

Test Report

Integration Test Report CanFblGM

General Information

		Version
Test Specification	Test Specification – GM FBL Diagnostics Tests	1.0
Requirement reference	GM SLP6 RequirementsSpec_SLP6.doc	
Test Person	Joe Benczarski	
Test Proceeding Date	2017-07-12	
Test Report Recipients	Nexteer Automotive Corporation, Vector North America, Inc., Vector Informatik GmbH	

Test environment

Component	Description	Version / Derivative
Target System	RH850	R7F701363
Generation Tool	GENy	01.04.0050.0000
Compiler	Green Hills	2015.1.7
Test Database	demo_sw.dbc	-
Frequency / Oscillator	16MHz / 160MHz	-
Memory	2 MB Code Flash / 128 KB RAM	-
Communication channels	CAN0	-

Component under test

Component		Configuration	Version	Date
GM FBL		Standard	6.04.03	2017-05-29

Overall Test result	OK
Details / hints	-

Revision history of template

Version	Editor	Description
1.0.0 dated 2015-03-02	A. Wenckebach	Create template
1.0.1 dated 2015-07-20	S. Schleith	Corrections and additions
1.0.2 dated 2016-04-01	A. Wenckebach	Add test case for Basic NVM configuration
1.0.3 dated 2016-07-28	S. Loos	Added Updater test cases
1.1.0 dated 2016-11-21	A. Wenckebach	Added TJA1145 test cases, improvements
1.1.1 dated 2016-11-25	T. Müller	Added XCP test cases
1.1.2 dated 2017-01-19	T. Müller	Removed test 21.04.00, added exclude element
1.2.0 dated 2017-03-29	S. Loos / A. Wenckebach	Removed obsolete Testcases. Improvements.
1.3.0 dated 2017-07-06	A. Wenckebach	Improvements

Test Report distribution

After performing the test, this test report and procedure test results must be filed in project folder _Doc.

An electronic copy must be put in the ALMplus delivery.

Abbreviations / Glossary

Abbreviation	Description
OK	The obtained result was in accordance to the expected.
N. a.	Not applicable. The selected test configuration excludes this test.
N. t.	Not tested. The test was not performed.
Failed	The obtained result was not in accordance to the specification.

Table of contents

1 Detailed description of test environment	1
2 Setup and General Test Proceeding	1
2.1 DUT Preparation Phase	1
2.2 Check Test Entry Criteria.....	1
2.3 Specify Tests to be performed during the Integration / Module Test.....	1
2.4 Test End Phase	1
2.5 Post Test Phase.....	1
3 Test Category	2
3.1 Requirement tests	2
3.1.1 Set-1 Tests	2
3.1.2 Set-2 Tests	4
3.1.3 Set-3 tests.....	6
3.1.4 Set-4 tests.....	8
3.1.5 Set-5 tests.....	10
3.1.6 Set-6 tests.....	12
3.1.7 Watchdog Timing Measurement.....	15
4 Compression Module.....	16
5 Updater	17
6 Tja1145 PN Transceiver integration.....	18
7 XCP	19

1 Detailed description of test environment

For details on compiler options, module version please check

DeliveryDescription_CBD1500635.html.

2 Setup and General Test Proceeding

2.1 DUT Preparation Phase

Assemble test system. Build application. Prepare data files for individual test-cases.

2.2 Check Test Entry Criteria

Are necessary resources available?

No issues on delivery?

Compiler options as required in place?

2.3 Specify Tests to be performed during the Integration / Module Test

Evaluate criteria to select/deselect test

A) No XCP delivered

⇒ Exclude 23.00.00 – 23.02.00

Test End Phase

For each test: retrieve and store test results/configuration.

2.4 Post Test Phase

Place the Test Report into the ALMplus delivery folder ..\Dxx_doc\30_TestReports

Distribute Test Report(s) to customer.

