AUTOSAR CanCM User Manual

Document Change History				
Date (YYYY-MM-DD)	Ver.	Editor	Chap	Content
2020-12-22	1.0.0.0	SeungJae Kim	All	Initial version
2021-10-18	1.0.1.0	LinhTT36	4.3	Add Change Log
2022-02-25	1.0.2.0	HieuTM8	4.2 4.3	Change scope of the Release Add Change Log
2022-04-28	1.0.3.0	HieuTM8	4.2 4.3	Update module version Add Change log
2022-06-29	1.0.3.1	NhatTD2	4.3	Add Change log
2022-08-02	1.0.4.0	NhatTD2	4.2 4.3	Update module version Add Change log
2022-10-14	1.0.5.0	NhatTD2	4.2 4.3	Update module version Add Change log
2022-11-06	1.0.6.0	Jiwon Oh	4.2 4.3 7.2	Update module version Add Change log Update Generator Error Message
2022-12-12	1.0.6.1	NhatTD2	4.2 4.3 7.2	Update module version Add Change log Update Generator Error Message
2023-07-03	1.0.7.0	Yubin Song	4.2 4.3	Update module version Add Change log
2023-09-22	1.0.8.0	Yubin Song	4.2 4.3	Update module version Add Change log
2024-02-02	1.0.9.0	HieuTM8	4.2 4.3	Update module version Add Change log

Table of Contents

1 Overview

The CanCM module was developed in correspondence with the HKMC communication design specification, and if more detailed functional description is needed 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 the user.
- Not Supported (N): Deprecated item

2 Reference

Sl. No.	Title	Version
1	ES95480-00 (High Speed CAN Design Specification)	-
2	ES95400-00 (CAN DESIGN SPECIFICATION)	-

3 AUTOSAR System

3.1 Overview of Software Layers

In the Hyundai Autoever AUTOSAR platform, the CAN Communication Stack consists of the following detailed modules.

- CanIf: Responsible for sending and receiving CAN messages
- PduR: PDU delivery between communication modules
- IpduM: In charge of sending and receiving Multiplexed PDUs
- CanTp: In charge of transmitting and receiving large amounts of data based on the Transport Protocol
- CanSM: In charge of CAN communication channel status control and bus-off processing
- CanTrcv: CAN transceiver hardware control
- OsekNm: Responsible for synchronization of CAN communication channel's SLEEP entry
- CanCM: Responsible for activating and deactivating CAN communication based on battery voltage and HKMC specifications

3.2 CanCM Module

The CanCM module performs the following operations for CAN communication control according to conditions.

- CAN Enable and disable CAN communication according to timing setting
- CAN Enable and disable CAN communication according to battery voltage conditions
- Com Activating and deactivating Com communication by switching ComM communication mode

Additional, CanCM also support checking Wake-up channels, is handled by CanCM

4 Product Release Notes

4.1 Overview

This chapter aims to provide the release information for the Huyndai Autoever CanCM module. Describes the limitations and specifics about the software product release version

4.2 Scope of the Release

All information in this document is limited to the following the Huyndai Autoever CanCM module.

Module name	AUTOSAR version	Module version
CanCM	-	1.0.9

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

4.3 Change Log

4.3.1. Version 1.0.9.0

- Improvement
  - Update to support R40 platform

Cause	To integrate CanCM_R44 module to R40 platform
Operation effect	None
Setting effect	1. Update Scons (Generation/CanCM) - Set *AUTOSAR44* to "true" - Update *InputFilesList* for CanCM_R44 2. Add *CanCM.bat* file in Build/bat_package for generator 3. Update Rte configuration - Add *RteBswExclusiveAreaImpl* Container in *RteBswModuleInstance* of CanCM - *RteExclusiveAreaImplMechanism* is ALL_INTERRUPT - *RteBswExclusiveAreaRef* refer to MAIN_PROTECTION in Bswmd of CanCM
ASW Action	None

- Improvement
  - Update to improve memory

Cause	Update Tun_CanCMFlagStatus type to improve memory
Operation effect	None
Setting effect	None
ASW Action	None

- Improvement
  - Update to fix Race Condition

Cause	Update E-Code to fix Race Condition violations
Operation effect	None

Setting effect	None
ASW Action	None

4.3.2. Version 1.0.8.0

- Improvement
  - Remove Gentool Version dependency

Cause	RTU Platform uses lower framework version
Operation effect	None
Setting effect	None
ASW Action	None

