

SCOPE OF APPLICATION All Project/Engineering	HYUNDAI AutoEver	SHT/SHTS 1 / 20
Responsibility: Classic AUTOSAR Team	AUTOSAR CRC User Manual	DOC. NO
<h1>AUTOSAR CRC User Manual</h1>		

Document Change History				
Date (YYYY-MM-DD)	Ver.	Editor	Chap	Description
2021-03-10	1.0.0.0	YJ.Yun	All	Initial version
2021-07-21	1.0.1.0	JH.Lim	4.3	Update Module release notes
2021-12-31	1.0.2.0	HiepVT1	4.3	* Applying change of company name * Update Change log
2022-04-15	1.0.2.1	SH.Park	4.3	Update Change log
2022-07-06	1.0.2.2	ThuanLD5	4.3	Update Change log
2022-08-12	1.0.3.0	ThuanLD5	4.3	Update Change log
2022-11-18	1.0.3.1	ThuanLD5	4.3	Update Change log

Edition Date: 2022-11-18	File Name Crc_UM.pdf	Creation 2022-11-18 Le Duc Thuan	Check 2022-11-18 SungHo Park	Approval 2022-11-18 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 CRC Library	5
3.2.1 Sequence Diagrams	5
3.2.2 Architecture (Library)	7
4. PRODUCT RELEASE NOTES	8
4.1 Overview	8
4.2 Scope of the release	8
4.3 Module release notes	8
4.3.1 Change Log	8
4.4 Limitations	10
4.5 Deviation	10
5. CONFIGURATION GUIDE	11
5.1 Crc module	11
5.1.1 CrcGeneral Container	11
6. APPLICATION PROGRAMMING INTERFACE (API)	12
6.1 Type Definitions	12
6.2 Macro Constants	12
6.3 Functions	12
6.3.1 Initialization/Deinitialization Related API	12
6.3.2 Calculation of 8 bit CRC	12
6.3.3 Calculation of 8 bit CRC with 0X2F polynomial	13
6.3.4 Calculation of 16 bit CCITT-FALSE CRC16	13
6.3.5 Calculation of 16 bit with 0x8005 polynomial	14
6.3.6 Calculation of 32 bit Ethernet CRC	14
6.3.7 Calculation of 32 bit 0xF4ACFB13 polynomial CRC	15
6.3.8 Calculation of 64 bit 0x42F0E1EBA9EA3693 polynomial CRC	16
6.3.9 GetVersion Information	16
7. GENERATOR	18
7.1 Generator Option	18
7.2 Generator Error Message	18
7.2.1 Crc	18
8. APPENDIX	19
8.1 Bswmd (Bsw Module Description)	19
8.1.1 Bsw module version setting	19
8.2 Exclusive Areas	19
8.3 Example	19
8.3.1 Calculation of CRC8	19

8.3.2	Calculation of CRC8H2F()	19
8.3.3	Calculation of CRC16()	19
8.3.4	Calculation of CRC16ARC()	19
8.3.5	Calculation of CRC32()	19
8.3.6	Calculation of CRC32P4()	20
8.3.7	Calculation of CRC64()	20

1. Overview

This document provides caution or reference when using AUTOSAR platform for CRC use, when setting parameters or designing system. Please refer to the Reference document for details.

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 the user
- Not Supported (N): Not used