3 Test Category

3.1 Requirement tests

3.1.1 Set-1 Tests

Set the configuration as follows. :

GENy “FblDrvCan_<HW>” / DaVinci “Fbl”:

- Enable “Stay in Boot” setting
- Disable FblStart Function
To avoid compile error, comment out:
 - in fbl_apnv.c call to */* ApplFblReadSharedRam(..); */*
(Reprogramming from Appl now is only possible using Stay in boot feature)
 - in comdat.h error */* # error "Decide where to read and set tester target address" */*
- Set Project State to “Integration”
- Disable “Gap Fill” (for GENy in FblCan_14229_Gm->GM FBL)
- Prepared User Config File is “**MandatoryDeliveryPreconfig_Tsi_Set1-2.cfg**”, adapt it to your environment

GENy “FblCan_14229_Gm->GM FBL” / DaVinci “FblGm” configuration Settings:

- Enable “RAM Integrity Check”
 - Set Start-/End-Address to appropriate value
- Enable “ROM Integrity Check”
 - Enter PMA Address length pool data (compare 06.01.00)

Generate the new configuration files, compile the FBL, and load the new DUT onto the target hardware.

Check IntegrationTests.xlsx for to be used Download containers. Check Gm_Slp6_Tsi.cfg for tabs 08.05.00 and 21.01.00 for below tests.

Test No	Test Name	Test Entry	Expected result	Result
08.05.00	Response Timing	DID \$7E7E need to be implemented in ApplFblReadDataBy-Identifier(). (Use <i>gen_tsi_standard.bat</i> to add it to <i>fbl_apdi.c</i>)	Canoe trace shows 1 (after <100ms) and 3 further response pendings (each <15s) before the final response is sent. Difference between Response pendings is close to (kFblDiagTimeP2Star / 2) (~+-2% deviation), first comes after (kFblDiagTimeP2 / 2).	N.t.

16.02.00	Stay-in-Boot Mode Enabled	Download an application. Verify it is running. vFlash: set "Force Boot Mode". Download again an application. Wait. Cause a reset to jump into bootloader. The bootloader responds to the ping messages and the download starts.	FBL responds to "ping" message and does not start Operating Software upon reset.	N.t.
21.01.00	Session Timer	Check the Bootloader stays in extended and programming session as long as tester present messages are received, check it resets as soon as no messages are received any more within 5s. <i>Check Canoe tab 20.01.00 for executing the test</i>	Bootloaders stays in active session as long as tester present messages are received. Verify breakpoint in fbl_diag_core->FblEcuReset is hit. Verify correct NRCs are send inside and outside session (check Canoe sequences)	N.t.

Save the configuration and Test results (e.g. use script /_Tsi_Set1/GetResults_AND_cleanup.cmd).

3.1.2 Set-2 Tests

Before performing the following tests, change the configuration used in Set-1 (via GENy) as follows:

All Set-2 Test Cases

General:

- On the Configuration Menu in GENy, select “Edit Channel (channel 0)”. Select both “DUT” and “DUT1” nodes. This will enable the multiple nodes switch
 - Note that multiple nodes addresses additionally are to be configured in several locations
 - through GENy user config (prepared to be 0x45 (handle 0) and 0x46 (handle 1))
 - FblWrapperCom_Can configuration

CAN Message	Direction	Indication/Confirmation Function	Message Object	Nodes	Addressing Mode	Receive Handle
Diag Phys Req	Request	FblCwProcessPhysicalRangeRequest	0	2	NormalFixed	FblRxRange0Hdl
Diag Phys Req	Request	FblCwProcessPhysicalRangeRequest	2	2	Normal	FblRxCanMsg0Hdl
Diag Func Req	Request	FblCwProcessFunctionalRangeRequest	1	1	NormalFixed	FblRxRange1Hdl
Diag Phys Res	Response	FblTpConfirmation	0*	2	NormalFixed	V_NULL

- Additionally edit “Messages”, below “Connections” for request / response messages for different nodes

FblDrvCan_<HW>:

- Enable FblStart Function
- Disable Stay in boot
- Prepared User Config File is **MandatoryDeliveryPreconfig_Tsi_Set1-2.cfg**, adapt it to your environment
- Verify disabled “Gap Fill”

FblCan_14229_Gm->GM FBL configuration Settings:

- Disable “RAM Integrity Check”
- Disable “ROM Integrity Check”

For deliveries that support multiple device-drivers

If your delivery supports multiple devices (such as flash and EEPROM), change the configuration as follows:

- Add Secondary Device Type to configuration, use it in FlashBlocktable.

