This command has to to tell the TSW , which flash types have to be erased and programmed and must be called prior to the commands 0x02 and above.

The flash selection is done by setting the byte field SADR to the corresponding flash base address.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	FA	01	01	...

Parameters

5	+0	+1
(1..n*)	**CS**	**SADR**

Details

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>CS</td><td>1</td><td>xx</td><td>Chip select
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>0</td><td>erase all

0: only with command 0x02 selected areas will be erased.

1: complete PFlash will be erased first. BootCtrl and EcuSecu areas are special cases.</td></tr>
<tr><td>1</td><td>Factory program/erase mode disable

0: Factory mode enabled (only relevant for devices that support factory mode).

1: Factory mode disabled.</td></tr>
<tr><td>6:2</td><td>reserved
must be set to 0</td></tr>
<tr><td>7</td><td>PFlash erase handling

0: erase selected range only if it is not programmed.

1: force erase: do not check range, erase always selected range.</td></tr>
</table>
</td></tr>
<tr><td>SADR</td><td>4</td><td>xxxxxxxx</td><td>Start address
start adress of flash</td></tr>
</table>

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	FA	xx	xx	01	...

Parameters

Response data

6 (1..n*)	+0	+4	+8
	FLSID	**FLSSIZE**	**FLSST**

Details

Response data

Name	Length	Fmt	Description
FLSID	**4**	**xxxxxxxx**	**Flash Identifier**
FLSSIZE	**4**	**xxxxxxxx**	**Flash size**
FLSST	**1**	**xx**	**Flash state** 0x00: Ok

.

02 - Select erase/program adresses

This command has to be used after command 0x01 to select the address to be erased/ programmed for the flash selected with command 0x01.
To program multible ranges of one flash type you can set all ranges with one call, or process one range after the other.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	FA	02	01	...

Parameters

5 (1..n*)	+0	+4
	SADR	**EADR**

Details

Name	Length	Fmt	Description
SADR	**4**	**xxxxxxxx**	**Start address** First byte address of range to be erased/ programmed. Address must be page aligned (last byte of address must be 0). note: Erased will be always a complete sector / block. Refer to the manual to get a list of the sector configuration.
EADR	**4**	**xxxxxxxx**	**End address** Last byte-address to be erased/programmed. Address must be page aligned (last byte of address must be FF)

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	FA	xx	xx	01

.

10 - Download data for flash id 0

This command has to be used to download flash data for a specific flash, where "X" selects the target flash type (= index of the corresponding flash selected with subcommand 0x01, starting to count with "0". Note, that if decompression is enabled, all source data files for the corresponding flash must be available in a compressed format.

For example (CFITerm):

LOADFILE globalecusw.hex,80000000,200000,255,80FA1001,C0FA00000100,00,last

downloads the hex-file contents of globalecusw.hex starting from address 0x80000000 with a size of 0x200000 to the target flash with index 0. Fillbyte is 0x00. Responce check (C0FA000001) inclusive ERR (00: no error) information.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	FA	10	01	...

Parameters

programm data

5	+0
(1..n*)	**PDATA**

Details

programm data

Name	Length	Fmt	Description
PDATA	**1**	**xx**	**programm data** dummy

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	FA	xx	xx	01	...

Parameters

response of command "Download data"

6	+0	+1
(1*)	**ERR**	**NBUF**

Details

response of command "Download data"

Name	Length	Fmt	Description
ERR	**1**	**xx**	**Flash driver error codes** **00_H:** no error **01_H:** Erase failed (HW indicated error) **02_H:** Programming failed (HW indicated error) **03_H:** Verify failed **04_H:** Flash type not supported **05_H:** Invalid boot file **06_H:** Invalid TPROT signature **07_H:** No CB found **08_H:** Invalid SB/TPROT signature **09_H:** Invalid block signature **$0A_H$:** Invalid flash signature **$0B_H$:** No signature file loaded **$0C_H$:** Failed to unlock **$0D_H$:** CB on different position **$0E_H$:** Forbidden up-or downgrade (crossing border TPROT 8) **$0F_H$:** TPROT not found on correct position **80_H:** Flash size unknown, defaults to 2MByte. This value is or'ed with other errorcodes and can be masked if not needed.
NBUF	**1**	**xx**	**Next Buffer** (only parallel programming, otherwise ignore value). If more than one flash is to be programmed in parallel mode, this byte informs the tester, which flash is recommended to be handled next.

.

11 - Download data for flash id 1

not implemented now

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	FA	11	01	...

Parameters

dummy

5	+0
(1..n*)	**DUMMY**

Details

dummy

Name	Length	Fmt	Description
DUMMY	**1**	**xx**	**dummy element** dummy

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	FA	xx	xx	01

.

12 - Download data for flash id 2

not implemented now

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	FA	12	01	...

Parameters

dummy

5	+0
(1*)	**DUMMY**

Details

dummy

Name	Length	Fmt	Description
DUMMY	**1**	**xx..**	**dummy element** dummy

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	FA	xx	xx	01

.

1F - Finish programming session

This command must be issued at the end of the data download to wait until all internal buffers have run empty.

If no signature file is loaded and tuning protection is not enabled, a CRC (IEEE 802.3) is calculated of the whole programming range for each flash separatly, which has been selected by command 0x01 and 0x02.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	FA	1F	01

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	FA	xx	xx	01	...

Parameters

response of command "Finish programming session"

6	+0
(1*)	**AUT**

response of command "Finish programming session"

7	+0
(1..n*)	**CRC**

Details

response of command "Finish programming session"

Name	Length	Fmt	Description
AUT	**1**	**xx**	**Authentification status** **00_H:** Ok **09_H:** Invalid block signature (BootCtrl) **40_H:** 0x40+: invalid block signature of range N **signature check failed** if the used certificate is the root cause, check also fails. Additional use "reset flash driver" command to check ESS state.

response of command "Finish programming session"

Name	Length	Fmt	Description
CRC	**4**	**xxxxxxxx**	**Range CRC for flash n** The CRC is only calculated, if tuningprotection is not active and no signature file has been loaded with command 0x81. Otherwise this value is 0. If HSM is active and selected range contains a EcuSecu range, CRC will be calculated with already existing EcuSecu data. CRC can (will) be different, because these ranges are skiped.