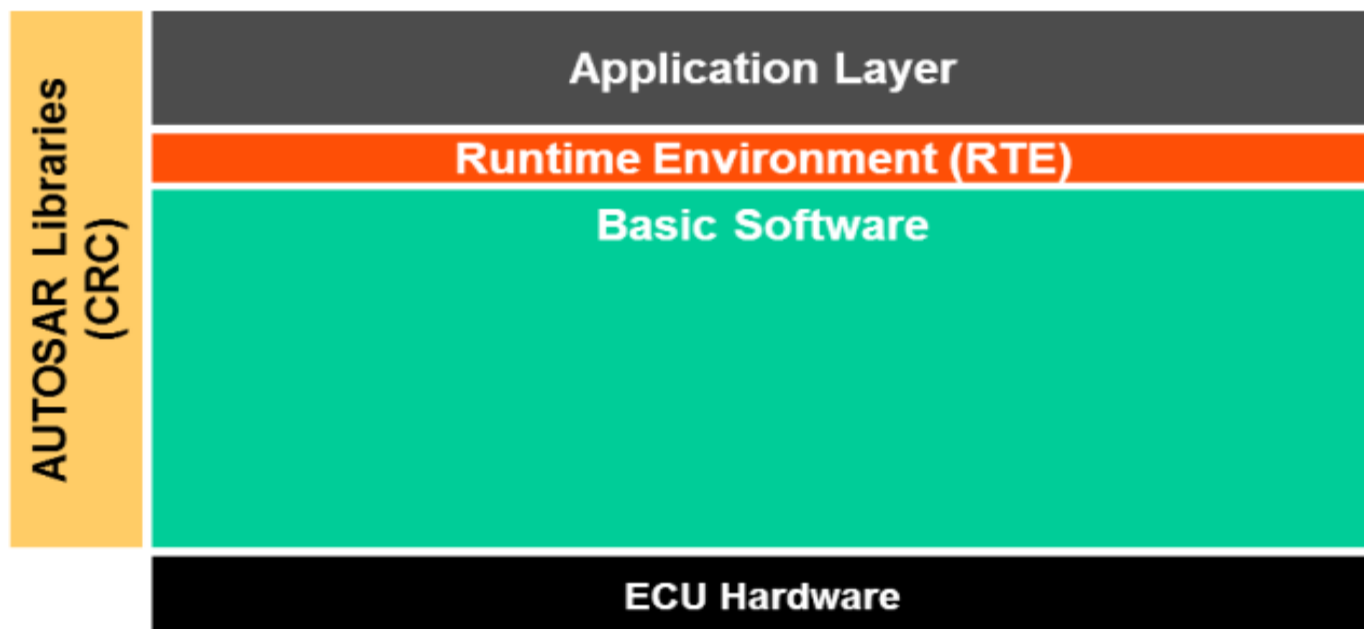
2. Reference

Sl. No.	Title	Version
1.	AUTOSAR_SWS_CRCLibrary.pdf	4.4.0

3. AUTOSAR System

3.1 Overview of Software Layers

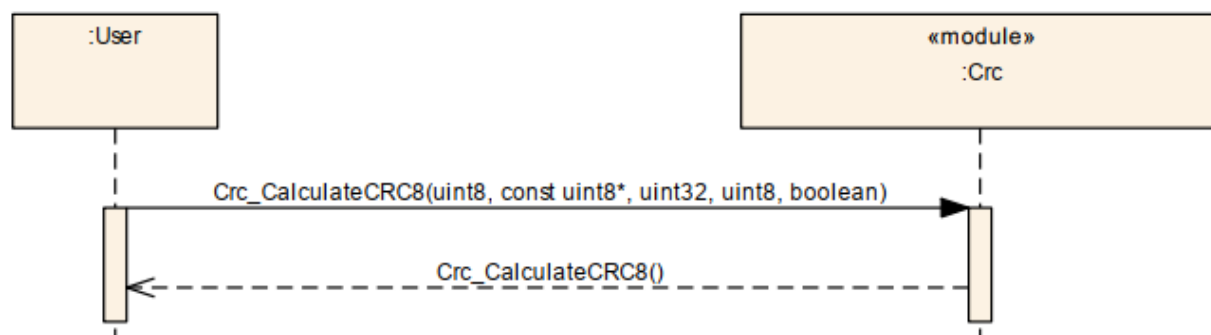
The CRC-related layered architecture of the AUTOSAR platform is as follows.