For deliveries that support multiple Application Files

FblCan_14229_Gm->GM Modules configuration

- Enable “Multiple application support”

Generate the new configuration files, compile the FBL, and load the new DUT onto the target hardware.

Check IntegrationTests.xlsx for to be used Download containers. Check Gm_Slp6_Tsi.cfg for tabs 04.03.00 12.02.00 below tests.

Test No	Test Name	Test Entry	Expected result	Result
04.03.00	Multiple nodes	<p>ApplFblCanParamInit() implemented to select different diagnostic-addresses (for this test e.g. at compile time)</p> <p>Check 11-bit and 29bit response identifiers When sending some diagnostic service to the ECU; depending on initialization in ApplFblCanParamInit() with handle 0 or handle 1. Tester ID shall be F1.</p>	<p>Initialize with Handle 0:</p> <p>Request 29bit: 14DA45F1 Response: 14DAF145 Request 11bit: 645 Response Flow control: 14DAFC45 Response Final: 14DAF145 Request: 14DA46F1 Response: No response! Request 11bit: 646 Response: No response!</p> <p>Initialize with Handle 1:</p> <p>Request: 14DA46F1 Response: 14DAF146 Request 11bit: 646 Response Flow control: 14DAFC46 Response Final: 14DAF146 Request: 14DAF145 Response: No Response! Request 11bit: 645 Response: No response!</p>	N.t.
12.02.00	Data spans multiple Address-Regions	<p>Create Operating S/W GBF file with two address-regions and one or more calibration-file entries. (Download just Operating S/W. Use Set2_12_02_00_-MultiRegion in _TsiCommon to generate files.)</p>	<p>vFlash downloads operating S/W successfully.</p> <p>Read out with CANoe Fbl "Module DIDs" and verify DIDs for programmed modules are responded positively (0xC1 usually 0x11111111/ 0xD1 usually 0x4141).</p> <p>0xC2/D2 (cal1) / C3/D3 (cal2) / C4/D4 (cal3) are answered with negative response (no cal partition programmed).</p>	N.t.
14.02.00	Interrupt Calibration D/L	<p>Only download cals of one partition, thus avoiding starting of the application. Interrupt Calibration download (via power-loss, connection-failure, or reset).</p>	<p>FBL restarts successfully after an interrupted download. FBL allows download to complete after an interrupted download.</p> <p>Read out with CANoe Fbl "12.02.00 DIDs" and verify all 0xC2/D2 (cal1) / C3/D3 (cal2) / C4/D4 (cal3) are positively answered</p>	N.t.
16.02.00	Stay-in-Boot Mode Disabled	<p>Download an application. Verify it is running. vFlash: set "Force Boot Mode".</p> <p>Download again an application. Wait. Cause a reset to jump into bootloader.</p> <p>The bootloader does not respond to the ping messages and the download never starts.</p>	<p>FBL does not respond to "ping" message, and starts Operating Software upon reset.</p> <p>Note: Application then will answer to ping with NRC 31</p>	N.t.

After performing the above tests, save the generated configuration files (all found in the GenData folder), the GENy configuration file (found in the CANdb folder), modified source-files, and generated data files.

3.1.3 Set-3 tests

Before performing the following tests, change the configuration used in Set-2 as follows:

General:

- On the Configuration Menu, select “Edit Channel (channel 0)”. Select only “DUT” node (de-select “DUT1” node). Enter Node Address 0x45 in GENy

FblDrvCan_<Hw>->Memory Configuration:

