

SCOPE OF APPLICATION All Project/Engineering	HYUNDAI AutoEver	SHT/SHTS 1 / 18
Responsibility: Classic AUTOSAR Team	AUTOSAR MemIf User Manual	DOC. NO
AUTOSAR MemIf User Manual		

Document Change History				
Date (YYYY-MM-DD)	Ver.	Editor	Chap	Content (before revision-> after revision)
2021-03-11	1.0.0.0	YJ.Yun	All	Initial version
2021-12-07	1.0.1.0	JH.Lim	4.3	Update module release note
2022-07-06	1.0.2.0	ThuanLD5	4.3	Update module release note
2022-09-30	1.0.3.0	ThuanLD5	4.3	Update module release note
2022-11-16	1.0.4.0	ThuanLD5	4.3	Update module release note
2022-11-24	1.0.4.1	ThuanLD5	4.3	Update module release note

Edition Date: 2022-07-06	File Name MemIf_UM.pdf	Creation 2022-07-06 Le Duc Thuan	Check 2022-07-06 SungHo Park	Approval 2022-07-06 JiHoon Jung
Document Management System				

Table of Contents

1. OVERVIEW	4
2. REFERENCE	4
3. AUTOSAR SYSTEM	5
3.1 Overview of Software Layers	5
3.2 AUTOSAR Memory Stack	6
4. PRODUCT RELEASE NOTES	8
4.1 Overview.....	8
4.2 Scope of the release	8
4.3 Module Release Note.....	8
4.3.1 Change log	8
4.4 Limitations	10
4.5 Deviation	10
5. CONFIGURATION GUIDE	11
5.1 MemIfGeneral Container.....	11
6. APPLICATION PROGRAMMING INTERFACE (API).....	12
6.1 Type Definitions	12
6.1.1 MemIf_StatusType.....	12
6.1.2 MemIf_JobResultType	12
6.1.3 MemIf_ModeType	12
6.2 Macro Constants.....	12
6.3 Functions	12
6.3.1 MemIf_SetMode	12
6.3.2 MemIf_Read	13
6.3.3 MemIf_Write	13
6.3.4 MemIf_Cancel.....	14
6.3.5 MemIf_GetStatus	14
6.3.6 MemIf_GetJobResult	14
6.3.7 MemIf_InvalidateBlock.....	15
6.3.8 MemIf_GetVersionInfo	15
6.3.9 MemIf_EraseImmediateBlock	15
6.4 Noted	16

7. GENERATOR.....	17
7.1 Generator Option	17
7.2 Generator Error Message	17
7.2.1 Error Messages	17
7.2.2 Warning Messages	17
7.2.3 Information Messages.....	17
8. APPENDIX	18
8.1 Bswmd (Bsw Module Description)	18
8.1.1 Bsw module version setting	18

1. Overview

It is written based on Autosar standard SRS / SWS and if you need more detailed functional description when using the module, refer to the reference document below

The interpretation of the category related to setting is as follows

- Changeable (C) : Items that can be set by the user
- Fixed (F) : Items that cannot be changed by user
- NotSupported (N) : Items that not supported