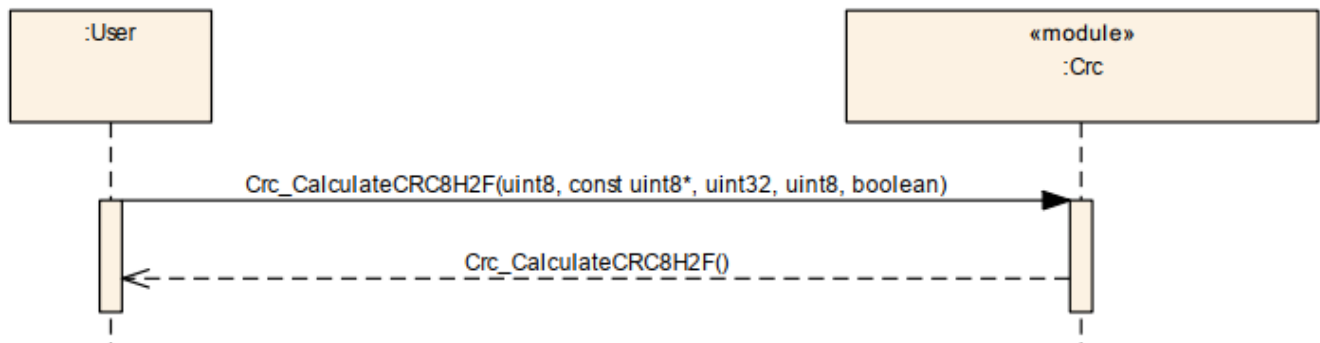
3.2 AUTOSAR CRC Library

3.2.1 Sequence Diagrams

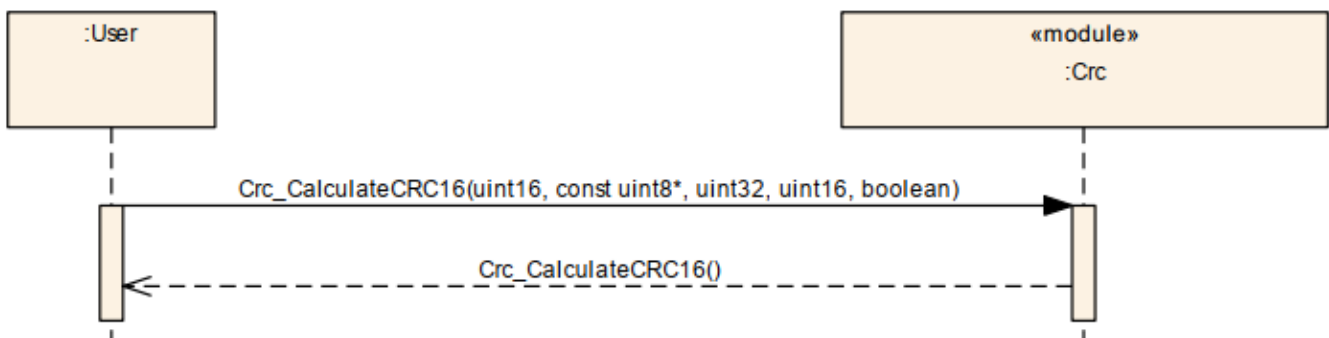
3.2.1.1 CRC_CalculateCRC8()



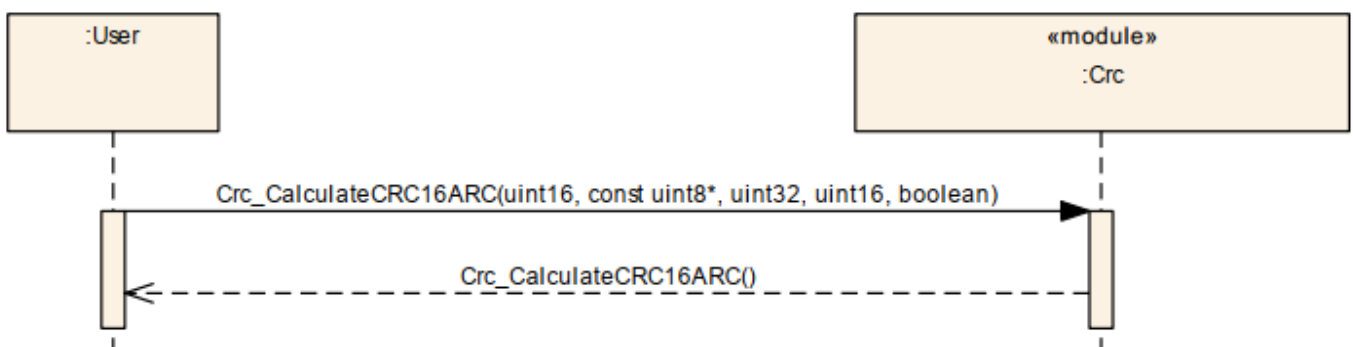
3.2.1.2 Crc_CalculateCRC8H2F()



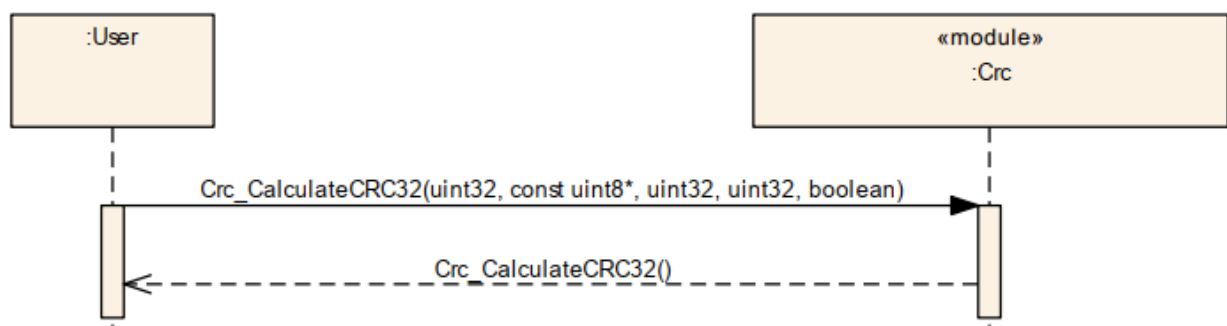
3.2.1.3 Crc_CalculateCRC16()