- Remove Secondary Device Types
- Change Project State to “Production (Reduced Checks)”
 - Keep appNBID/keyNBID Dummy init blocks, e.g. by defining and asking for FBL_FORCE_NV_ELEMENTS_FOR_TEST additionally:

```
#if ( FBL_PROJECT_STATE >= FBL_INTEGRATION ) || defined (FBL_FORCE_NV_ELEMENTS_FOR_TEST)
# if defined(FBL_TEST_KEY_NBID) || defined(FBL_TEST_APP_NBID)
...
```
- Use **MandatoryDeliveryPreconfig_Tsi_Set3_Integrity_Word.cfg**. It changes the configuration by:
 - Adding #define FBL_ENABLE_VERIFY_INTEGRITY_WORD

FblCan_14229_Gm->GM FBL configuration Settings:

- Enable “ROM Integrity Check”, make sure wrong checksum is configured

FblCan_14229_Gm->GM Modules configuration

- Disable “Calibration Module support”
- Disable “Multiple application Module support” if applicable
- In DaVinci enable “Security Class Vendor”

Generate the new configuration files, compile the FBL, and load the new DUT onto the target hardware.

Check IntegrationTests.xlsx for to be used Download containers. Check Gm_Slp6_Tsi.cfg tab 12.13.00 for below tests.

Test No	Test Name	Test Entry	Expected result	Result
12.13.00	Integrity Word, Rom error check	<p>ROM Integrity Error is reported by the Fbl, application downloads still correctly in case Integrity Word option is configured.</p> <p><i>Check Canoe tab 12.13.00 for executing the test step 1.</i></p>	<p>Test step 1: Read BIS via sequencer and interpret the response according to the Expected result</p> <p>Verify Bits state shows Rom Integrity error while in Bootloader (in Canoe send Read BIS sequence from tab 12.13.00). First byte in response is 0x01:</p> <ul style="list-style-type: none">- 22 F0 F2- 62 F0 F2 x1 xx <p>Download single application without calcs and verify correct download</p> <p>Test step 1: Download single appl -> download succeeds</p>	N.t.

After performing the above tests, save the generated configuration files (all found in the GenData folder), and the GENy configuration file (found in the CANDb folder).

3.1.4 Set-4 tests

Before performing the following tests, change the configuration used in Set-3 as follows:

FblDrvCan_<Hw>->Memory Configuration:

- Add Device “Dummy” to FlashBlock Table (Segment Size equal to “Flash” device)
- Add additional real devices, if applicable
- Configure Flashblock that contains Calibration file to “Dummy” device.
- Add fbl_dio.c and fbl_dio.h to the build project (this adds a second device which implements the His “Dummy” Device functions; these simply map to the internal flash device), add fbl_dio.h to fbl_inc.h
- Change Project State to “Integration”
- Use **MandatoryDeliveryPreconfig_Tsi_Set4_Multi_Appp.cfg**. It changes the configuration by:
 - Remove #define FBL_ENABLE_VERIFY_INTEGRITY_WORD

FblCan_14229_Gm->GM Module configuration:

- Enable “Gap Fill”
- Enable “Calibration module support”
- Enable “Multiple Application support”
 - If you do not have a multi processor project, configure this
 - create dummy Logical Block entry for 2nd application
 - Add a flashblock for existing device type to non existing memory (no download will happen to this). Assign it to Logical block 2nd application
 - ApplFblIsValidApp(): Remove check for 2nd application that is present through (FBL_MTAB_NO_OF_BLOCKS > 1)
- In DaVinci disable “Security Class Vendor”

Generate the new configuration files, compile the FBL, and load the new DUT onto the target hardware.

Check IntegrationTests.xlsx for to be used Download containers.

Test No	Test Name	Test Entry	Expected result	Result
11.01.00 08.06.00	Erase during Operating Software D/L	Create Operating S/W and optionally cal files to fill all programmable memory (for each application, if several available). (.._TsiCommon Set4_11_01_00_FillAllAppl) If presence pattern is in conflict, leave Gap for the pattern.	FBL is able to download Demo Application that fills all memory once again. Memory unused by Demo Application contains fill char. Check this with all available drivers (use within Fill All file(s) to all available mem).	N.t.

After performing the above tests, save the generated configuration files (all found in the GenData folder), the GENy configuration file (found in the CANdb folder), and all generated data files.