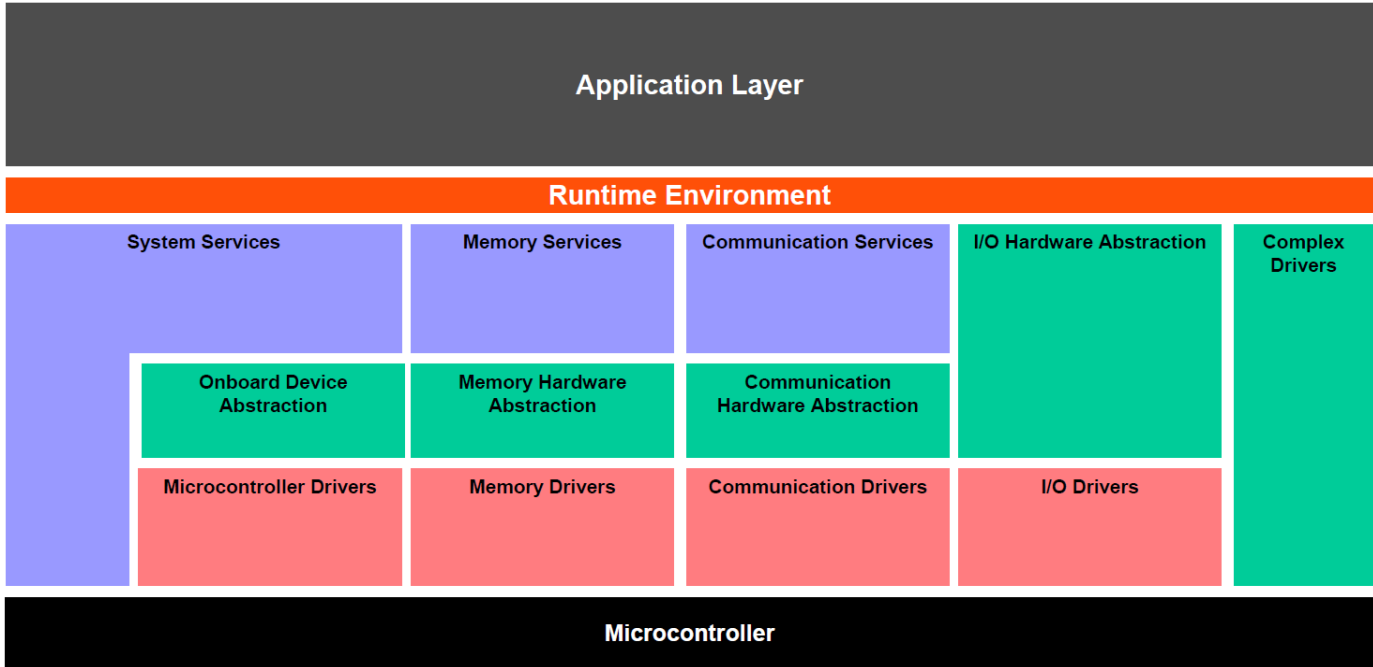
2. Reference

Sl. No.	Title	Version
1.	AUTOSAR_SWS_MemoryAbstractionInterface.pdf	4.4.0
2	AUTOSAR_SWS_NVRAMManager.pdf	4.4.0