.

80 - Download signature

Can only be used for IFX and JDP.

This command is used to download a signature file prior to flash programming. This is mandatory, if tuning protection is activated.

For example (CFITerm):

LOADFILE signature.hex,,,255,80FA8001,C0FA000001,00,last

downloads the whole file signature.hex.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	FA	80	01	...

Parameters

Signature data

5	+0
(1*)	**SDATA**

Details

Signature data

Name	Length	Fmt	Description
SDATA	**1**	**xx**	**Signature data** dummy

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	FA	xx	xx	01

.

81 - Download certificate

This command is used to download a certificate file prior to flash programming. This is mandatory, if tuning protection is activated and the certificate is availble at FSW container.

Each time a certificate is loaded, the one before is erased. For next signature check (finish) the new certificate is used.

Note

Certificates bigger than 1 KB will be rejected.

certificate usage

If no certificate is (successfully) loaded, existing certificate in ECU will be used.

For example (CFITerm):

LOADFILE certificate.hex,,,255,80FA8101,C0FA000001,00,last

downloads the whole file certificate.hex.

Errors are reported only once. Download procedure is reset automatical.

Without data the command returns state of certificate loading.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	FA	81	01	...

Parameters

Certificate

5	+0
(1*)	**SDATA**

Details

Certificate

Name	Length	Fmt	Description
SDATA	**1**	**xx..**	**Certificate data** dummy

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	FA	xx	xx	01	...

Parameters

response of command "Download certificate"

6	+0
(1*)	**DLCEST**

Details

response of command "Download certificate"

Name	Length	Fmt	Description
DLCEST	**1**	**xx**	**download state of certificate** **00_H:** no certificate loaded **01_H:** loading **02_H:** certificate completely loaded, ready to use

.

F0 - Get flash driver status

This command can be used to read the status of the flash driver for each flash. It can be issued anytime and shows if eg. erasing or programming is active, finished or if an error occured during a flash operation. According to the flash selection with command 0x01, the status values are responded in same order.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	FA	F0	01

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	FA	xx	xx	01	...

Parameters

response of command "Get flash driver status"

6	+0	+1	+5	+9
(1..n*)	**STAT**	**ERR**	**ACADD**	**ACER**

Details

response of command "Get flash driver status"

Name	Length	Fmt	Description
STAT	**1**	**xx**	**state of flash driver** **00_H:** no flash-/erase- order pending **02_H:** active erasing

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>03_H: active programming
04_H: finished programming
05_H: active programming - skip EcuSecu range
80_H: error</td></tr>
<tr><td>ERR</td><td>4</td><td>xxxxxxxx</td><td>Error

Error information: 1 byte error code, 3 byte error information.

<table>
<tr><th>Byte</th><th>description</th></tr>
<tr><td>0</td><td>error byte: Describes the error, if any. Otherwise value is 0x00 (see HEX table)</td></tr>
<tr><td>1..3</td><td>error info: Additional error information, eg. error code by HW-statemachine of flash. Meaning depends on type of the error code</td></tr>
</table>

00_H: no error
01_H: Erase failed (HW indicated error)
02_H: Programming failed (HW indicated error)
03_H: Verify failed
04_H: Flash type not supported
05_H: Invalid boot file
06_H: Invalid TPROT signature
07_H: No CB found
08_H: Invalid SB/TPROT signature
09_H: Invalid block signature
$0A_H$: Invalid flash signature
$0B_H$: No signature file loaded
$0C_H$: Failed to unlock
$0D_H$: CB on different position
$0E_H$: Forbidden up-or downgrade (crossing border TPROT 8)
$0F_H$: TPROT not found on correct position
80_H: Flash size unknown, defaults to 2MByte. This value is or'ed with other errorcodes and can be masked if not needed.</td></tr>
<tr><td>ACADD</td><td>4</td><td>xxxxxxxx</td><td>Actual programming address</td></tr>
<tr><td>ACER</td><td>4</td><td>xxxxxxxx</td><td>Actual erasing address</td></tr>
</table>

.

F1 - Get timing info

not implemented now

This command can be used after finished programming to read the programming time per page (min, max and average) for each programmed flash. Evaluation of the erasing time is actually not implemented and always responded as "0".

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	FA	F1	01

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	FA	xx	xx	01	...

Parameters

response of command "Get timing info"

6 (1*)	+0	+4	+8
	AVPTIM	**MINPTIM**	**MAXPTIM**
... (1*)	**+12**	**+16**	**+20**
	AVETIM	**MINETIM**	**MAXETIM**

Details

response of command "Get timing info"

Name	Length	Fmt	Description
AVPTIM	**4**	**xxxxxxxx**	**Average programming time**
MINPTIM	**4**	**xxxxxxxx**	**Minimum programming time**
MAXPTIM	**4**	**xxxxxxxx**	**Maximum programming time**
AVETIM	**4**	**xxxxxxxx**	**Average erase time**
MINETIM	**4**	**xxxxxxxx**	**Minimum erase time**
MAXETIM	**4**	**xxxxxxxx**	**Maximum erase time**

.

F2 - Read unique ID

This command can be used anytime and returns 32 bytes with the unique chip identifier (chip series number). The number is right-justified (leading zeros).

Table 2. Length of CPU Id

type	number of bytes	specific name
IFX	16	Unique Chip ID - UCB_IFX
JDP	32	Unique Identifier - UID

type	number of bytes	specific name
FSL	16	Unique Device ID

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	FA	F2	01

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	FA	xx	xx	01	...

Parameters

Response data

6	+0
(1*)	**UNIID**

Details

Response data

Name	Length	Fmt	Description
UNIID	**1**	**xx..**	**Unique Identifier** 32 bytes unique chip identifier

.

FF - Reset flash driver

This command resets the internal flash driver statemachines and aborts pending flash operations.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	FA	FF	01	...