4.3.3. Version 1.0.7.0

- Improvement
  - Remove Dem Module dependencies

Cause	CanCM cannot be Compiled without “Dem” module
Operation effect	None
Setting effect	None
ASW Action	None

- Improvement
  - Fix A-SPICE Final Inspection Findings

Cause	All remaining A-SPICE final inspection findings should be fixed
Operation effect	None
Setting effect	None
ASW Action	None

4.3.4. Version 1.0.6.1

- Improvement
  - Fix the settings for each VARIANT to consider (ERR8810007, ERR8810008)

Cause	The configurations should be consider for each VARIANT (ERR8810007, ERR8810008)
Operation effect	Validation Rules ID (ERR8810007, ERR8810008) are checked by checking the Variants container size in new PostBuild process
Setting effect	None

ASW Action	None

- Task
  - Fix violations for ES95411-00 CodeProver and UNECE

Cause	Fix violations for ES95411-00 CodeProver and UNECE
Operation effect	None
Setting effect	None
ASW Action	None

4.3.5. Version 1.0.6.0

- Improvement
  - Fix Detach “Remote Wakeup”, “BAT Monitoring” functionality

Cause	CanCM can support the “Remote Wakeup” functionality only
Operation effect	CanCM is always the “Normal” Mode state when remove the “CanCMBatMonConfig” Container(Do not check the ADC for Mode)
Setting effect	None
ASW Action	None

- Improvement
  - Update T-code to match PostBuild requirements

Cause	All PB Module have to refer the Variant name from EcuC. And have to follow “<Mip>_ConfigType <Mip>_Config[_<PredefinedVariant.shortName>]” naming rule.
Operation effect	None
Setting effect	None
ASW Action	None

4.3.6. Version 1.0.5.0

- Task
  - Fix the Configuration Audit, Quality Gate Inspection

Cause	Fix the Configuration Audit, Quality Gate Inspection
Operation effect	None
Setting effect	None
ASW Action	None

- Improvement

- Fix the ASPICE Inspection results

Cause	Fix the ASPICE Inspection results
Operation effect	None
Setting effect	None
ASW Action	None

- Improvement
  - Improvement memmap section declaration

Cause	Improvement memmap section declaration
Operation effect	None
Setting effect	None
ASW Action	None

- Task
  - Improve the 100% Traceability in TM

Cause	Improve the 100% Traceability in TM
Operation effect	None
Setting effect	None
ASW Action	None

4.3.7. Version 1.0.4.0

- Improvement
  - Fix UNECE violations

Cause	Fix UNECE violations
Operation effect	None
Setting effect	None
ASW Action	None

4.3.8. Version 1.0.3.1

- Task
  - Editorial Changed of Work Products

Cause	Editorial Changed of Work Products
Operation effect	None
Setting effect	None

ASW Action	None

4.3.9. Version 1.0.3.0

- Change Request
  - Fix UNECE security coding rule violations

Cause	CanCM should follow UNECE security coding rule
Operation effect	None
Setting effect	None
ASW Action	None

4.3.10. Version 1.0.2.0

- Change Request
  - Update MEMMAP Section

Cause	Update memmap section code for all array
Operation effect	None
Setting effect	None
ASW Action	None

4.3.11. Version 1.0.1.0

- Change Request
  - Merge R44 changes

Cause	Migrate latest codes of CanSM R44 in the R44 Repository
Operation effect	None
Setting effect	None
ASW Action	None

- Change Request
  - Apply ASPICE compliance update

Cause	Fix findings to follow the ASPICE Process
Operation effect	None
Setting effect	None
ASW Action	None

4.3.12. Version 1.0.0.0

- New Feature

■ Initial Version

Cause	Initial Version
Operation effect	None
Setting effect	None
ASW Action	None

4.4 Limitations

Precautions when using ADC Read function of IoHwAb module directly in Asw

All Asw Tasks that call ADC read functions (IoHwAb_AnaInReadDirect, IoHwAb_AnaInDirReadDirect) must be tied to internal resources such as CanCM Tasks, or wrap the function parts through ExclusiveArea (SuspendAllInterrupt/ResumeAllInterrupt or SuspendOsInterrupt/ResumeOsInterrupt method) so that they are not wrapped between them. do.