3. AUTOSAR System

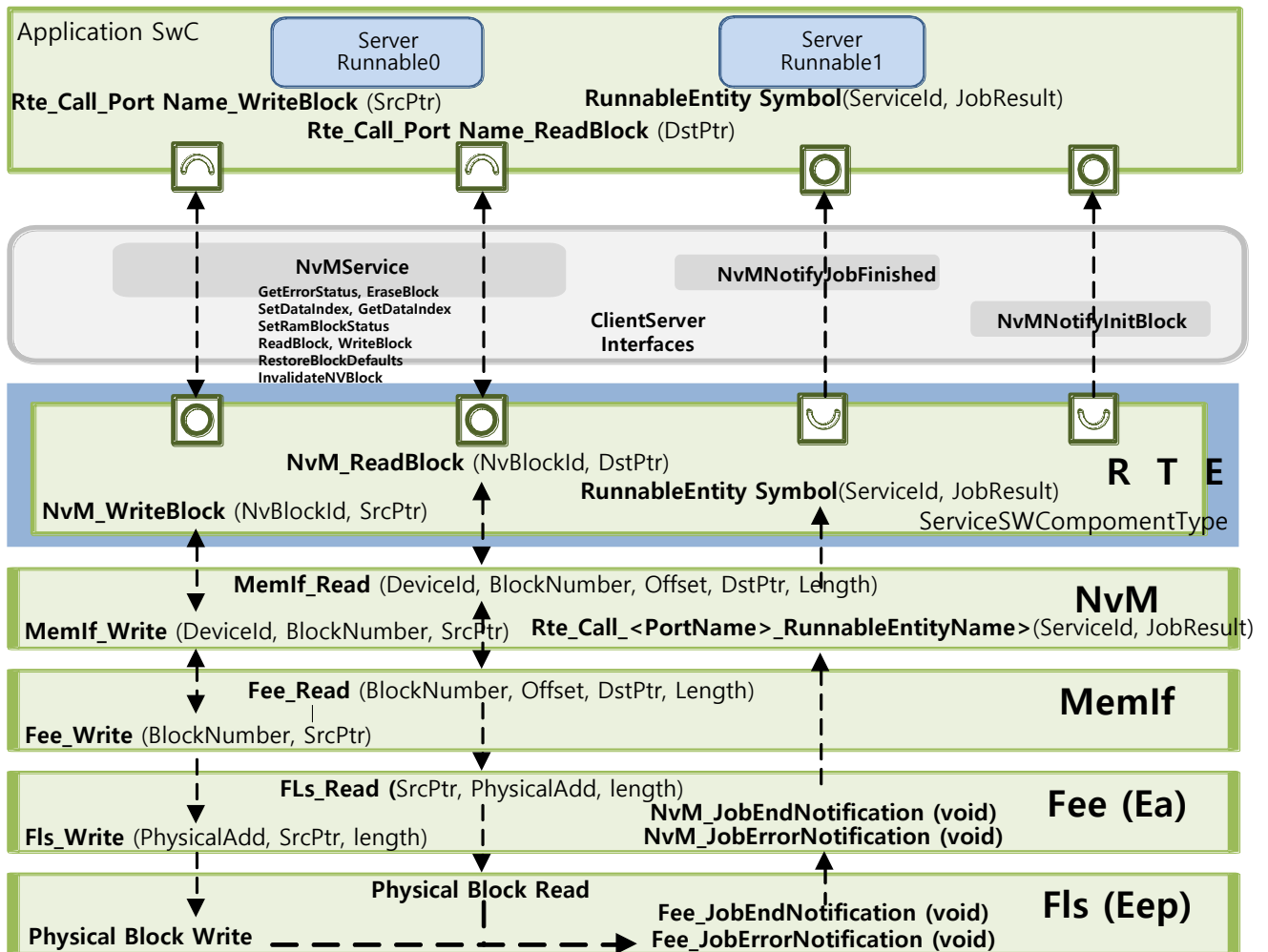
3.1 Overview of Software Layers

The Layered Architecture of the AUTOSAR platform is as follows. The AUTOSAR platform can be divided into Service Layer, ECU Abstraction Layer, Complex Device Drivers and Microcontroller Abstraction Layer.



3.2 AUTOSAR Memory Stack

Memory Stack module means NvM / MemIf / Ea / Eep / Fee / Fls module. Here, Fee and Fls modules are modules belonging to Mcal. For basic information, refer to the User Manual and Integration Manual of each MCU manufacturer. In order to use EEPROM, NvM and MemIf modules are basically required, and Fee / Fls module is used to use internal EEPROM, and Ea / Eep modules are additionally needed to use external EEPROM. The interface between Autosar layer and each module to use EEPROM is as follows.



Write requests (From SWS_NvM_00698)

Applications have to adhere to the following rules during write request for implicit synchronization between application and NVRAM manager:

- 1) The application fills a RAM block with the data that has to be written by the NvM module.
- 2) The application issues the NvM_WriteBlock or NvM_WritePRAMBlock request which transfers control to the NvM module.
- 3) From now on the application must not modify the RAM block until success or failure of the request is signaled or derived via polling. In the meantime the contents of the RAM block may be read.
- 4) An application can use polling to get the status of the request or can be informed via a callback function asynchronously.
- 5) After completion of the NvM module operation, the RAM block is reusable for modifications.

Read requests (From SWS_NvM_00699)

Applications have to adhere to the following rules during read request for implicit synchronization between application and NVRAM manager:

- 1) The application provides a RAM block that has to be filled with NVRAM data from the NvM module's side.
- 2) The application issues the NvM_ReadBlock request which transfers control to the NvM module.
- 3) From now on the application must not read or write to the RAM block until success or failure of the request is signaled or derived via polling.
- 4) An application can use polling to get the status of the request or can be informed via a callback function.
- 5) After completion of the NvM module operation, the RAM block is available with new data for use by the application.

4. Product Release Notes

4.1 Overview

The purpose of this chapter is to provide release-related content for HYUNDAI AUTOEVER MemIf Product and describes restrictions and specifics for MemIf Software product release version.

4.2 Scope of the release

All contents of this document are limited to the following HYUNDAI AUTOEVER MemIf modules.