Parameters

Reset

5	+0
(0..n*)	**RST**

Details

Reset

Name	Length	Fmt	Description
RST	**1**	**xx**	**Enable reset of CPU (optional)** Not implemented: If this byte is appended with the value 0x55, the CPU is resetted by forcing a SW initiated system reset.

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	FA	xx	xx	01	...

Parameters

response of command "Reset flash driver"

6	+0	+1	+2	+3
(1*)	**ID**	**HWCFG**	**ESS**	**TP**

Details

response of command "Reset flash driver"

Name	Length	Fmt	Description
ID	**1**	**xx**	**TSW-FA version ID**
HWCFG	**1**	**xx**	**HWCFG**
ESS	**1**	**xx**	**EcuSecuSrv status** **01_H:** EcuSecu Ok **10_H:** needed EcuSecu function not available

TSW-FA version ID:

type	**ID**	**description**
IFX	0x0C	added consideration of feature "DAP over CAN Physical Layer"
JDP/ IFX	0x0D	added: EcuSecu handling of broken IB link chain
JDP	0x0E	bugfix: core 0 handling additional for Dev 2 and Dev 3
IFX	0x0F	internal changes and speed improvement of command 0x10
JDP	0x10	added first version of TP handling for Dev1 - do not use!
JDP	0x11	correction of TP handling for Dev1
IFX/ JDP	0x12	added: optional loading of external certificate for verification
JDP	0x18	added: reprogramming BAF for JDP Dev3+ Cut 2 (project specific version of TSW FA)
JDP	0x19	added: Factory erase/progam mode for JDP Dev3+ Cut 2 and JDP Dev4 Cut 2b

Name	Length	Fmt	Description
			40_H: HSM state is not active **41_H:** EcuSecu state is not active **42_H:** SSM projects: address information for PAV area not available **50_H:** PKCS11: init error **51_H:** PKCS11: open session error **52_H:** PKCS11: create obj. error (certificate) **53_H:** PKCS11: verify init error **54_H:** PKCS11: verify error **55_H:** PKCS11: digest init **56_H:** PKCS11: digest update **57_H:** PKCS11: digest final **FF_H:** EcuSecuSrv not found or application ECU (SSM) **PKCS11: create obj. error** used certificate can cause an "create obj. error" if it not belong to the project.
TP	**1**	**xx**	**TP status** 0x00 - no tuningprotection / HSM not active

Bit(s)	Description
0	**1: RBSN set**
1	**1: PFlash protections active**
2	**1: HSM is activ**
3	**1: HSM flash protections active**
4	**1: EcuSecu is activ**
7:5	**not used**

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)

Code	Description
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**Address for flash type at sub command 0x01 out of range**
0xA0	**flash type not supported**
0xA4	**to less data received for programming range**
0xA6	**Invalid signature length**
0xA7	**Invalid address range, not found in signature file**
0xB0	**Command not allowed**
0xB1	**Last command not finished successfully, flash driver state error**
0xC0	**Certificate invalid. Internal ID not correct**
0xC1	**Certificate to long. Internal length greater 1 KB**
0xC2	**HEX file of certificate to long. Length greater than internal length information or buffer limit reached**
0xFD	**special memory range not correct selected (BootCtrl, EcuSecu(if active HSM or TP))**
0xFF	**Invalid address range**

TSW_FB - Tuning protection

Test step description

This teststep is used to activate the tuning protection, readout tuning-protection status and unlock the debugger access temporary.

Note

Tuning protection level set with this command can be undone!

Test step commands

.

00 - read ECU unique numbers

Read unique numbers: chip unique number from original area and PAV area, RB serial number from PAV area.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	FB	00	01

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	FB	xx	xx	01	...

Parameters

response of command "read ECU unique numbers"

6	+0	+32	+64
(1..n*)	**UCNUM**	**UCNUMPV**	**RBSN**

Details

response of command "read ECU unique numbers"

Name	Length	Fmt	Description
UCNUM	**32**	**xx..**	**Controler unique number** Chip id is 16 Bytes and left aligned.
UCNUMPV	**32**	**xx..**	**Controler unique number from PAV area**
RBSN	**12**	**xx..**	**RB serial number from PAV area**

.

01 - Activate tuning-protection

This command is used to activate the tuning-protection components.

Note

The configured features becomes operative after reset of ECU.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	FB	01	01	...

Parameters

Configuration flags

5	+0	+1	+2	+3
(1*)	**FLASH**	**FLS_OPT**	**HSM**	**HSM_OPT**

...	+4
(1*)	**DBG**

Robert Bosch serial number

10	+0
(1*)	**BOSCHSN**

Details

Configuration flags

Name	Length	Fmt	Description
FLASH	**1**	**xx**	**Flash configuration flags** **Bit(s)** / **Description** **7** — **not used, must be set to 0** **6** — **not used, must be set to 0** **5** — **not used, must be set to 0** **4** — **not used, must be set to 0** **3** — **not used, must be set to 0** **2** — **Activate PFlash OTP sectors (selected by infoblock configuration)** **1** — **Activate PFlash write and debugger protection** **0** — **Write RB serial number**
FLS_OPT	**1**	**xx**	**Flash optional configuration flags** **Bit(s)** / **Description** **7** — **not used, must be set to 0** **6** — **not used, must be set to 0** **5** — **not used, must be set to 0** **4** — **not used, must be set to 0**

Name	Length	Fmt	Description		
				Bit(s)	**Description**
				3	**not used, must be set to 0**
				2	**not used, must be set to 0**
				1	**not used, must be set to 0**
				0	**Activate TP watchdog**
HSM	**1**	**xx**	**HSM configuration flags**		
				Bit(s)	**Description**
				7	**not used, must be set to 0**
				6	**not used, must be set to 0**
				5	**not used, must be set to 0**
				4	**not used, must be set to 0**
				3	**Activate HSM OTP sectors**
				2	**Set HSM data Flash exclusive**
				1	**Set HSM code Flash exclusive**
				0	**Activate HSM boot**
HSM_OPT	**1**	**xx**	**HSM optional configuration flags**		
				Bit(s)	**Description**
				7	**not used, must be set to 0**
				6	**not used, must be set to 0**
				5	**not used, must be set to 0**
				4	**not used, must be set to 0**
				3	**not used, must be set to 0**
				2	**not used, must be set to 0**
				1	**Activate IO Pins for HSM**
				0	**Activate destructive debug protection**
DBG	**1**	**xx**	**Debugger configuration flags**		
				Bit(s)	**Description**
				7	**not used, must be set to 0**
				6	**not used, must be set to 0**
				5	**not used, must be set to 0**
				4	**not used, must be set to 0**
				3	**not used, must be set to 0**
				2	**not used, must be set to 0**
				1	**not used, must be set to 0**
				0	**Activate Debugger Protection**