4.5 Deviations

None

5 Configuration Guide

The CanCM setting of the AUTOSAR platform distributed by Hyundai Auto is a setting reflecting Hyundai Auto Policy's policy. Therefore, you should consult with Hyundai Auto.

The following chapters summarize all configuration parameters of module

5.1 CanCMGlobalConfig

Parameter Name	Value	Category
1) CanCMDemStatusReport	True	C
2) CanCMDisableDMOnAbnormalVoltage	From SRS	C
3) CanCMWakeupSupport	From SRS	C
4) CanCMHysteresisSupport	True	C
CanCMDevErrorDetect	True	C
CanCMMainFunctionPeriod	0.005	C
5) CanCMFilteringConstant	128	C

1) Whether to use the function to notify the DEM module when the battery voltage is within the critical range

2) Whether the Com module's DM (Deadline Monitoring) disable function is used when the battery voltage is within the Abnormal range

3) Whether to use the function to check the remote wake-up periodically in the NO COMM state

4) When determining Non-Critical voltage return from Critical voltage or Normal voltage return from Abnormal,

5) Coefficient used for filtering the input Adc value when monitoring the battery voltage. (See IoHwAb manual 8.1.3)

5.2 CanCMBatMonConfig

Parameter Name	Value	Category
1) CanCMBatAnalogInputRef	-	C
2) CanCMAdcDefault	500	C

1) ADC port designation for battery voltage monitoring:
   - CanCM module periodically sets the ADC value of the specified port.
   - Read and check the voltage status to control the communication function.
   - If two or more ports are specified, the ADC values of all the designated ports are read and the largest value is used.
2) Specify the initial value of the battery voltage. (See IoHwAb manual 8.1.3)

5.3 CanCMChannelConfig

Parameter Name	Value	Category
CanCMChannelId	**Automated**	**C**
[1)]CanCMNetworkActivationTime	**0.1**	**C**
[2)]CanCMVoltageErrorDelayTime	**0.12**	**C**
[3)]CanCMTimeoutMonitoringStartTime	**1**	**C**
[4)]CanCMVoltageAbnormalUpper	**-**	**C**
[4)]CanCMVoltageAbnormalLower	**-**	**C**
[5)]CanCMVoltageCriticalUpper	**-**	**C**
[5)]CanCMVoltageCriticalLower	**-**	**C**
[6)]CanCMVoltageHysteresis	**-**	**C**
CanCMComMChannelId	**Automated**	**C**

1) Time until the transmission function of the CAN message (Application message) is activated after the communication mode of the corresponding channel is switched to FULL COMMUNICATION.

   The corresponding time is "between FULL COMMUNICATION transition and First Application Message", which is different from the time "between Wakeup Message and First Application Message".
   As Wakeup message transmission takes time after FULL COMMUNICATION transition, the later time is shorten than the former one.
   So, If you configure tNwActive to "min" value of ES specification, the time "between Wakeup message and First Application Message" will take less time than "min".

2) Time until the voltage error state is released after the battery voltage returns from the critical range to the non-critical range (CAN message transmission function is activated immediately after release)

3) Wait time before activating the Rx Deadline Monitoring function of the Com module after the communication mode of the corresponding channel is changed to FULL COMMUNICATION.

4) Abnormal Voltage range designation

If the battery voltage is measured within the range, disable the Rx Deadline Monitoring function of the Com module to prevent DTC recording of the received signal timeout. When the battery voltage returns to the normal range, the function is re-enabled.
When the CanCMVoltageAbnormalUpper and CanCMVoltageAbnormalLower values are both 0, the above Rx Deadline Monitoring function control is not performed.

5) Critical Voltage range designation
If the battery voltage is measured within the range, transmission of CAN messages is prohibited. When the current state is changed to the normal voltage state, the CAN message is allowed to be transmitted again.
If the CanCMVoltageCriticalUpper and CanCMVoltageCriticalLower values are both 0, the above transmission function control is not performed.

6) Hysteresis value assignment
When returning from the critical range voltage to the non-critical range voltage, return by applying the hysteresis value
Judge whether or not.