3.1.5 Set-5 tests

Before performing the following tests, change the configuration used in Set-4 as follows:

FblDrvCan_<Hw> Memory Configuration:

- Remove Device “Dummy” from Memory Configuration and from FlashBlock table

FblDrvCan_<HW>:

- Disable Compression
 - Remove Compression module from the build project (if available)
- Disable “Gap Fill”
- Use **MandatoryDeliveryPreconfig_Tsi_Set5_SBA.cfg**

FblCan_14229_Gm->GM Module configuration:

- Disable “Calibration Modules support”
- Disable “Multiple Application support” if applicable
 - Remove 2nd Dummy application if applicable

Generate the new configuration files, compile the FBL, and load the new DUT onto the target hardware.

Check IntegrationTests.xlsx for to be used Download containers.

Test No	Test Name	Test Entry	Expected result	Result
07.01.00	Multiple Devices / Compiled-in Device-Driver	Create two dummy data files with GM-Header First download the larger, that the smaller module.	Verify Erasure of compiled in driver, e.g. by checking: Data files are successfully downloaded. Memory not used by second and smaller data file is erased.	N.t.
20.10.00	Pipelined Verification	Do application download. During Verification after \$36 reception, the Fbl will need to be able to receive Can frames already. Verify Flow Control responses to 2 nd and following \$36 on Application download come in a time close to minimum response time (e.g. ~<375µs for 500KBaud; should be similar to response time of 01 28 request) Request 1X XX 36 YY YY YY YY YY Response 30 00 00 FF FF FF FF FF	Response time is close to minimum. Check Write window for automatic measurement result (Tester F1 must be used), check for generated warnings.	N.t.

After performing the above tests, save the generated configuration files (all found in the GenData folder), the GENy configuration file (found in the CANdb folder), and all generated data files.

3.1.6 Set-6 tests

Before performing the following tests, change the configuration used in Set-5 as follows:

FblDrvCan_<HW>

- Enable Data Processing -> Compression Mode
 - If no compression module is delivered: Edit fbl_ap.h and remove include to cmpr.h, stub functions and below macro:
 - Make sure fbl_cfg (through user config) includes:


```
#if defined( FBL_ENABLE_COMPRESSION_MODE )
# define DATA_TYPE_COMPRESSED_USER 0x403ul
# define COMPRESSION_TYPE_USER (vuint8)0u
#endif
```
 - Make sure fbl_ap.c includes:


```
cmprLst[] = {
...
# if defined( DATA_TYPE_COMPRESSED_USER )
{
    V_NULL, V_NULL, V_NULL, V_NULL
}
# endif
}
```
 - If compression module is delivered, add cmpr.c again to the project
 - Change Project State to “Production (default)”
 - Keep appNBID/keyNBID Dummy init blocks, e.g. by defining and asking for FBL_FORCE_NV_ELEMENTS_FOR_TEST additionally:


```
#if ( FBL_PROJECT_STATE >= FBL_INTEGRATION ) || defined( FBL_FORCE_NV_ELEMENTS_FOR_TEST )
# if defined(FBL_TEST_KEY_NBID) || defined(FBL_TEST_APP_NBID)
...

```
- Disable “Application Task”
- Configure Diagnostic buffer to smallest possible value for compression:
 - 1188 byte in case no BDL is configured
 - 4095 byte else
- Use the Preconfig Template **MandatoryDeliveryPreconfig_Tsi_Set6.cfg**.
 - Configure partition 3 (where cal3 is located) as protected area
 - */* Partition 3 as protected area */*

```
#define FBL_BOOTINFO_NUM_PROTECTED_CALIBRATIONS 1u
#define FBL_BOOTINFO_PROTECTEDCAL_PARTID_1 3u
```
 - Check FBL_BOOTINFO_BDL_TIMEOUT_IN_1MS_UNITS > 0 in case you delivery BDL option (should already be inherited from Demo config)
- If you store NBIDs (app-NBID, key-NBID) with some NVM component different to Basic NVM (FblNbid), add error directive to GENy user preconfig (template + demo), so that untested state is documented:

```
#if defined( FBL_HDR_ENABLE_BASIC_NVM_HANDLING )
# error “Basic Nvm feature is untested (different NVM used for testing).”
# error “Please test it if you want to use it (e.g. repeat test case 20.05.00).”
#endif
```