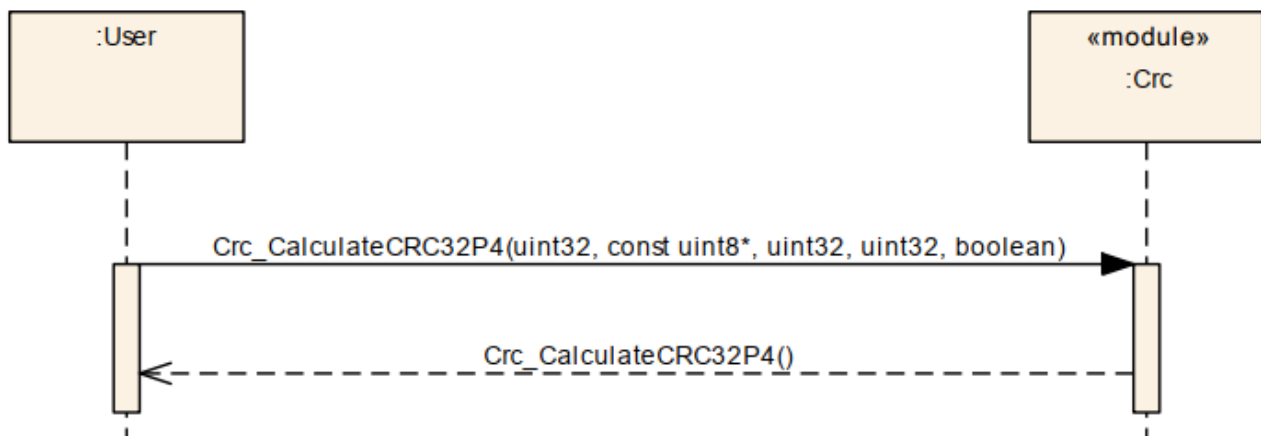
3.2.1.4 Crc_CalculateCRC16ARC()



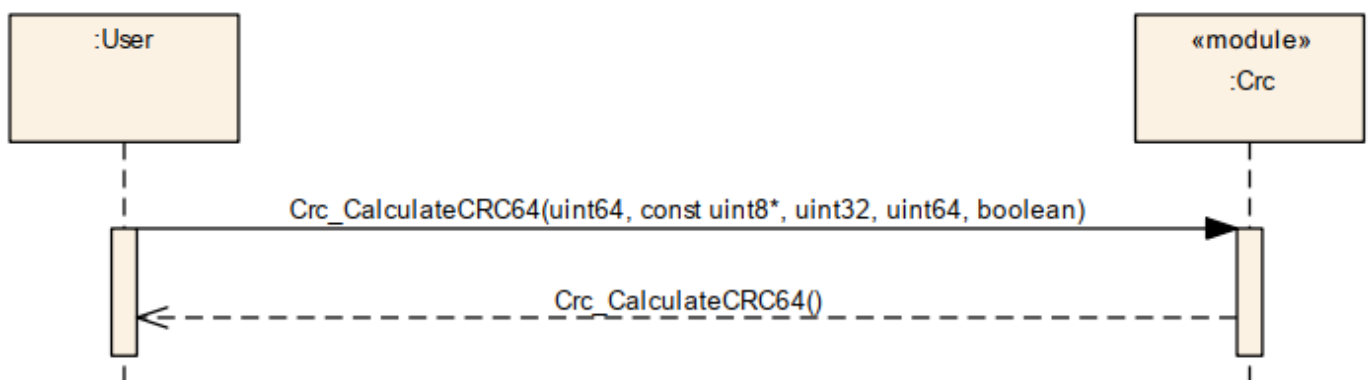
3.2.1.5 Crc_CalculateCRC32()



3.2.1.6 Crc_CalculateCRC32P4()



3.2.1.7 Crc_CalculateCRC64()



3.2.2 Architecture (Library)

3.2.2.1 Library can be called by BSW modules (that including the RTE), SW-Cs, libraries or integration code.

3.2.2.2 Library can be re-entrant.

3.2.2.3 Library does not require any initialization.

3.2.2.4 Library are synchronous (they do not have wait points)

4. Product Release Notes

4.1 Overview

In this chapter, the purpose is to provide release related contents of Hyundai AutoEver CRC Product, It describes limitations and specifics for the CRC product release version.

