MICROSAR vCanCcGbt

Technical Reference

Version 1.0.2

Authors	Simon Gutjahr
Status	Not Released

Document Information

History

Author	Date	Version	Remarks
Simon Gutjahr	[2017-07-10]	1.00.00	Initial version
Simon Gutjahr	[2017-08-03]	1.00.01	Updated API description
Simon Gutjahr	[2017-08-21]	1.00.02	Updated API description

Reference Documents

No.	Source	Title	Version
[1]	AUTOSAR	AUTOSAR_SWS_DET.pdf	4.0.3
[2]	AUTOSAR	AUTOSAR_BasicSoftwareModules.pdf	1.0.0
[3]		GBT27930-2015EN.pdf	2015
[4]	Vector	TechnicalReference_PostBuildLoadable.pdf	1.0.0

Caution
We have configured the programs in accordance with your specifications in the questionnaire. Whereas the programs do support other configurations than the one specified in your questionnaire, Vector´s release of the programs delivered to your company is expressly restricted to the configuration you have specified in the questionnaire.

Contents

Illustrations

Tables

1 Component History

The component history gives an overview over the important milestones that are supported in the different versions of the component.

Component Version	New Features
1.00.00	Initial component version

Table 1-1 Component history

2 Introduction

This document describes the functionality, API and configuration of the MICROSAR BSW module vCanCcGbt as specified in [3].

Supported AUTOSAR Release	4.x	
Supported Configuration Variants:	pre-compile, post-build-loadable, post-build-selectable	
Vendor ID:	vCanCcGbt_VENDOR_ID	30 decimal (= Vector-Informatik, according to HIS)
Module ID:	vCanCcGbt_MODULE_ID	255 decimal (according to Vector internal numbering)

In the document [3] a Communication protocol between Off-Board Conductive Charger (Charger) and Battery Management System (BMS) for Electric Vehicle is defined by the Standardization Administration of the Peoples Republic of China. The module vCanCcGbt implements the BMS side of the communication protocol needed for charging a battery of an electric vehicle.

In the MICROSAR architecture, the vCanCcGbt interacts with the PduR to transmit and receive the communication protocol messages and with an upper layer. The upper layer is notified of received protocol messages of the Charger and has to provide BMS data to the vCanCcGbt.

Architecture Overview

The following figure shows where the vCanCcGbt is located in the AUTOSAR architecture.