Module	Autosar version	Module version
MemIf	4.4.0	1.0.4

※ Module version means the Sw version of each module's BswModule Description (Bswmd) file.

4.3 Module Release Note

4.3.1 Change log

➤ Version 1.0.4.1

- Improvement
 - Final Inspection result

Cause	Document and items need to be changed to meet inspection result
ASW Impact	N/A
Configuration Impact	N/A
ASW Action	N/A

➤ Version 1.0.4.0

- Improvement
 - Change defined Device ID value

Cause	Value of Device ID need to be changed to compatible with NvM_R44-1.2.0.0
ASW Impact	N/A
Configuration Impact	N/A
ASW Action	N/A

- Improvement
 - Configuration Audit, Quality gate inspection result

Cause	Document and items need to be changed to meet inspection result
ASW Impact	N/A
Configuration Impact	N/A
ASW Action	N/A

➤ Version 1.0.3.0

- Improvement

- Fix UNECE security coding rule violations

Cause	Code need to be changed to meet UNECE security coding rules
ASW Impact	N/A
Configuration Impact	N/A
ASW Action	N/A

➤ Version 1.0.2.0

- Improvement

- Module release artifacts improvements

Cause	Artifacts of module need to be changed and improved to meet Autoever requirements about copyright contents and folder structure
ASW Impact	N/A
Configuration Impact	N/A
ASW Action	N/A

- Improvement

- Change support MCAL version information

Cause	MCAL version information need to be changed for supporting R44 MCAL
ASW Impact	N/A
Configuration Impact	N/A
ASW Action	N/A

➤ Version 1.0.1.0

- Improvement

- Solving MemIf file inclusion error when applying MCAL-2.0.0

Cause	Error occurs during compilation after generation The problem is MemIf.h is referencing Fee.h then Fee.h file internally refer to Fls_17_Dmu_Cfg.h again so that some typedefs can not be referenced
ASW Impact	N/A
Configuration Impact	N/A
ASW Action	N/A

- Improvement

- Solving MemIf expected wrong AUTOSAR version of Fee

Cause	MemIf module can not compile successfully. MemIf expected Fee version does not match with Fee MCAL AUTOSAR version.
ASW Impact	N/A
Configuration Impact	N/A
ASW Action	N/A

- Feature

● Solving the generator version error

Cause	Generator has mismatch the version information. Need to update version policy in the generator.
ASW Impact	N/A
Configuration Impact	N/A
ASW Action	N/A

➤ Version 1.0.0.0

- 신규 기능

● Initial Version

원인	Initial Version
동작 영향	없음
설정 영향	없음
ASW 조치 사항	없음

4.4 Limitations

None

4.5 Deviation

None

5. Configuration Guide

5.1 MemIfGeneral Container

Parameter Name	Value	Category
DevErrorDetect	True	F
NumberOfDevice ¹⁾	User Defined (From SRS) (*)	F
VersionInfoApi	False	F

(*) Set to 1 when using only EA module or FEE module.

Set to 2 when using EA module and FEE module at the same time.

6. Application Programming Interface (API)

6.1 Type Definitions

6.1.1 MemIf_StatusType

Type:	Enumeration		
Range	MEMIF_UNINIT	0x00	The underlying abstraction module or device driver has not been initialized
	MEMIF_IDLE	0x01	The underlying abstraction module or device driver is currently idle.
	MEMIF_BUSY	0x02	The underlying abstraction module or device driver is currently busy.
	MEMIF_BUSY_INTERNAL	0x03	The underlying abstraction module is busy with internal management operations. The underlying device driver can be busy or idle.
Description:	Denotes the current status of the underlying abstraction module and device drivers.		

6.1.2 MemIf_JobResultType

Type:	Enumeration		
Range	MEMIF_JOB_OK	0x00	The job has been finished successfully.
	MEMIF_JOB_FAILED	0x01	The job has not been finished successfully.
	MEMIF_JOB_PENDING	0x02	The job has not yet been finished.
	MEMIF_JOB_CANCELED	0x03	The job has been canceled.
	MEMIF_BLOCK_INCONSISTENT	0x04	The requested block is inconsistent, it may contain corrupted data.
	MEMIF_BLOCK_INVALID	0x05	The requested block has been marked as invalid, the requested operation can not be performed.
Description:	Denotes the result of the last job.		