Robert Bosch serial number

Name	Length	Fmt	Description
BOSCHSN	**12**	**xx..**	**Bosch serial number**

Name	Length	Fmt	Description
			Robert Bosch serial number to be written in PAV area (FLASH Bit0 must be set).

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	FB	xx	xx	01

.

02 - Tuning protection status

This command is used to get the CPU's tuningprotection status. The BOSCH serial number and chip serial number are read from TPROT-PAV flash area.

Note

The configured features become operative after reset of ECU.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	FB	02	01

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	FB	xx	xx	01	...

Parameters

Response Tuning Protection Status request.

6 (1..n*)	+0	+1	+2	+3
	FLASH	**FLS_OPT**	**HSM**	**HSM_OPT**

... (1..n*)	+4
	DBG

Response Tuning Protection Status request.

11 (1..n*)	+0	+32
	CHIPID	**BOSCHSN**

Details

Response Tuning Protection Status request.

Name	Length	Fmt	Description
FLASH	**1**	**xx**	**Flash configuration flags** please, see cmd 01
FLS_OPT	**1**	**xx**	**Flash optional configuration flags** please, see cmd 01
HSM	**1**	**xx**	**HSM configuration flags** please, see cmd 01
HSM_OPT	**1**	**xx**	**HSM optional configuration flags** please, see cmd 01
DBG	**1**	**xx**	**Debugger configuration flags** please, see cmd 01

Response Tuning Protection Status request.

Name	Length	Fmt	Description
CHIPID	**32**	**xx..**	**chip-id: Value is read from TPROT PAV flash area and not from CPU registers.**
BOSCHSN	**12**	**xx..**	**Bosch serial number**

.

03 - Internal Tuning protection status

This command is used to get the CPU's tuningprotection extended status information. Helpfull for failure analysis. You should use CMD 02 in normal usecase.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	FB	03	01

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	FB	xx	xx	01	...

Parameters

Response Tuning Protection Status request.

6 (1..n*)	+0	+1	+2	+3
	FLASH	**FLS_OPT**	**HSM**	**HSM_OPT**

... (1..n*)	+4	+5
	DBG	**INT**

Details

Response Tuning Protection Status request.

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>FLASH</td><td>1</td><td>xx</td><td>Flash configuration flags
please, see cmd 01</td></tr>
<tr><td>FLS_OPT</td><td>1</td><td>xx</td><td>Flash optional configuration flags
please, see cmd 01</td></tr>
<tr><td>HSM</td><td>1</td><td>xx</td><td>HSM configuration flags
please, see cmd 01</td></tr>
<tr><td>HSM_OPT</td><td>1</td><td>xx</td><td>HSM optional configuration flags
please, see cmd 01</td></tr>
<tr><td>DBG</td><td>1</td><td>xx</td><td>Debugger configuration flags
please, see cmd 01</td></tr>
<tr><td>INT</td><td>4</td><td>xxxxxxxx</td><td>Internal TP states. Only for troubleshooting
Additional information about the internal states of the tuning protection.
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>31:7</td><td>not used</td></tr>
<tr><td>6</td><td>UCB_HSM Confirmed</td></tr>
<tr><td>5</td><td>UCB_DBG Confirmed</td></tr>
<tr><td>4</td><td>Reserved</td></tr>
<tr><td>3</td><td>UCB_OTP Confirmed</td></tr>
<tr><td>2</td><td>UCB_HSMCOTP Confirmed</td></tr>
<tr><td>1</td><td>UCB_DFlash Confirmed</td></tr>
<tr><td>0</td><td>UCB_PFlash Confirmed</td></tr>
</table></td></tr>
</table>

.

10 - Debugger unlock challenge

This command is used to get the CPU's tuningprotection challenge from TPROT-PAV flash area to calculate passwords.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	FB	10	01

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	FB	xx	xx	01	...

Parameters

Unlock challenge.

6	+0	+32
(1..n*)	**CHIPID**	**BOSCHSN**

Details

Unlock challenge.

Name	Length	Fmt	Description
CHIPID	**32**	**xx..**	**chip-id: Value is read from TPROT PAV flash area and not from RAM location.**
BOSCHSN	**16**	**xx..**	**Bosch serial number**

.

60 - Clear tuning-protection

This command is used to clear active tuning-protections if set with TSW_FB.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	FB	60	01	...

Parameters

Configuration flags

5	+0	+1	+2	+3	+4
(1*)	**FLASH**	**FLS_OPT**	**HSM**	**HSM_OPT**	**DBG**

Details

Configuration flags

Name	Length	Fmt	Description
FLASH	**1**	**xx**	**Flash configuration flags** Description at Cmd 01.
FLS_OPT	**1**	**xx**	**Flash optional configuration flags** Description at Cmd 01.
HSM	**1**	**xx**	**HSM configuration flags** Description at Cmd 01.
HSM_OPT	**1**	**xx**	**HSM optional configuration flags** Description at Cmd 01.

Name	Length	Fmt	Description
DBG	**1**	**xx**	**Debugger configuration flags** Description at Cmd 01.