4.2 Scope of the release

All contents of this document are limited to the following Hyundai AutoEver CRC modules.

Module	Autosar version	Module version
Crc_R44	4.4.0	1.0.3

4.3 Module release notes

4.3.1 Change Log

➤ Version 1.0.3.1

- 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

- 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 1.0.3.0

- Improvement

■ Fix UNECE security coding rule violations

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

➤ Version 1.0.2.2

- 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

➤ Version 1.0.2.1

- Improvement

■ ASPICE Compliance update

Cause	Update documents follow ASPICE
ASW Impact	N/A
Configuration Impact	N/A
ASW Action	N/A

➤ Version 1.0.2.0

- Feature

■ Applying change of company name

- Improvement

■ Improved security coding to respond to UNECE Cyber Security regulations

Cause	Improved security coding to respond to UNECE Cyber Security regulations
ASW Impact	N/A
Configuration Impact	N/A
ASW Action	N/A

➤ Version 1.0.1.0

- 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

- Feature

■ Initial Version

Cause	Initial Version
ASW Impact	N/A
Configuration Impact	N/A
ASW Action	N/A

4.4 Limitations

Among the setting items, CRC_8_HARDWARE, CRC_8H2F_HARDWARE, CRC_16_HARDWARE, CRC_16_ARC_HARDWARE, CRC_32_HARDWARE, CRC_32P4_HARDWARE and CRC_64_HARDWARE must be supported by H / W, which is not supported in the current CRC module.

4.5 Deviation

None

5. Configuration Guide

5.1 Crc module

5.1.1 CrcGeneral Container

Refer to the following settings.

Parameter Name	Value	Category
Crc8Mode	CRC_8_RUNTIME	C
Crc8H2FMode	CRC_8H2F_RUNTIME	C
Crc16Mode	CRC_16_RUNTIME	C
Crc16ARCMODE	CRC_16_ARC_RUNTIME	C
Crc32Mode	CRC_32_RUNTIME	C
Crc32P4Mode	CRC_32P4_RUNTIME	C
Crc64Mode	CRC_64_RUNTIME	C

6. Application Programming Interface (API)

6.1 Type Definitions

None

6.2 Macro Constants

None

6.3 Functions

The user needs to ensure the input parameters valid when using CRC APIs.

- If the return value is 0, the user needs to consider there is an error in CRC API's Crc_DataPtr
- if Crc_Length is 0, the function will return in CRC_INITIAL_VALUE for the true in Crc_IsFirstCall, otherwise it will return the last calculated value.
- User should make sure the Crc_Length is NOT greater than the length of data block to be calculated.

6.3.1 Initialization/Deinitialization Related API

None

6.3.2 Calculation of 8 bit CRC

Function Name	Crc_CalculateCRC8	
Syntax	FUNC(uint8, CRC_CODE) Crc_CalculateCRC8 (P2CONST(uint8, CRC_CONST, CRC_APPL_CONST)Crc_DataPtr, uint32 Crc_Length, uint8 Crc_StartValue8, boolean Crc_IsFirstCall)	
Service ID	0x01	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	Crc_DataPtr	Pointer to the start address of the data block to be calculated.
	Crc_Length	Length of the data block to calculated in bytes.
	Crc_StartValue8	Initial value when algorithm starts.
	Crc_IsFirstCall	TRUE: First call in a sequence or individual CRC calculation; start from initial value, ignore Crc_StartValue8. FALSE: Subsequent call in a call sequence; Crc_StartValue8 is interpreted to be the return value of the previous function call.
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	uint8	8 bit result of CRC calculation.
Description	Calculates CRC (8 bit) using CRC-SAE-J1850. This service makes a CRC8	

	calculation on Crc_Length data bytes.
Preconditions	None
Configuration Dependency	This function shall be configurable with any one of the available CRC8 calculation methods by the configuration parameter : CRC_8_MODE