6.1.3 MemIf_ModeType

Type:	Enumeration		
Range	MEMIF_MODE_SLOW	0x00	The underlying memory abstraction modules and drivers are working in slow mode.
	MEMIF_MODE_FAST	0x01	The underlying memory abstraction modules and drivers are working in fast mode.
Description:	Denotes the operation mode of the underlying abstraction modules and device drivers.		

6.2 Macro Constants

None

6.3 Functions

6.3.1 MemIf_SetMode

Service name:	MemIf_SetMode
Syntax:	void MemIf_SetMode(MemIf_ModeType Mode)

Service ID[hex]:	0x0	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	Mode	--
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	Invokes the "SetMode" functions of all underlying memory abstraction modules	

6.3.2 MemIf_Read

Service name:	MemIf_Read	
Syntax:	<pre>Std_ReturnType MemIf_Read (uint8 DeviceIndex, uint16 BlockNumber, uint16 BlockOffset, uint8* DataBufferPtr, uint16 Length)</pre>	
Service ID[hex]:	0x02	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	DeviceIndex	--
	BlockNumber	--
	BlockOffset	--
	Length	--
Parameters (inout):	None	
Parameters (out):	DataBufferPtr	--
Return value:	Std_ReturnType	In case development error detection is enabled for the Memory Abstraction Interface and a development error is detected the function shall return E_NOT_OK else it shall return the value of the called function of the underlying module.
Description:	Invokes the "Read" functions of all underlying memory abstraction modules selected by the parameter DeviceIndex	

6.3.3 MemIf_Write

Service name:	MemIf_Write	
Syntax:	<pre>Std_ReturnType MemIf_Write (uint8 DeviceIndex, uint16 BlockNumber, const uint8* DataBufferPtr)</pre>	
Service ID[hex]:	0x03	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	DeviceIndex	--
	BlockNumber	--
	DataBufferPtr	--
Parameters (inout):	None	
Parameters (out):	None	
Return value:	Std_ReturnType	In case development error detection is enabled for the Memory Abstraction Interface and a development error is detected the

		function shall return E_NOT_OK else it shall return the value of the called function of the underlying module.
Description:	Invokes the "Write" functions of all underlying memory abstraction modules selected by the parameter DeviceIndex	

6.3.4 MemIf_Cancel

Service name:	MemIf_Cancel	
Syntax:	void MemIf_Cancel (uint8 DeviceIndex)	
Service ID[hex]:	0x04	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	DeviceIndex	--
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None.	
Description:	Invokes the "Cancel" functions of all underlying memory abstraction modules selected by the parameter DeviceIndex	

6.3.5 MemIf_GetStatus

Service name:	MemIf_GetStatus	
Syntax:	MemIf_StatusType MemIf_GetStatus (uint8 DeviceIndex)	
Service ID[hex]:	0x05	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	DeviceIndex	--
Parameters (inout):	None	
Parameters (out):	None	
Return value:	MemIf_StatusType	--
Description:	Invokes the "GetStatus" functions of all underlying memory abstraction modules selected by the parameter DeviceIndex	

6.3.6 MemIf_GetJobResult

Service name:	MemIf_GetJobResult	
Syntax:	MemIf_JobResultType MemIf_GetJobResult (uint8 DeviceIndex)	
Service ID[hex]:	0x06	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	DeviceIndex	--
Parameters (inout):	None	
Parameters (out):	None	
Return value:	MemIf_JobResultType	In case development error detection is enabled for the Memory Abstraction Interface and a development error is detected according to SWS_MemIf_00022 the function shall return MEMIF_JOB_FAILED else it shall return the value of the called

	function of the underlying module.
Description:	Invokes the "GetJobResult" functions of all underlying memory abstraction modules selected by the parameter DeviceIndex

6.3.7 MemIf_InvalidateBlock

Service name:	MemIf_InvalidateBlock	
Syntax:	Std_ReturnType MemIf_InvalidateBlock (uint8 DeviceIndex, unit16 BlockNumber)	
Service ID[hex]:	0x07	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	DeviceIndex	--
	BlockNumber	--
Parameters (inout):	None	
Parameters (out):	None	
Return value:	Std_ReturnType	In case development error detection is enabled for the Memory Abstraction Interface and a development error is detected according to SWS_MemIf_00022 the function shall return E_NOT_OK else it shall return the value of the called function of the underlying module.
Description:	Invokes the "InvalidateBlock" functions of all underlying memory abstraction modules selected by the parameter DeviceIndex	