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	FB	xx	xx	01

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)

Code	Description
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0xFA	**HAL: Function not implemented**
0xEF	**HAL: Locking or unlocking of flash failed**
0xEE	**HAL: Desired flash area was not erased**
0xED	**HAL: Programming failed**
0xEC	**HAL: Erasing failed**
0xEB	**HAL: Target DCF address not aligned**
0xEA	**HAL: Range to big**
0xE9	**HAL: Compare failed**
0xE8	**HAL: HSM already active**
0xE7	**HAL: UCB for HSMCOTP activation failed**
0xE6	**HAL: no HSM code found, activation aborted**
0xE5	**HAL: Serial Number could not be written**
0xE4	**HAL: Passwords could not be written**
0xE3	**HAL: DCF start/stop pattern could not be written**
0xE2	**HAL: End address of Block could not be calculated**
0xE1	**HAL: Two Infoblocks are in the same Memory Block**
0xE0	**HAL: Infoblock is not in an valid Flash Block**
0xDF	**HAL: HSM Inhibit could not be written**
0xDE	**HAL: HSM alternate Flash interface could not be written**
0xDD	**HAL: HSM Exclusive could not be written**
0xDC	**HAL: OTP Low could not be written**
0xDB	**HAL: OTP 256 could not be written**
0xDA	**HAL: Lifecycle could not be written**
0xD9	**HAL: Censoring could not be activated**
0xD8	**HAL: SN needed to enable LOCKs**
0xD7	**HAL: Enabling Read-Write-Protection failed**
0xD6	**HAL: Enabling UTEST protection failed**
0xD5	**HAL: Enabling Debugger lock failed**
0xD4	**HAL: writing unused lock failed**
0xD3	**HAL: Enabling TDM failed**
0xD2	**HAL: Enabling Destructive Debug failed**
0xD1	**HAL: Enabling Tuning protection watchdog failed**
0xD0	**HAL: HSM needed for HSM Exclusive**
0xCF	**HAL: Enabling Tp Level Lock failed**
0xCE	**HAL: Writing the dummy HSM ROM Keys failed**

Code	Description
0xCD	HAL: Writing BAF Callback failed
0xCB	HAL: HSM needed for HSM OTP
0xCA	HAL: EcuSecu needed for RWP
0xC9	HAL: Enabling HSM Pins needs HSM Core activation
0xB0	HAL: UCB not supported
0xAF	HAL: UCB has no free Set
0xAE	HAL: No information of PAV address found in IB
0x97	HAL: HSM activation already done
0x96	HAL: HSMPINS can not be activated if HSM is active
0x8F	HAL: HSM Exclusive could not be written
0x8E	HAL: HSM doesn't implement API needed for UTEST / UCB access
0x8D	HAL: HSM doesn't write DCF
0x8C	HAL: HSM doesn't write passwd
0x8B	HAL: HSM HSM doesn't write lifecycle
0x8A	HAL: HSM doesn't write PAV Area
0x71	HAL: TP Status could not be detected. Use "03 - Internal Tuning protection status"
0x70	HAL: requested level not implemented
0x91	EcuSecu Lib error
0x92	BootCtrl to old
0xA0	To weaken the protection not allowed

TSW_FC - Tuning protection

Test step description

This teststep is used to activate the tuning protection, readout tuning-protection status and unlock the debugger access temporary.

Note

Tuning protection level set with this command can NOT be undone!

Test step commands

.

00 - read ECU unique numbers

Read unique numbers: chip unique number from original area and PAV area, RB serial number from PAV area.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	FC	00	01

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	FC	xx	xx	01	...

Parameters

response of command "read ECU unique numbers"

6	+0	+32	+64
(1..n*)	**UCNUM**	**UCNUMPV**	**RBSN**

Details

response of command "read ECU unique numbers"

Name	Length	Fmt	Description
UCNUM	**32**	**xx..**	**Controler unique number** Chip id is 16 Bytes and left aligned.
UCNUMPV	**32**	**xx..**	**Controler unique number from PAV area**
RBSN	**12**	**xx..**	**RB serial number from PAV area**

.

01 - Activate tuning-protection

This command is used to activate the tuning-protection components.

Note
The configured features becomes operative after reset of ECU.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	FC	01	01	...

Parameters

Configuration flags

5	+0	+1	+2	+3
(1*)	**FLASH**	**FLS_OPT**	**HSM**	**HSM_OPT**

...	+4
(1*)	**DBG**

Robert Bosch serial number

10	+0
(1*)	**BOSCHSN**

Details

Configuration flags

Name	Length	Fmt	Description
FLASH	**1**	**xx**	**Flash configuration flags** **Bit(s) – Description** **7** – **not used, must be set to 0** **6** – **not used, must be set to 0** **5** – **not used, must be set to 0** **4** – **not used, must be set to 0** **3** – **not used, must be set to 0** **2** – **Activate PFlash OTP sectors (selected by infoblock configuration)** **1** – **Activate PFlash write and debugger protection** **0** – **Write RB serial number**
FLS_OPT	**1**	**xx**	**Flash optional configuration flags** **Bit(s) – Description** **7** – **not used, must be set to 0** **6** – **not used, must be set to 0** **5** – **not used, must be set to 0** **4** – **not used, must be set to 0**

Name	Length	Fmt	Description
			Bit(s) / Description: 3: not used, must be set to 0 2: not used, must be set to 0 1: not used, must be set to 0 0: Activate TP watchdog
HSM	1	xx	HSM configuration flags Bit(s) / Description: 7: not used, must be set to 0 6: not used, must be set to 0 5: not used, must be set to 0 4: not used, must be set to 0 3: Activate HSM OTP sectors 2: Set HSM data Flash exclusive 1: Set HSM code Flash exclusive 0: Activate HSM boot
HSM_OPT	1	xx	HSM optional configuration flags Bit(s) / Description: 7: not used, must be set to 0 6: not used, must be set to 0 5: not used, must be set to 0 4: not used, must be set to 0 3: not used, must be set to 0 2: not used, must be set to 0 1: Activate IO Pins for HSM 0: Activate destructive debug protection
DBG	1	xx	Debugger configuration flags Bit(s) / Description: 7: not used, must be set to 0 6: not used, must be set to 0 5: not used, must be set to 0 4: not used, must be set to 0 3: not used, must be set to 0 2: not used, must be set to 0 1: not used, must be set to 0 0: Activate Debugger Protection

Robert Bosch serial number

Name	Length	Fmt	Description
BOSCHSN	12	xx..	Bosch serial number

Name	Length	Fmt	Description
			Robert Bosch serial number to be written in PAV area (FLASH Bit0 must be set).