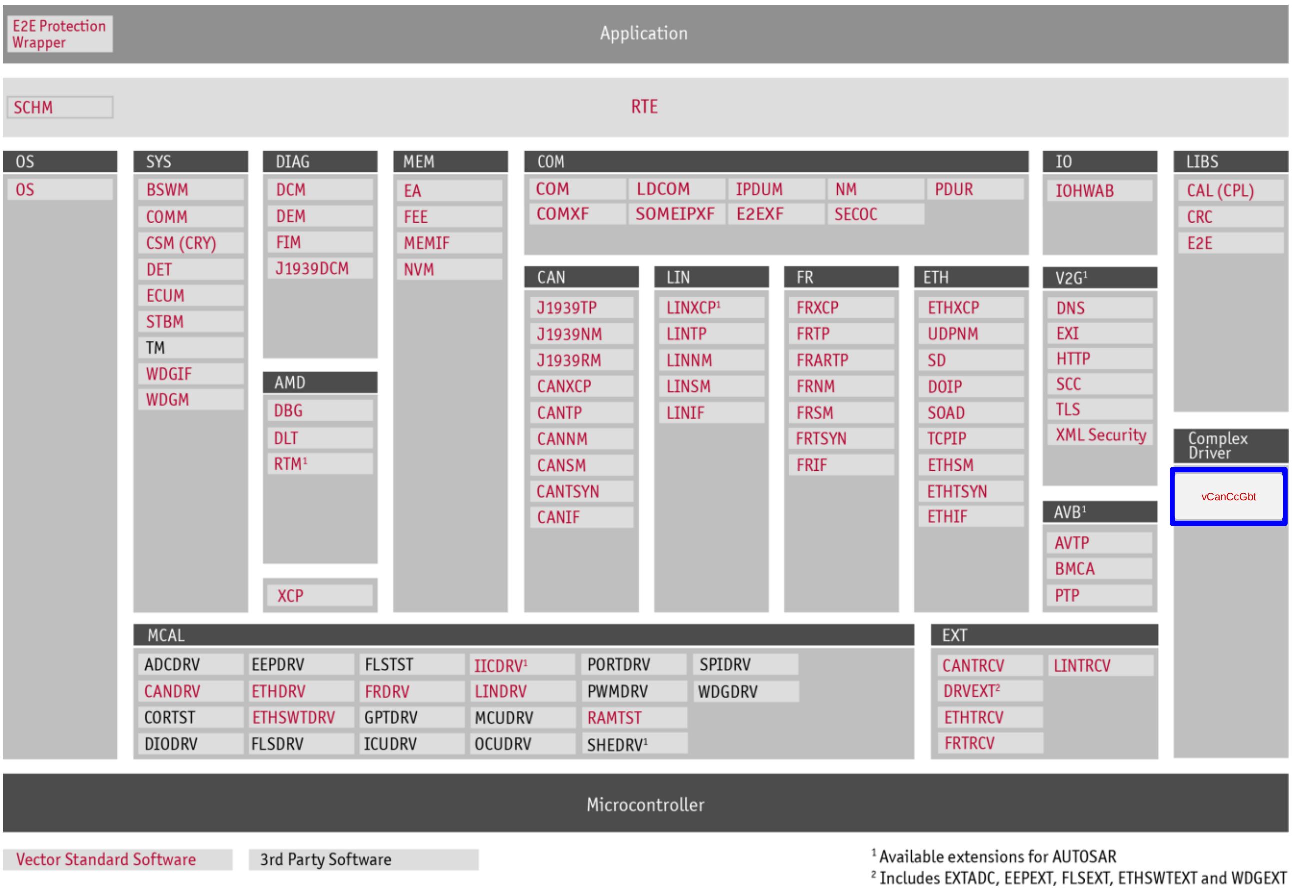

Figure 2-1 AUTOSAR 4.2 Architecture Overview

The next figure shows the interfaces to adjacent modules of the vCanCcGbt module. These interfaces are described in chapter 5.