-Critical Voltage judgment criteria
BatVol <= CanCMVoltageCriticalLower or BatVol >= CanCMVoltageCriticalUpper

-Non-Critical Voltage judgment criteria (Hysteresis is applied when returning from Critical)
CanCMVoltageCriticalLower + CanCMVoltageHysteresis <BatVol <CanCMVoltageCriticalUpper – CanCMVoltageHysteresis

-Abnormal Voltage judgment standard
BatVol <= CanCMVoltageAbnormalLower or BatVol >= CanCMVoltageAbnormalUpper

-Normal Voltage interplate standard (Hysteresis is applied when returning from Abnormal)
CanCMVoltageAbnormalLower + CanCMVoltageHysteresis <BatVol <CanCMVoltageAbnormalUpper-CanCMVoltageHysteresis

(BatVol: Voltage ADC value)

Since all communication control times are calculated by the periodic task of the CanCM module, errors may occur depending on the execution cycle of the task. The values of Can items 4) to 6) are based on the ADC input value of the port designated as the voltage monitoring input. The unit of time for the values of items 1) to 3) is 'seconds'. All setting values are set by referring to the parameters corresponding to the HKMC CAN ES design specifications.

5.4 CanCMDemEventParameterRefs

Parameter Name	Value	Category
1)CANCM_E_BAT_FAIL	-	C

1) Specify the event ID to notify the Dem module when the battery voltage is changed from the normal range to the critical range.

5.5 CanCMWakeupParameterRefs

Parameter Name	Value	Category

1)CanCMEcuMWakeupSourceRef	-	C
2)CanCMIoHwAbWakeupPinNameRef	-	C

1) Designate wake-up source defined in EcuM. When the CanCM module detects remote wake-up on the corresponding CAN channel, it calls the EcuM interface with the specified wake-up source as an argument.

2) Designate IO pin to input wake-up signal. The CanCM module detects remote wake-up by periodically reading the set pin when the ComM status is changed to NO COMM. Therefore, when the wake-up frame is detected on the bus in the Sleep or Stand-By state, the transceiver must maintain the voltage level of the set pin in the LOW state before switching to the Normal mode.

6 Application Programming Interface (API)

6.1 Type Definitions

None

6.2 Macro Constants

None

6.3 Functions

Describes all functionalities of CanCM module

6.3.1 Initialization

Function Name	CanCM_Init
Syntax	FUNC(void, CANCM_CODE) CanCM_Init(void)
Service ID [Hex]	0x01
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters (In)	None
Parameters (Inout)	None
Parameters (Out)	None
Return Value	None
Description	This service Initializes internal and external interfaces of the CanCM for the further processing.
Preconditions	None
Configuration Dependency	None
Available via	CanCM.h

Function Name	CanCM_GetVersionInfo
Syntax	FUNC(void, CANCM_CODE)CanCM_GetVersionInfo(P2VAR(Std_VersionInfoType, AUTOMATIC, CANCM_APPL_DATA) ptrVersionInfo)

Service ID [Hex]	0x07	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (In)	None	
Parameters (Inout)	None	
Parameters (Out)	ptrVersionInfo	Pointer to where to store the version information of this module.
Return Value	None	
Description	Get version the CanCM module.	
Preconditions	None	
Configuration Dependency	None	
Available via	CanCM.h	

6.3.2 Network State Monitoring

Function Name	CanCM_ComModeIndication	
Syntax	FUNC(void, CANCM_CODE) CanCM_ComModeIndication (NetworkHandleType channelIdComm, ComM_ModeType commState)	
Service ID [Hex]	0x03	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (In)	channelIdComm	ComM channel id
	commState	COMM_NO_COMMUNICATION COMM_SILENT_COMMUNICATION COMM_FULL_COMMUNICATION
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	None	
Description	Receive state of channels from ComM through this API	
Preconditions	None	
Configuration Dependency	None	
Available via	CanCM_Cbk.h	

Function Name	CanCM_GetCurrentNetworkState	
Syntax	FUNC(Std_ReturnType, CANCM_CODE) CanCM_GetCurrentNetworkState(NetworkHandleType channelIdCanCm, P2VAR(CanCM_CommStatType, AUTOMATIC, CANCM_APPL_DATA) ptrCommStat)	
Service ID [Hex]	0x05	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (In)	channelIdCanCm	CanCM channel id