6.3.8 MemIf_GetVersionInfo

Service name:	MemIf_GetVersionInfo	
Syntax:	void MemIf_GetVersionInfo (Std_VersionInfoType* VersionInfoPtr)	
Service ID[hex]:	0x08	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	None	
Parameters (inout):	None	
Parameters (out):	VersionInfoPtr	Pointer to standard version information structure
Return value:	None.	
Description:	Returns version information	

6.3.9 MemIf_EraseImmediateBlock

Service name:	MemIf_EraseImmediateBlock	
Syntax:	Std_ReturnType MemIf_EraseImmediateBlock (uint8 DeviceIndex, unit16 BlockNumber)	
Service ID[hex]:	0x09	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	DeviceIndex	--
	BlockNumber	--
Parameters (inout):	None	
Parameters (out):	None	
Return value:	Std_ReturnType	In case development error detection is

		enabled for the Memory Abstraction Interface and a development error is detected the function shall return E_NOT_OK else it shall return the value of the called function of the underlying module
Description:	Invokes the " EraseImmediateBlock " functions of all underlying memory abstraction modules selected by the parameter DeviceIndex	

6.4 Noted

None

7. Generator

7.1 Generator Option

None

7.2 Generator Error Message

7.2.1 Error Messages

- 1) ERR022001: The value configured for parameter 'Module Id' in container 'BSW-MODULE-DESCRIPTION' in provided MDT file is not correct. Module ID of MemIf must be 22.
 - This error occurs, If value of ModuleId in file BSWMDT is not equals with the ModuleId of MemIf.
- 2) ERR022002: The parameter 'Parameter Name' in the container 'Container Name' should be configured.
 - This error occurs, if any of the mandatory configuration parameters mentioned below is not configured in ECU Configuration Description File.

Container Name	Parameter Name
BSW-IMPLEMENTATION	AR-RELEASE-VERSION
	VENDOR-ID
	SW-VERSION
BSW-MODULE-DESCRIPTION	MODULE-ID

- 3) ERR022003: The value configured for the parameter 'Parameter Name' in the container 'Container Name' is incorrect. It should be 1.0.0 for example.
 - This error occurs, if the parameter 'Parameter Name' is not configured as per the pattern.

Parameter Name	Container Name	Pattern	Example
SW-VERSION	BSW-IMPLEMENTATION	[0-9]+.[0-9]+.[0-9]	1.0.0

- 4) ERR022004: AUTOSAR Release version <Version> configured for the parameter 'AR-RELEASE-VERSION' in provided MDT file is not correct. AUTOSAR Release version should be one of the following: <Versions>.
 - This error occurs, if the value of the element AR-RELEASE-VERSION present in the Bsw Module Description template is configured other than 4.4.0.
- 5) ERR022005: The value of parameter ' MemIfNumberOfDevices' in container 'MemIfGeneral' is not match with the number of device inputs.
 - This error occurs, If value of MemIfNumberOfDevices is not match with the number of device inputs.

7.2.2 Warning Messages

None

7.2.3 Information Messages

None

8. Appendix

8.1 Bswmd (Bsw Module Description)

8.1.1 Bsw module version setting

When compiling each module, if version information does not match, an error is generated by Compile. At this time, the version information must be modified in the BswImplementation Container as follows of Bswmd.

The screenshot shows the 'BswImplementation_MemIf' container settings. The 'Path' is set to 'BswImplementation_MemIf [BswImplementation]'. The 'Navigator' on the left shows the container structure. The 'Container Details - Elements > Bsw Implementation' panel on the right contains the following fields:

Short Name*:	BswImplementation_MemIf	Edit
Programming Language:	C	
Sw Version:	0.0.1	
Vendor Id:	76	Convert...
Ar Release Version:	4.4.0	
Behavior:	BswInternalBehavior_MemIf [/Bsw_MemIf/MemIf/BswInternalBehavior_MemIf]	Browse...