- If you use Basic NVM FBL_HDR_ENABLE_BASIC_NVM_HANDLING should be set by default
- If you use a Fee, be sure appNBID/keyNBID are set to zero when starting the tests and are not set to 0 on every reset (through defining FBL_TEST_KEY_NBID, FBL_TEST_APP_NBID)
- Other NVM solutions are unexpected, please talk with the program responsible

FblCan_14229_Gm->GM Module configuration:

- Enable “Gap Fill”
- Enable “Calibration module support”
- Disable “Multiple Application Modules support”

Generate the new configuration files, compile the FBL, and load the new DUT onto the target hardware.

Check IntegrationTests.xlsx for to be used Download containers.

Test No	Test Name	Test Entry	Expected result	Result
20.05.00	NBID NVM handling	<p>The following Application download containers are required (create b) and c) by modifying existing a)):</p> <p>a) “App00” with App-NBID 0 (default)</p> <p>b) “App15” with App-NBID 15</p> <p>c) “App16” with App-NBID 16</p> <p><i>You can use the prepared script in ._TsiCommon for creatin test containers.</i></p>	<p>Please perform the following download steps and verify the result:</p> <ol style="list-style-type: none">1. Download App00 -> successful2. Download App15 -> successful3. Download App00 -> PEC Err_App-NBID4. Download App15 -> successful5. Download App16 -> successful6. Download App00 -> PEC Err_App-NBID7. Download App15 -> PEC Err_App-NBID8. Download App16 -> successful <p><i>If the NBID value is not cleared again, further tests with App00 provoke error on download. Clearing the NBID value might be useful.</i></p>	OK

24.00.00	PROTECTED CALIBRATION (BDL)	Create container in folder TsiCommon\Set6_BDL	<p>Download the following sequences and verify results as expected.</p> <p>Sequence1 Download files from ApplHdr_Std_2Part_3cal: 1. <i>DemoAppl_sign.gbf</i> 2. <i>Cal1/2/3_sign.gbf</i> Verify application is running.</p> <p>Sequence2.1 (in case of Bdl option) In case Download files from ApplHdr_DeltaBdl (update): 3. <i>Bdl_Delta_APP_2Part3cal_APP_2Part3cal_V2_WD_sign.gbf</i> 4. <i>Bdl_Delta_CAL3_2Part3cal_CAL3_2Part3cal_V2_cmpr_WD_sign.gbf</i> Verify application is not running (partition 2 is not valid).</p> <p>Sequence2.1 (in case no Bdl option) In case Download files from ApplHdr_Std_2Part_3cal: 5. <i>DemoAppl_sign.gbf</i> Verify application is not running (partition 2 is not valid).</p> <p>Sequence2.2 Download files of ApplHdr_Std_2Part_3cal: 6. <i>Cal1/2_sign.gbf</i> Application is running.</p> <p>Sequence3 (only in case of Bdl option) Download files of ApplHdr_DeltaBdl (revert update): 7. <i>Bdl_Delta_APP_2Part3cal_V2_APP_2Part3cal_WD_sign.gbf</i> 8. <i>Bdl_Delta_CAL3_2Part3cal_V2_CAL3_2Part3cal_cmpr_WD_sign.gbf</i> 9. <i>Cal1_sign.gbf</i> 10. <i>Cal2_sign.gbf</i> Application must run and memory shows old application and cal3 data.</p>	N.t.
----------	-------------------------------------	--	--	------

After performing the above tests, save the generated configuration files (all found in the GenData folder), the GENy configuration file (found in the CAnDb folder), and all generated data files.

3.1.7 Watchdog Timing Measurement

Configure the watchdog to be triggered every 1ms. This is the minimum accuracy supported by the bootloader. Toggle I/O **<TODO>** during AplFblWDTrigger().

Implement I/O inside the WD-trigger routine. Measure the accuracy of the watchdog timing with an oscilloscope and document the result here (e.g. by providing a screenshot of the performed measurement). Measurement is done during download of the DemoAppl project inclusive cal modules (DemoAppl\Appl\ApplHdr_3Part_each_1Cal).