6.3.3 Calculation of 8 bit CRC with 0X2F polynomial

Function Name	Crc_CalculateCRC8H2F	
Syntax	FUNC(uint8, CRC_CODE) Crc_CalculateCRC8H2F (P2CONST(uint8, CRC_CONST, CRC_APPL_CONST)Crc_DataPtr, uint32 Crc_Length, uint8 Crc_StartValue8H2F, boolean Crc_IsFirstCall)	
Service ID	0x05	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	Crc_DataPtr	Pointer to the start address of the data block to be calculated.
	Crc_Length	Length of the data block to calculated in bytes.
	Crc_StartValue8H2F	Initial value when algorithm starts.
	Crc_IsFirstCall	TRUE: First call in a sequence or individual CRC calculation; start from initial value, ignore Crc_StartValue8H2F. FALSE: Subsequent call in a call sequence; Crc_StartValue8H2F is interpreted to be the return value of the previous function call.
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	uint8	8 bit result of CRC calculation.
Description	Calculates CRC (8 bit) using 0x2F polynomial. This service makes a CRC8 calculation with the Polynomial 0x2F on Crc_Length data bytes.	
Preconditions	None	
Configuration Dependency	This function shall be configurable with any one of the available CRC8H2F calculation methods by the configuration parameter : CRC_8H2F_MODE	

6.3.4 Calculation of 16 bit CCITT-FALSE CRC16

Function Name	Crc_CalculateCRC16	
Syntax	FUNC(uint16, CRC_CODE) Crc_CalculateCRC16 (P2CONST(uint8, CRC_CONST, CRC_APPL_CONST)Crc_DataPtr, uint32 Crc_Length, uint16 Crc_StartValue16, boolean Crc_IsFirstCall)	
Service ID	0x02	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	Crc_DataPtr	Pointer to the start address of the data block to be calculated.
	Crc_Length	Length of the data block to calculated in bytes.
	Crc_StartValue16	Initial value when algorithm starts.

	Crc_IsFirstCall	TRUE: First call in a sequence or individual CRC calculation; start from initial value, ignore Crc_StartValue16. FALSE: Subsequent call in a call sequence; Crc_StartValue16 is interpreted to be the return value of the previous function call.
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	uint16	16 bit result of CRC calculation.
Description	Calculates CRC (16 bit) using CRC-CCITT standards. This service makes a CRC16 calculation on Crc_Length data bytes.	
Preconditions	None	
Configuration Dependency	This function shall be configurable with any one of the available CRC16 calculation methods by the configuration parameter : CRC_16_MODE	

6.3.5 Calculation of 16 bit with 0x8005 polynomial

Function Name	Crc_CalculateCRC16ARC	
Syntax	FUNC(uint16, CRC_CODE) Crc_CalculateCRC16ARC (P2CONST(uint8, CRC_CONST, CRC_APPL_CONST)Crc_DataPtr, uint32 Crc_Length, uint16 Crc_StartValue16, boolean Crc_IsFirstCall);	
Service ID	0x08	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	Crc_DataPtr	Pointer to the start address of the data block to be calculated.
	Crc_Length	Length of the data block to be calculated in bytes.
	Crc_StartValue16	Initial value when algorithm starts.
	Crc_IsFirstCall	TRUE: First call in a sequence or individual CRC calculation; start from initial value, ignore Crc_StartValue16. FALSE: Subsequent call in a call sequence; Crc_StartValue16 is interpreted to be the return value of the previous function call.
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	uint16	16 bit result of CRC calculation.
Description	Calculates CRC (16 bit) using 0x8005 polynomial. This service makes a CRC16 calculation on Crc_Length data bytes.	
Preconditions	None	
Configuration Dependency	This function shall be configurable with any one of the available CRC16ARC calculation methods by the configuration parameter : CRC_16_ARC_MODE	

6.3.6 Calculation of 32 bit Ethernet CRC

Function Name	Crc_CalculateCRC32	
Syntax	FUNC(uint32, CRC_CODE) Crc_CalculateCRC32 (P2CONST(uint8, CRC_CONST, CRC_APPL_CONST)Crc_DataPtr, uint32 Crc_Length, uint32 Crc_StartValue32, boolean Crc_IsFirstCall)	
Service ID	0x03	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	Crc_DataPtr	Pointer to the start address of the data block to be calculated
	Crc_Length	Length of the data block to be calculated in bytes.
	Crc_StartValue32	Initial value when algorithm starts
	Crc_IsFirstCall	TRUE: First call in a sequence or individual CRC calculation; start from initial value, ignore Crc_StartValue32. FALSE: Subsequent call in a call sequence; Crc_StartValue32 is interpreted to be the return value of the previous function call.
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	uint32	32 bit result of CRC calculation.
Description	Calculates CRC (32 bit) using CRC-IEEE 802.3 standards by runtime based method. This service makes a CRC32 calculation on Crc_Length data bytes.	
Preconditions	None	
Configuration Dependency	This function shall be configurable with any one of the available CRC32 calculation methods by the configuration parameter : CRC_32_MODE	