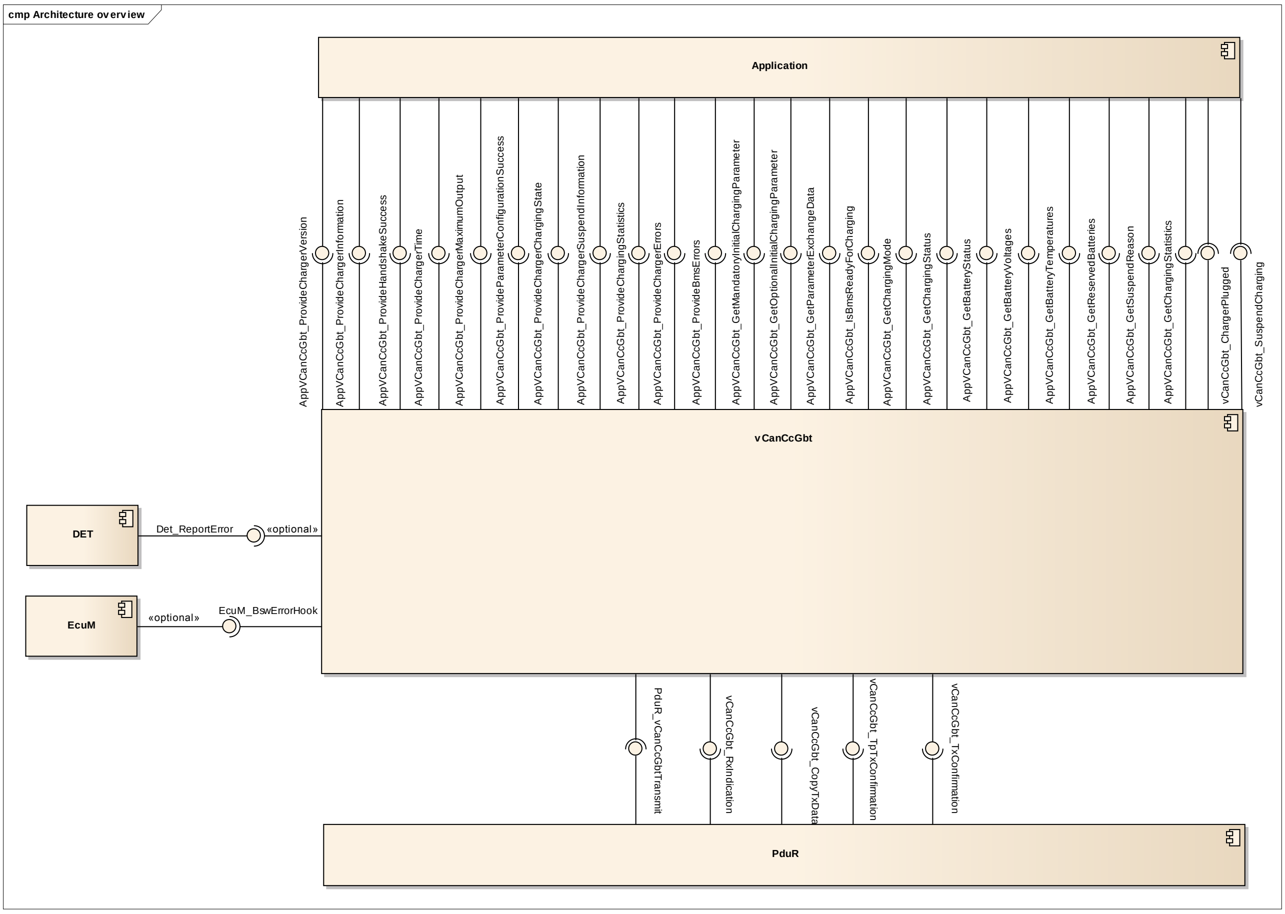
cmp Architecture overview
Application
AppVCanCcGbt_ProvideChargerVersion
AppVCanCcGbt_ProvideChargerInformation
AppVCanCcGbt_ProvideHandshakeSuccess
AppVCanCcGbt_ProvideChargerTime
AppVCanCcGbt_ProvideChargerMaximumOutput
AppVCanCcGbt_ProvideParameterConfigurationSuccess
AppVCanCcGbt_ProvideChargerChargingState
AppVCanCcGbt_ProvideChargerSuspendInformation
AppVCanCcGbt_ProvideChargingStatistics
AppVCanCcGbt_ProvideChargerErrors
AppVCanCcGbt_ProvideBmsErrors
AppVCanCcGbt_GetMandatoryInitialChargingParameter
AppVCanCcGbt_GetOptionalInitialChargingParameter
AppVCanCcGbt_GetParameterExchangeData
AppVCanCcGbt_IsBmsReadyForCharging
AppVCanCcGbt_GetChargingMode
AppVCanCcGbt_GetChargingStatus
AppVCanCcGbt_GetBatteryStatus
AppVCanCcGbt_GetBatteryVoltages
AppVCanCcGbt_GetBatteryTemperatures
AppVCanCcGbt_GetReservedBatteries
AppVCanCcGbt_GetSuspendReason
AppVCanCcGbt_GetChargingStatistics
vCanCcGbt_ChargerPlugged
vCanCcGbt_SuspendCharging
vCanCcGbt
DET
Det_ReportError
«optional»
EcuM
«optional»
EcuM_BswErrorHook
PduR_vCanCcGbtTransmit
vCanCcGbt_RxIndication
vCanCcGbt_CopyTxData
vCanCcGbt_TpTxConfirmation
vCanCcGbt_TxConfirmation
PduR

Figure 2-2 Interfaces to adjacent modules of the Mirror module

3 Functional Description

3.1 Features

The features listed in the following tables cover the complete functionality specified for the vCanCcGbt.

- Table 3-1 Supported features
- Table 3-2 Not supported features
-

The following features specified in [3] are supported:

Supported Features
BMS side charging protocol communication

Table 3-1 Supported features

3.1.1 Deviations

The following features specified in [3] are not supported:

Not Supported Features
Diagnostic messages
Request handling

Table 3-2 Not supported features

3.2 Initialization

The vCanCcGbt uses a global state (vCanCcGbt_ModuleInitialized) to determine whether the module is initialized and operational. This state is initially set to VCANCCGBT_UNINIT. If initialization by startup code is not supported, the initialization routine should call vCanCcGbt_InitMemory() to set the global state to VCANCCGBT_UNINIT.

By calling vCanCcGbt_Init(), the vCanCcGbt module is set to the state VCANCCGBT_INIT, and internal states are set to their initial states. The module is now operational.

To stop the vCanCcGbt module, vCanCcGbt_DeInit() may be called, which sets the global state to VCANCCGBT_UNINIT again.

3.3 States

The vCanCcGbt module has a global state, and separate states for each Tx PDU and the charging protocol.

3.3.1 Global State

The global state is switched by the services vCanCcGbt_InitMemory(), vCanCcGbt_Init(), and vCanCcGbt_DeInit().