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	FC	xx	xx	01

.

02 - Tuning protection status

This command is used to get the CPU's tuningprotection status. The BOSCH serial number and chip serial number are read from TPROT-PAV flash area.

Note

The configured features become operative after reset of ECU.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	FC	02	01

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	FC	xx	xx	01	...

Parameters

Response Tuning Protection Status request.

6	+0	+1	+2	+3
(1..n*)	**FLASH**	**FLS_OPT**	**HSM**	**HSM_OPT**

...	+4
(1..n*)	**DBG**

Response Tuning Protection Status request.

11	+0	+32
(1..n*)	**CHIPID**	**BOSCHSN**

Details

Response Tuning Protection Status request.

Name	Length	Fmt	Description
FLASH	**1**	**xx**	**Flash configuration flags**

Name	Length	Fmt	Description
			please, see cmd 01
FLS_OPT	**1**	**xx**	**Flash optional configuration flags** please, see cmd 01
HSM	**1**	**xx**	**HSM configuration flags** please, see cmd 01
HSM_OPT	**1**	**xx**	**HSM optional configuration flags** please, see cmd 01
DBG	**1**	**xx**	**Debugger configuration flags** please, see cmd 01

Response Tuning Protection Status request.

Name	Length	Fmt	Description
CHIPID	**32**	**xx..**	**chip-id: Value is read from TPROT PAV flash area and not from CPU registers.**
BOSCHSN	**12**	**xx..**	**Bosch serial number**

.

03 - Internal Tuning protection status

This command is used to get the CPU's tuningprotection extended status information. Helpfull for failure analysis. You should use CMD 02 in normal usecase.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	FC	03	01

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	FC	xx	xx	01	...

Parameters

Response Tuning Protection Status request.

6	+0	+1	+2	+3
(1..n*)	**FLASH**	**FLS_OPT**	**HSM**	**HSM_OPT**

...	+4	+5
(1..n*)	**DBG**	**INT**

Details

Response Tuning Protection Status request.

Name	Length	Fmt	Description
FLASH	**1**	**xx**	**Flash configuration flags**

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>please, see cmd 01</td></tr>
<tr><td>FLS_OPT</td><td>1</td><td>xx</td><td>Flash optional configuration flags
please, see cmd 01</td></tr>
<tr><td>HSM</td><td>1</td><td>xx</td><td>HSM configuration flags
please, see cmd 01</td></tr>
<tr><td>HSM_OPT</td><td>1</td><td>xx</td><td>HSM optional configuration flags
please, see cmd 01</td></tr>
<tr><td>DBG</td><td>1</td><td>xx</td><td>Debugger configuration flags
please, see cmd 01</td></tr>
<tr><td>INT</td><td>4</td><td>xxxxxxxx</td><td>Internal TP states. Only for troubleshooting
Additional information about the internal states of the tuning protection.
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>31:7</td><td>not used</td></tr>
<tr><td>6</td><td>UCB_HSM Confirmed</td></tr>
<tr><td>5</td><td>UCB_DBG Confirmed</td></tr>
<tr><td>4</td><td>Reserved</td></tr>
<tr><td>3</td><td>UCB_OTP Confirmed</td></tr>
<tr><td>2</td><td>UCB_HSMCOTP Confirmed</td></tr>
<tr><td>1</td><td>UCB_DFlash Confirmed</td></tr>
<tr><td>0</td><td>UCB_PFlash Confirmed</td></tr>
</table></td></tr>
</table>

.

10 - Debugger unlock challenge

This command is used to get the CPU's tuningprotection challenge from TPROT-PAV flash area to calculate passwords.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	FC	10	01

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	FC	xx	xx	01	...

Parameters

Unlock challenge.

<table>
<tr><th rowspan="2">6
(1..n*)</th><th>+0</th><th>+32</th></tr>
<tr><td>CHIPID</td><td>BOSCHSN</td></tr>
</table>

Details

Unlock challenge.

Name	Length	Fmt	Description
CHIPID	**32**	**xx..**	**chip-id: Value is read from TPROT PAV flash area and not from RAM location.**
BOSCHSN	**16**	**xx..**	**Bosch serial number**

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	TSW execution was successful
0xFA	HAL: Function not implemented
0xEF	HAL: Locking or unlocking of flash failed
0xEE	HAL: Desired flash area was not erased
0xED	HAL: Programming failed
0xEC	HAL: Erasing failed
0xEB	HAL: Target DCF address not aligned
0xEA	HAL: Range to big
0xE9	HAL: Compare failed
0xE8	HAL: HSM already active
0xE7	HAL: UCB for HSMCOTP activation failed
0xE6	HAL: no HSM code found, activation aborted
0xE5	HAL: Serial Number could not be written
0xE4	HAL: Passwords could not be written
0xE3	HAL: DCF start/stop pattern could not be written
0xE2	HAL: End address of Block could not be calculated
0xE1	HAL: Two Infoblocks are in the same Memory Block
0xE0	HAL: Infoblock is not in an valid Flash Block
0xDF	HAL: HSM Inhibit could not be written
0xDE	HAL: HSM alternate Flash interface could not be written
0xDD	HAL: HSM Exclusive could not be written
0xDC	HAL: OTP Low could not be written
0xDB	HAL: OTP 256 could not be written
0xDA	HAL: Lifecycle could not be written
0xD9	HAL: Censoring could not be activated
0xD8	HAL: SN needed to enable LOCKs
0xD7	HAL: Enabling Read-Write-Protection failed
0xD6	HAL: Enabling UTEST protection failed
0xD5	HAL: Enabling Debugger lock failed
0xD4	HAL: writing unused lock failed
0xD3	HAL: Enabling TDM failed
0xD2	HAL: Enabling Destructive Debug failed
0xD1	HAL: Enabling Tuning protection watchdog failed
0xD0	HAL: HSM needed for HSM Exclusive
0xCF	HAL: Enabling Tp Level Lock failed
0xCE	HAL: Writing the dummy HSM ROM Keys failed
0xCD	HAL: Writing BAF Callback failed
0xCB	HAL: HSM needed for HSM OTP