<TODO>

Configuration WD Trigger time 1ms:

- Measurement of several downloads (CAN uncompressed, Ethernet compressed and uncompressed)
 - Trigger setup to show deviations of trigger between **<TODO>**ms and **<TODO>**ms
- =>no deviation found

Note that compressed download shows reduction in WD trigger accuracy. Within cmpr_lzma_cfg.h CMPR_WATCHDOG_CYCLE_COUNT setting is critical here for good WD accuracy, as compression is the bottleneck in this setup, lower values should increase the accuracy and higher values decrease it.

Comment: Deviation of **< <TODO>** μ s measured. This should allow for any common watchdog with a required trigger rate above **2ms**.

4 Compression Module

If your delivery contains a compression module(s) perform the below test cases for each compression delivered (ARLE/LZMA).

To prepare this

- Enable FblDrvCan_<HW> -> Data Processing -> Compression Mode
- Generate the new configuration files, compile the FBL, and load the new DUT onto the target hardware.

Generate the following containers (check batch file for below test numbers in _TsiCommon):

- Uncompressed application
- Compressed application
- Compressed application, corrupted (modify first data Byte->byte 10)

Hint: The length of the compressed data envelope is stored in address 0x02...0x07 of the container.

Test No	Test Name	Test Entry	Expected result	Result
20.02.00	Compression 1	Download uncompressed container	Download of container is successful	N.t.
20.03.00	Compression 2	Download compressed container	Download of container is successful	N.t.

5 Updater

If your delivery contains a Bootloader Updater perform the below test cases.

To prepare this

- Use the Preconfig Template [MandatoryDeliveryPreconfig_Tsi_Set1-2--6.cfg](#)
- Generate the new configuration files, compile the FBL, and load the new DUT onto the target hardware.

Generate the following containers

- DemoUpdaterFbl with “new” FBL which e.g. toggles a LED to show the successful update.

Check “Test entry” and “Expected result” for each test in [SPECDOC21931](#)

Test No	Test Name	Test Entry	Expected result	Result
TCASE-418642	Invalidation Variant Presence Pattern	external	external	N.t.
TCASE-418099	Complete Update Turnaround Cycle	external	external	N.t.
TCASE-418100	Reset Safety I	external	external	N.t.
TCASE-418102	Without Debugger	external	external	N.t.

6 Tja1145 PN Transceiver integration

If your delivery contains a Tja1145 integration, do the below tests (in any configuration).

Test No	Test Name	Test Entry	Expected result	Result
20.40.00	Partial Networking – TJA1145 Normal Mode	Within the user callback function named ApplTrcvNormalMode() place the CAN transceiver in Normal Mode. Reset and run the flash bootloader. Perform a flash download over CAN.	No error frames on the CAN bus and the flash download is successful. TJA1145 INH pin remains high.	OK
20.40.01	Partial Networking – TJA1145 Sleep Mode	Within the user callback function named ApplTrcvSleepMode() place the CAN transceiver in Sleep Mode. Reset and run the flash bootloader.	After 60 seconds without any network traffic the CAN transceiver is placed into sleep mode. TJA1145 INH pin transitions low in sleep mode	OK
20.40.02	Partial Networking – TJA1145 WakeUp Event	While the CAN Transceiver is in Sleep Mode create a wakeup event i.e. CAN message	The TJA1145T INH pin will transition from low to high. If the INH pin is tied into a power supply the micro will turn on	OK

7 XCP

If your delivery contains XCP perform the below test cases.

To prepare this

- Load GENy Preconfig Preconfig_Tsi.cfg
- Generate the new configuration files, compile the FBL, and load the new DUT onto the target hardware.

Use CANape to download via XCP.

Test No	Test Name	Test Entry	Expected result	Result
23.00.00	Download via XCP	Download Application via XCP	Download is working, Application is running	N.a.
23.02.00	Upload via XCP	Upload Application via XCP Compare resulting hex-file with application	Upload succeeds	N.a.