In the state VCANCCGBT_UNINIT, all services of the vCanCcGbt return immediately, reporting VCANCCGBT_E_UNINIT to the DET if enabled. If they have a return value, an error (typically E_NOT_OK) is returned.

In the state VCANCCGBT_INIT, services are operational.

3.3.2 Tx PDU State

Each transmitted PDU (which uses the J1939Tp) has its own state to ensure that a value provided to CanIf is not overwritten with a new one before the CanIf was able to transmit it on the CAN bus. This state is protected by an exclusive area.

The Tx PDU state depends on the vCanCcGbt_TpTxConfirmation() being called after each call to PduR_vCanCcGbtTransmit().

3.3.3 Charging State

The vCanCcGbt has a state for the charging process with a Charger. This state maintains a state machine which controls which phase of the communication protocol is active and which messages should be send and which information are provided to the upper layer. This state is protected by an exclusive area.

The upper layer must inform the vCanCcGbt when the physical connection to the Charger is established by calling the API function vCanCcGbt_ChargerIsPlugged(). This brings the state machine into the initial communication protocol state and starts the charging process. This API functions must be also called when the physical connection to the Charger is lost.

The upper layer must also inform the vCanCcGbt when the charging process should be stopped by calling the API function vCanCcGbt_SuspendCharging().

3.4 Main Functions

The vCanCcGbt_MainFunction() is used by the vCanCcGbt module to supervise the Tx confirmation timeout and to trigger the transmission of messages and inform the upper layer about received messages depending on the charging state. It is important that this main function is called with the timing configured via “Main Function Period.”

3.5 Error Handling

3.5.1 Development and Runtime Error Reporting

By default, development errors are reported to the DET using the service `Det_ReportError()` as specified in [1], if development error reporting is enabled (i.e. pre-compile parameter `VCANCCGBT_DEV_ERROR_DETECT==STD_ON`). Runtime errors are reported on the same way, if runtime error reporting is enabled (i.e. pre-compile parameter `VCANCCGBT_RUNTIME_ERROR_DETECT==STD_ON`).

If another module is used for development error reporting, the function prototype for reporting the error can be configured by the integrator, but must have the same signature as the service `Det_ReportError()`.

The reported vCanCcGbt ID is 255.

The reported service IDs identify the services which are described in section 5.2 as well as the callback functions described in section 5.4. The following table presents the service IDs and the related services and callback functions:

Service ID	Service
0x01	vCanCcGbt_Init()
0x02	vCanCcGbt_DeInit()
0x03	vCanCcGbt_GetVersionInfo()
0x04	vCanCcGbt_MainFunction()
0x40	vCanCcGbt_TxConfirmation()
0x41	vCanCcGbt_TpTxConfirmation()
0x42	vCanCcGbt_RxIndication()
0x43	vCanCcGbt_InitMemory()
0x44	vCanCcGbt_CopyTxData()
0x45	vCanCcGbt_ChargerPlugged()
0x46	vCanCcGbt_SuspendCharging()

Table 3-3 Service IDs

The development errors reported to DET are described in the following table:

Error Code	Description
0x01	An API was called while the module was uninitialized, i.e. before vCanCcGbt_Init of after vCanCcGbt_DeInit
0x02	The Init API was called twice, i.e. after vCanCcGbt_Init and before vCanCcGbt_DeInit
0x03	An API service was called with a NULL pointer
0x04	An API service was called with a wrong ID
0x05	vCanCcGbt_Init() function called with invalid init structure

Table 3-4 Development errors reported to DET

The runtime errors reported to DET are described in the following table:

Error Code	Description
0x81	Timeout of transmission confirmation callback
0x82	Untimely CHM message
0x83	Untimely CRM message
0x84	Untimely CTS message
0x85	Untimely CML message
0x86	Untimely CRO message
0x87	Untimely CCS message
0x88	Untimely CST message

Error Code	Description
0x89	Untimely CSD message
0x90	Untimely CEM message

Table 3-5 Runtime errors reported to DET

4 Integration

This chapter gives necessary information for the integration of the MICROSAR vCanCcGbt into an application environment of an ECU.

4.1 Scope of Delivery

The delivery of the vCanCcGbt contains the files which are described in the chapters 4.1.1 and 4.1.2.

4.1.1 Static Files

File Name	Description
vCanCcGbt.c	Implementation of the vCanCcGbt module
vCanCcGbt.h	Main header of the vCanCcGbt module
vCanCcGbt_Cbk.h	Callback header of the vCanCcGbt module
vCanCcGbt_Callout.h	Callout header of the vCanCcGbt module
vCanCcGbt_Types.h	Global types header of the vCanCcGbt module
vCanCcGbt_Int.h	Internal header of the vCanCcGbt module

Table 4-1 Static files

4.1.2 Dynamic Files

The dynamic files are generated by the configuration tool [config tool].