Code	Description
0xCA	HAL: EcuSecu needed for RWP
0xC9	HAL: Enabling HSM Pins needs HSM Core activation
0xB0	HAL: UCB not supported
0xAF	HAL: UCB has no free Set
0xAE	HAL: No information of PAV address found in IB
0x97	HAL: HSM activation already done
0x96	HAL: HSMPINS can not be activated if HSM is active
0x8F	HAL: HSM Exclusive could not be written
0x8E	HAL: HSM doesn't implement API needed for UTEST / UCB access
0x8D	HAL: HSM doesn't write DCF
0x8C	HAL: HSM doesn't write passwd
0x8B	HAL: HSM HSM doesn't write lifecycle
0x8A	HAL: HSM doesn't write PAV Area
0x71	HAL: TP Status could not be detected. Use "03 - Internal Tuning protection status"
0x70	HAL: requested level not implemented
0x91	EcuSecu Lib error
0x92	BootCtrl to old
0xA0	To weaken the protection not allowed

TSW_FE - Single byte programming

Test step description

Date: 02.02.2016

Created: 18.01.2017 09:12.

This TSW downloads and writes a few bytes to a Flash address or copies a few bytes from a RAM buffer to Flash.

Please keep in mind, that writing to flash is only possible to whole pages. The length of a page in DFlash is 8 bytes, in PFlash 32 bytes.

Changing of flash (erasing or programming) is not allowed inside the info block chain.

Do not write to restricted areas (tbd). Use TSW_FC instead!

Also do not write to EEPROM emulation areas, because those areas have a special format. Use TSW_EE instead!

Test step commands

.

01 - Download Bytes

This command downloads some bytes and programs them at a given address into Flash memory. For large data, you can call this command with same parameters again and the data will be appended!

Note

Be careful with Specific Flash Area writing function, some places are OTP and non-erasable! It can harm the device!

Table 1. Specific flash area

IFX:	UCB

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	FE	01	01	...

Parameters

Parameter set for 'Download Bytes'

5	+0	+1	+5	+9
(1*)	**MEM**	**DEST**	**LEN**	**DATA**

Details

Parameter set for 'Download Bytes'

Name	Length	Fmt	Description
MEM	**1**	**xx**	**Memory Type** Specifies the memory that the data shall be copied to. **0_H:** Copy to FLASH **1_H:** Copy to RAM buffer **2_H:** Copy to SPECIFIC FLASH AREA
DEST	**4**	**xxxxxxxx**	**Destination address** If MEM_TYPE is "Flash", use absolute addresses. If MEM_TYPE is "RAM buffer" use relative address within the buffer (buffer size: 128 bytes)
LEN	**4**	**xxxxxxxx**	**Download length** Amount of Bytes to download
DATA	**1**	**xx..**	**Data to download**

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	FE	xx	xx	01	...

Parameters

Response set for Download Bytes

...	+0
(1*)	**EXP**

Details

Response set for Download Bytes

Name	Length	Fmt	Description
EXP	**4**	**xxxxxxxx**	**expected data** Number of bytes expected to complete the request. Is 0x00 if all data was received!

.

02 - Copy Bytes

This command copies bytes from the RAM buffer and programms them at a given address into Flash memory.

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	FE	02	01	...

Parameters

Parameter set for 'Copy Bytes'

5	+0	+4	+8
(1..n*)	**SOURCE**	**DEST**	**NUM**

Details

Parameter set for 'Copy Bytes'

Name	Length	Fmt	Description
SOURCE	**4**	**xxxxxxxx**	**Source offset** Offset address within RAM buffer (0..127)
DEST	**4**	**xxxxxxxx**	**Destination address in Flash** Absolute address in Flash
NUM	**4**	**xxxxxxxx**	**Number of bytes to copy** max: 128 (=size of RAM buffer)

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	FE	xx	xx	01

.

03 - Get Internals

This command returns internals of the TSW. E.g. the absolut address on runtime and the length of the RAM buffer. The purpose is, to use this address for CRC calculation with use of TSW_E4.

Command

1	2	3	4
SVC	**TSW**	**CMD**	**VER**
80	FE	03	01

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	FE	xx	xx	01	...

Parameters

Response set for Get Internals

6	+0	+4
(1*)	**ADDR**	**LEN**

Details

Response set for Get Internals

Name	Length	Fmt	Description
ADDR	**4**	**xxxxxxxx**	**RAM buffer address on runtime**

Name	Length	Fmt	Description
LEN	**4**	**xxxxxxxx**	**RAM buffer length in bytes**

.

04 - Erase sector by address

This command erases absolut sectors in flash. Additionaly a blank check before and/or after erasing could be activated. If blank check before erase is activated and the specified sector is already erased, no new erasing is triggered.

ATTENTION: Do not use this command if you are not REALLY sure what you are doing! Erasing of the wrong sector could cause permanently damage of the hardware!

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	FE	04	01	...

Parameters

Parameter set for 'Erase sector by address'

... (1*)	+0 CFG

Parameter set for 'Erase sector by address'

7 (1..n*)	+0 ADDR	+4 LEN

Details

Parameter set for 'Erase sector by address'
General parameter

Name	Length	Fmt	Description
CFG	**2**	**xxxx**	**General configuration**
			Bit(s) / **Description**
			15:3 — **Reserved for future, must be zero**
			2 — **Blank check before erase** **0_B:** force erase **1_B:** erase only if not erased
			1 — **Blank check after erase** **0_B:** no blank check **1_B:** blank check
			0 — **Skip erase** **0_B:** normal erase

Name	Length	Fmt	Description	
			Bit(s)	**Description**
				1_B: only blank checks

Parameter set for 'Erase sector by address'
Repeatable parameter

Name	Length	Fmt	Description
ADDR	**4**	**xxxxxxxx**	**Base address of sector to erase** Absolute base address of sector in Flash
LEN	**4**	**xxxxxxxx**	**Length of sector to erase** Exact length of this sector in Flash

Response

1	2	3	4	5	6
SVC	**TSW**	**ESVC**	**ETSW**	**VER**	**...**
C0	FE	xx	xx	01	...