6.3.7 Calculation of 32 bit 0xF4ACFB13 polynomial CRC

Function Name	Crc_CalculateCRC32P4	
Syntax	FUNC(uint32, CRC_CODE) Crc_CalculateCRC32P4 (P2CONST(uint8, CRC_CONST, CRC_APPL_CONST)Crc_DataPtr, uint32 Crc_Length, uint32 Crc_StartValue32, boolean Crc_IsFirstCall);	
Service ID	0x06	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	Crc_DataPtr	Pointer to the start address of the data block to be calculated
	Crc_Length	Length of the data block to be calculated in bytes.
	Crc_StartValue32	Initial value when algorithm starts
	Crc_IsFirstCall	TRUE: First call in a sequence or individual CRC calculation; start from initial value, ignore Crc_StartValue32. FALSE: Subsequent call in a call sequence; Crc_StartValue32 is interpreted to

		be the return value of the previous function call.
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	uint32	32 bit result of CRC calculation.
Description	Calculates CRC (32 bit) using 0xF4ACFB13 polynomial. This service makes a CRC32 calculation on Crc_Length data bytes.	
Preconditions	None	
Configuration Dependency	This function shall be configurable with any one of the available CRC32P4 calculation methods by the configuration parameter : CRC_32P4_MODE	

6.3.8 Calculation of 64 bit 0x42F0E1EBA9EA3693 polynomial CRC

Function Name	Crc_CalculateCRC64	
Syntax	FUNC(uint64, CRC_CODE) Crc_CalculateCRC64 (P2CONST(uint8, CRC_CONST, CRC_APPL_CONST)Crc_DataPtr, uint32 Crc_Length, uint64 Crc_StartValue64, boolean Crc_IsFirstCall)	
Service ID	0x07	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	Crc_DataPtr	Pointer to the start address of the data block to be calculated
	Crc_Length	Length of the data block to calculated in bytes.
	Crc_StartValue64	Initial value when algorithm starts
	Crc_IsFirstCall	TRUE: First call in a sequence or individual CRC calculation; start from initial value, ignore Crc_StartValue64. FALSE: Subsequent call in a call sequence; Crc_StartValue64 is interpreted to be the return value of the previous function call.
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Uint64	64 bit result of CRC calculation.
Description	Calculates CRC (64 bit) using 0x42F0E1EBA9EA3693 polynomial. This service makes a CRC64 calculation on Crc_Length data bytes.	
Preconditions	None	
Configuration Dependency	This function shall be configurable with any one of the available CRC64 calculation methods by the configuration parameter : CRC_64_MODE	

6.3.9 GetVersion Information

Function Name	Crc_GetVersionInfo	
Syntax	FUNC(void, CRC_CODE) Crc_GetVersionInfo (P2VAR(Std_VersionInfoType, AUTOMATIC, CRC_APPL_DATA) Versioninfo)	
Service ID	0x04	
Sync/Async	Synchronous	

Reentrancy	Reentrant	
Parameters (In)	None	
Parameters (Inout)	None	
Parameters (Out)	Versioninfo	Pointer represents where the version information of this module needs to be stored
Return Value	None	
Description	This function returns the version information of the module, which includes VendorId, ModuleId and Vendor Specific Version numbers. If Versioninfo is a NULL pointer then this service will simply return.	
Preconditions	None	
Configuration Dependency	None	

7. Generator

7.1 Generator Option

None

7.2 Generator Error Message

7.2.1 Crc

7.2.1.1 Error Messages

- 1) ERR2010001: Value of parameter VENDOR-ID of container BSW-IMPLEMENTATION is incorrect or not configured.
 - This error occurs, if value of parameter VENDOR-ID of container BSW-IMPLEMENTATION is not equal 76 or not.
- 2) ERR2010002: Value of parameter MODULE-ID of container BSW-MODULE-DESCRIPTION is incorrect or not configured.
 - This error occurs, if value of parameter MODULE-ID of container BSW-MODULE-DESCRIPTION is equal 201 or not.
- 3) ERR2010003: Value of SHORT-NAME parameter of container BSW-MODULE-DESCRIPTION is not configured.
 - This error occurs, if value of parameter SHORT-NAME of container BSW-MODULE-DESCRIPTION is configured or not.
- 4) ERR2010004: Value of parameter AR-RELEASE-VERSION of container BSW-IMPLEMENTATION should be configured.
 - This error occurs, if value of parameter AR-RELEASE-VERSION of container BSW-IMPLEMENTATION is configured or not.
- 5) ERR2010005: Value of parameter SW-VERSION of container BSW-IMPLEMENTATION should be configured.
 - This error occurs, if value of parameter SW-VERSION of container BSW-IMPLEMENTATION is configured or not.
- 6) ERR2010006: Value of parameter {0} of container {1} should follow the Pattern.
 - This error occurs, if value of parameter <Parameter Name> of container <Container Name> is follow the Pattern or not.