Parameters (Inout)	None	
Parameters (Out)	ptrCommStat	CANCM_COMM_STAT_INACTIVE CANCM_COMM_STAT_START CANCM_COMM_STAT_ACTIVE CANCM_COMM_STAT_SHUTDOWN
Return Value	E_OK	
	E_NOT_OK	
Description	This API aims to get current state of channel	
Preconditions	None	
Configuration Dependency	None	
Available via	CanCM.h	

6.3.3 Battery Voltage Monitoring

Function Name	CanCM_GetCurrentVoltageMode	
Syntax	FUNC(Std_ReturnType , CANCM_CODE) CanCM_GetCurrentVoltageMode(CanCM_ChannelType channelIdCanCm, P2VAR(CanCM_VolModType, AUTOMATIC, CANCM_APPL_DATA) ptrVolMode)	
Service ID [Hex]	0x04	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (In)	channelIdCanCm	CanCM channel id
Parameters (Inout)	None	
Parameters (Out)	ptrVolMode	CANCM_VOL_MODE_NORMAL CANCM_VOL_MODE_ABNORMAL CANCM_VOL_MODE_CRITICAL
Return Value	None	
Description	This API aims to get voltage mode of channel	
Preconditions	None	
Configuration Dependency	None	
Available via	CanCM.h	

Function Name	CanCM_GetVoltageFailCount
Syntax	FUNC(void, CANCM_CODE) CanCM_GetVoltageFailCount(P2VAR(uint16, AUTOMATIC, CANCM_APPL_DATA) ptrCount)
Service ID [Hex]	0x06
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters (In)	None
Parameters (Inout)	None

Parameters (Out)	ptrCount	uint16
Return Value	None	
Description	This API aims to get number of fail of voltage (Critical)	
Preconditions	None	
Configuration Dependency	None	
Available via	CanCM.h	

6.3.4 Check Wake-up

Function Name	CanCM_CheckWakeup	
Syntax	FUNC(void, CANCM_CODE) CanCM_CheckWakeup(EcuM_WakeupSourceType source)	
Service ID [Hex]	0x08	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (In)	source	EcuM_WakeupSourceType
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	None	
Description	This API aims to check wake-up a source	
Preconditions	None	
Configuration Dependency	None	
Available via	CanCM.h	

7 Generator

7.1 Generator Message

Options	Description
-G,--Generation	Symbolic parameters to be used for fore generation (skip validation).
-H,--Help	Display this help message.
-I,--Input <I>	ECU description file path of the module for which generation tool need to run.
-L,--Log	Symbolic parameters to be used for generation error log.
-M,--Module <M>	Specify module name and version to be generated code for.
-O,--Output <O>	Project-relative path to location where the generated code is to be placed.
-T,--Top_path <T>	Symbolic parameters to be used for set path of module.
-V,--Validate	Symbolic parameters to be used for invoking validation checks.

7.2.1 Error Messages

This chapter describes all error message following validation rules of CanCM module

ERR8810007 CanCMChannelId <id> is repeated.

ERR8810008 There are more than one <CanCMComMChannelId> in /AUTRON/CanCM/CanCMChannelConfig.

ERR8810009 Value of the parameter 'CanCMVoltageCriticalLower' in addition with value of the parameter 'CanCMVoltageHysteresis' in the container '' of the channel <index> should be smaller than value of the parameter 'CanCMVoltageAbnormalLower' in the container.

ERR8810010 Value of the parameter 'CanCMVoltageCriticalUpper' in subtraction with value of the parameter 'CanCMVoltageHysteresis' in the container '' of the channel <index> should be bigger than value of the parameter 'CanCMVoltageAbnormalUpper' in the container

ERR8810011 Value of the parameter 'CanCMVoltageAbnormalLower' in the container '' of the channel <index> should be smaller than value of the parameter 'CanCMVoltageAbnormalUpper' in the container

ERR8810023 The CanCM module supported Post-Build but there are no variants configured in EcuC.

ERR8810025 Mismatch post-build variant with EcuC module, the module's post-build variants should include all EcuC post-build variants.

7.2.2 Warning Messages

None

7.2.2 Information Messages

None

8. SWP Error Code

8.1 SWP Error Code List

None

9 Appendix

None