Parameters

Response set for 'Erase sector by address'

...	+0
(1..n*)	**RC**

Details

Response set for 'Erase sector by address'

Name	Length	Fmt	Description
RC	**1**	**xx**	**Return Code** For every set a return code is responded

Value	Meaning
0x00	Sector erasing successful
0xE5	Specified sector length too low
0xE6	Specified sector length too high
0xE7	Unknown sector base address
0xEB	Start address not aligned on page size
0xEC	Erasing failed
0xEE	Flash area is not erased (blank)

.

05 - Create complementary block

This command is mainly used to create a complementary block in flash memory derived from an already existing block in flash memory. A block in this context means a memory area defined by a startaddress (SRC or DST) and a length (NUM).

Hint: If the destination block is in flash, ensure that this block is page aligned!

> **Hint:** If the destination block is in flash, the internal RAM buffer is used as intermediate buffer. Therefore its contens are probably changed.
>
> This RAM buffer is also used in command 02. It's size are 128 bytes (offset 0 .. 127).

Command

1	2	3	4	5
SVC	**TSW**	**CMD**	**VER**	**...**
80	FE	05	01	...

Parameters

Parameter set for 'Create complementary block'

5	+0	+2	+6	+10
(1..n*)	**CFG**	**SRC**	**DST**	**NUM**

Details

Parameter set for 'Create complementary block'

<table>
<tr><th>Name</th><th>Length</th><th>Fmt</th><th>Description</th></tr>
<tr><td>CFG</td><td>2</td><td>xxxx</td><td>General configuration
<table>
<tr><th>Bit(s)</th><th>Description</th></tr>
<tr><td>15</td><td>Destination

0_B: Destination is flash
1_B: Destination is RAM buffer</td></tr>
<tr><td>14</td><td>Source

0_B: Source is flash
1_B: Source is RAM buffer</td></tr>
<tr><td>13:0</td><td>Reserved for future, must be zero</td></tr>
</table></td></tr>
<tr><td>SRC</td><td>4</td><td>xxxxxxxx</td><td>Source address

Absolute address in flash or
offset in RAM buffer (depends on CFG)</td></tr>
<tr><td>DST</td><td>4</td><td>xxxxxxxx</td><td>Destination address

Absolute address in flash or
offset in RAM buffer (depends on CFG)</td></tr>
<tr><td>NUM</td><td>4</td><td>xxxxxxxx</td><td>Number of bytes

Number of bytes to convert into a block</td></tr>
</table>

Response

1	2	3	4	5
SVC	**TSW**	**ESVC**	**ETSW**	**VER**
C0	FE	xx	xx	01

Response codes

Each testsoftware command responds one of the following standard response codes (**ESVC**) to indicate its execution status:

Code	Description
0x00	**TSW execution was successful**
0x01	**Command interface version not supported.**
0x02	**Invalid argument(s)**
0x03	**Wrong number of arguments**
0x04	**Extended TSW Errorcode** (see byte **ETSW** for details)
0x05	**Invalid HEX file loaded**
0x06	**Invalid CRC of HEX file**
0x07	**Wrong HAL for actual Device loaded.**
0x10	**Error in HAL-ADC** (see byte **ETSW** for details)
0x11	**Error in HAL-SENT** (see byte **ETSW** for details)
0x12	**Error in HAL-I2C** (see byte **ETSW** for details)
0x13	**Error in HAL-SPI** (see byte **ETSW** for details)
0x14	**Error in HAL-GPIO** (see byte **ETSW** for details)
0x15	**Error in HAL-MSC** (see byte **ETSW** for details)
0x16	**Error in HAL-CPU** (see byte **ETSW** for details)
0x17	**Error in HAL-PWM** (see byte **ETSW** for details)
0x18	**Error in HAL-CAN** (see byte **ETSW** for details)
0x19	**Error in HAL-FLX** (see byte **ETSW** for details)
0x1A	**Error in HAL-LIN** (see byte **ETSW** for details)
0x1B	**Error in HAL-ASC** (see byte **ETSW** for details)
0x1C	**Error in HAL-ETH** (see byte **ETSW** for details)
0x1D	**Error in HAL-C2C** (see byte **ETSW** for details)
0x20	**Error in HAL-STM** (see byte **ETSW** for details)
0x21	**Error in HAL-OCDS** (see byte **ETSW** for details)
0x22	**Error in HAL-FLS** (see byte **ETSW** for details)
0x23	**Error in HAL-AST** (see byte **ETSW** for details)
0x24	**Error in HAL-ECUSECU** (see byte **ETSW** for details)
0x25	**Error in HAL-GTM** (see byte **ETSW** for details)
0x30	**Error in HAL-MEMLAY** (see byte **ETSW** for details)

Each test step command may respond one of the following download-testsoftware response codes (ETSW) to indicate its execution status. This byte is only to be validated if byte ESVC indicates an extended error code (0x04) or a HAL error (0x10, 0x11..):

Code	Description
0x00	**TSW execution was successful**
0x01	**TSW: Too much data received with this frame**
0xEF	**HAL_FLS: Locking or unlocking of flash failed**
0xEE	**HAL_FLS: Desired flash area was not erased**

Code	Description
0xED	**HAL_FLS: Programming failed**
0xEC	**HAL_FLS: Erasing failed**
0xEB	**HAL_FLS: Start address not aligned on page size**
0xEA	**HAL_FLS: Desired range is too big for flash**
0xE9	**HAL_FLS: Compare failed**
0xE8	**HAL_FLS: Flash block is not allowed to program/erase**
0xE7	**HAL_FLS: Flash module not found**
0xE6	**HAL_FLS: Sector length too high**
0xE5	**HAL_FLS: Sector length too low**