File Name	Description
vCanCcGbt_Cfg.h	Generated header file of vCanCcGbt containing pre-compile switches and providing symbolic defines
vCanCcGbt_Cfg.c	Generated source file of vCanCcGbt containing pre-compile time configurable parameters
vCanCcGbt_Lcfg.h	Generated header file of vCanCcGbt containing link time configurable preprocessor symbols
vCanCcGbt_Lcfg.c	Generated source file of vCanCcGbt containing link time configurable parameters
vCanCcGbt_PBcfg.h	Generated header file of vCanCcGbt containing post-build time configurable preprocessor symbols
vCanCcGbt_PBcfg.c	Generated source file of vCanCcGbt containing post-build time configurable parameters

Table 4-2 Generated files

4.2 Critical Sections

4.2.1 VCANCCGBT_EXCLUSIVE_AREA_TXPDU

This exclusive area is used to avoid modifications to the vCanCcGbt_TxPduState from different contexts, by preventing that the functions vCanCcGbt_TxConfirmation, TpTxConfirmation, vCanCcGbt_MainFunction interrupt themselves or each other.

A global interrupt lock is recommended, since the functions could be called from different interrupt or task contexts.

This exclusive area can be omitted if the following two conditions are fulfilled:

- The system is in a polling mode and uses no interrupts
- The vCanCcGbt and CanIf module are running in the same task context

4.2.2 VCANCCGBT_EXCLUSIVE_AREA_STATE

This exclusive are is used to avoid modifications to the vCanCcGbt_ChargeState from different contexts, by preventing that the functions vCanCcGbt_TpTxConfirmation, vCanCcGbt_MainFunction interrupt themselves or each other.

A global interrupt lock is recommended, since the functions could be called from different interrupt or task contexts.

This exclusive area can be omitted if the following two conditions are fulfilled:

- The system is in a polling mode and uses no interrupts
- The vCanCcGbt and CanIf module are running in the same task context

5 API Description

5.1 Type Definitions

The types defined by the vCanCcGbt are described in this chapter.

Type Name	C-Type	Description	Value Range
vCanCcGbt_CEMErrorType	enum	Error reason of received Charger error message.	E_VCANCCGBT_ERROR_NONE No error
			E_VCANCCGBT_ERROR_BRM Charger has BRM message not received in time
			E_VCANCCGBT_ERROR_BCP Charger has BCP message not received in time
			E_VCANCCGBT_ERROR_BRO Charger has BRO message not received in time
			E_VCANCCGBT_ERROR_BCS Charger has BCS message not received in time
			E_VCANCCGBT_ERROR_BSM Charger has BSM message not received in time
			E_VCANCCGBT_ERROR_BST Charger has BST message not received in time
			E_VCANCCGBT_ERROR_BSD Charger has BSD message not received in time

Table 5-1 Type definitions

vCanCcGbt_MandatoryInitialChargingParameterType

Struct Element Name	C-Type	Description	Value Range
Version	uint32	Version information	0 - 4.294.967.295
TotalMaximumPermissibleChargingVoltage	uint16	Total maximum permissible charging voltage	0 – 65.535
PowerStorageBatterySystemCapacity	uint16	Power storage battery system capacity	0 – 65.535
PowerStorageBatterySystemVoltage	uint16	Power storage battery system voltage	0 – 65.535
BatteryType	uint8	Battery type	0 – 255