Parameter Name	Container Name	Pattern	Example
AR-RELEASE-VERSION	BSW-IMPLEMENTATION	4.[0-9]+.[0-9]+	4.4.0
SW-VERSION		1.[0-9]+.[0-9]+	1.0.0

7.2.1.2 Warning Messages

None

7.2.1.3 Information Messages

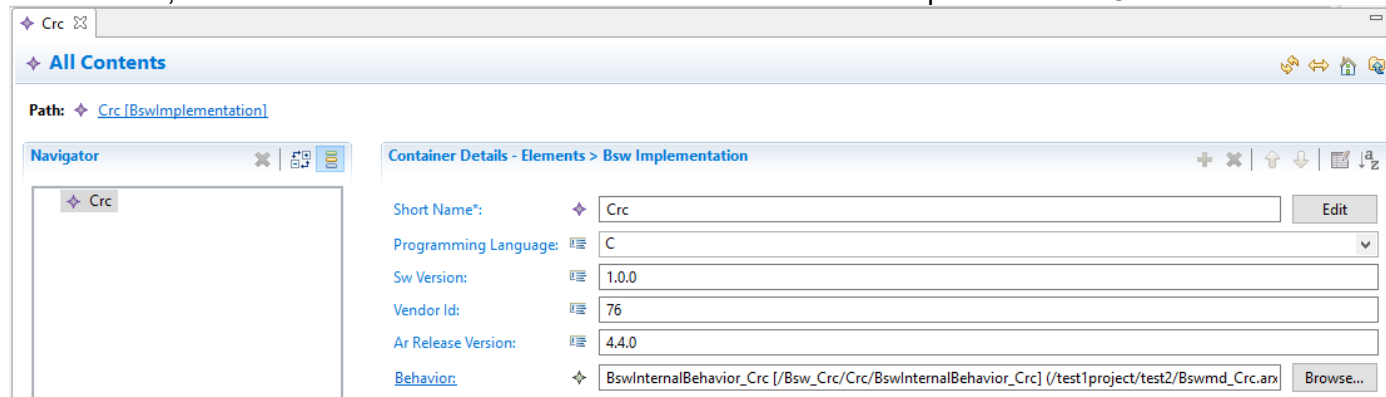
None

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. In this case, the version information must be modified in the BswImplementation Container as follows.



8.2 Exclusive Areas

None

8.3 Example

8.3.1 Calculation of CRC8

Data bytes 01h 02h 03h 04h 05h 06h 07h 08h: start value FFh:
Result = Crc_CalculateCRC8(&Array12345678[0], 8, 0xFF, TRUE);

8.3.2 Calculation of CRC8H2F()

Data bytes 01h 02h 03h 04h 05h 06h 07h 08h: start value FFh:
Result = Crc_CalculateCRC8H2F(&Array12345678[0], 8, 0xFF, TRUE);

8.3.3 Calculation of CRC16()

Data bytes 01h 02h 03h 04h 05h 06h 07h 08h: start value FFFFh:
Result = Crc_CalculateCRC16(&Array12345678[0], 8, 0xFFFF, TRUE);

8.3.4 Calculation of CRC16ARC()

Data bytes 01h 02h 03h 04h 05h 06h 07h 08h: start value FFFFh:
Result = Crc_CalculateCRC16ARC(&Array12345678[0], 8, 0xFFFF, TRUE);

8.3.5 Calculation of CRC32()

Data bytes 01h 02h 03h 04h 05h 06h 07h 08h: start value FFFFFFFFh:
Result = Crc_CalculateCRC32(&Array12345678[0], 8, 0xFFFFFFFF, TRUE);

8.3.6 Calculation of CRC32P4()

Data bytes 01h 02h 03h 04h 05h 06h 07h 08h: start value FFFFFFFFh:
Result = Crc_CalculateCRC32P4(&Array12345678[0], 8, 0xFFFFFFFF, TRUE);

8.3.7 Calculation of CRC64()

Data bytes 01h 02h 03h 04h 05h 06h 07h 08h: start value FFFFFFFFFFFFFFFFh:
Result = Crc_CalculateCRC64(&Array12345678[0], 8, 0xFFFFFFFFFFFFFFFF, TRUE);