Table 5-2 vCanCcGbt_MandatoryInitialChargingParameterType

vCanCcGbt_OptionalInitialChargingParameterType

Struct Element Name	C-Type	Description	Value Range
VehicleIdentificationNumber	uint8 array	Vehicle identification number	17 byte each 0 – 255
BMSSoftwareVersionNumber	uint8 array	BMS software version number	8 byte each 0 – 255
BatteryPackNo	uint32	Battery pack number	0 - 4.294.967.295
BatterySetChargingFrequency	uint32	Battery charging frequency	0 - 4.294.967.295
BatteryManufacturerName	uint8	Battery manufacturer name	0 – 255
BatterySetProductionDataYear	uint8	Battery set production year	0 – 255
BatterySetProductionDataMonth	uint8	Battery set production month	0 – 255
BatterySetProductionDataDay	uint8	Battery set production da	0 – 255
BatterySetPropertyRightMark	uint8	Battery set property right mark	0 – 255
IsBatteryManufacturerNameValid	boolean	Is associated filed valid	TRUE associated filed is valid FALSE associated filed is not valid
IsBatteryPackNoValid	boolean	Is associated filed valid	TRUE associated filed is valid FALSE associated filed is not valid
IsBatterySetProductionDataYearValid	boolean	Is associated filed valid	TRUE associated filed is valid FALSE associated filed is not valid
IsBatterySetProductionDataMonthValid	boolean	Is associated filed valid	TRUE associated filed is valid FALSE associated filed is not valid
IsBatterySetProductionDataDayValid	boolean	Is associated filed valid	TRUE associated filed is valid FALSE associated filed is not valid
IsBatterySetChargingFrequencyValid	boolean	Is associated filed valid	TRUE associated filed is valid FALSE associated filed is not valid
IsBatterySetPropertyRightMarkValid	boolean	Is associated filed valid	TRUE associated filed is valid FALSE associated filed is not valid

Struct Element Name	C-Type	Description	Value Range
IsVehicleIdentificationNumberValid	boolean	Is associated filed valid	TRUE associated filed is valid FALSE associated filed is not valid
IsBMSSoftwareVersionNumberValid	boolean	Is associated filed valid	TRUE associated filed is valid FALSE associated filed is not valid

Table 5-3 vCanCcGbt_OptionalInitialChargingParameterType

vCanCcGbt_MandatoryParameterExchangeDataType

Struct Element Name	C-Type	Description	Value Range
MaximumPermissibleChargingVoltageSingleBattery	uint16	Maximum permissible charging voltage of single battery	0 – 65.535
MaximumPermissibleChargingCurrent	uint16	Maximum permissible charging current	0 – 65.535
TotalNominalEnergyOfPowerStorageBattery	uint16	Total nominal energy of power storage battery	0 – 65.535
TotalMaximumPermissibleChargingVoltage	uint16	Total maximum permissible charging voltage	0 – 65.535
ChargeStatusPowerStorageBatteryVehicle	uint16	Charge status power storage battery vehicle	0 – 65.535
CurrentBatteryVoltagePowerStorageBatteryVehicle	uint16	Current battery voltage power storage battery vehicle	0 – 65.535
MaximumPermissibleTemperature	uint8	Maximum permissible temperature	0 – 255

Table 5-4 vCanCcGbt_MandatoryParameterExchangeDataType

vCanCcGbt_BCLDataType

Struct Element Name	C-Type	Description	Value Range
VoltageDemand	uint16	Voltage demand	0 – 65.535
CurrentDemand	uint16	Current demand	0 – 65.535
ChargingMode	uint8	Charging mode	0 – 255

Table 5-5 vCanCcGbt_BCLDataType

vCanCcGbt_BCSDataType

Struct Element Name	C-Type	Description	Value Range
MeasuredValueOfChargingVoltage	uint16	Measured value of charging voltage	0 – 65.535
MeasuredValueOfChargingCurrent	uint16	Measured value of charging current	0 – 65.535
HighestVoltageOfBattery	uint16	Highest voltage of battery	0 – 65.535
EstimatedRemainingChargingTime	uint16	Estimated remaining charging time	0 – 65.535
HighestVoltageOfBatteryGroupNumber	uint8	Highest voltage of battery group number	0 – 255
CurrentChargeState	uint8	Current charge state	0 – 255

Table 5-6 vCanCcGbt_BCSDataType

vCanCcGbt_BSMDataType

Struct Element Name	C-Type	Description	Value Range
SerialNumberHighestVoltgeBattery	uint8	Serial number of highest voltage battery	0 – 255
HighestTemperatureBattery	uint8	Highest temperature of batter	0 – 255
SerialNumberHighestTemperatureDetectionPoint	uint8	Serial number of highest temperature detection point	0 – 255
LowestTemperatureBattery	uint8	Lowest temperature battery	0 – 255
SerialNumberLowetTemperatureDetectionPoint	uint8	Serial number lowest temperature detection point	0 – 255
VoltageOfSingleBateryIsOverHighOrOverLow	uint8	Voltage of single battery is over high or over low	0 – 255
StateOfChargeOfBatte	uint8	State of charge of	0 – 255

Struct Element Name	C-Type	Description	Value Range
ryCompleteVehicle		battery for the complete vehicle	
ChargingOvercurrentBattery	uint8	Charging over current battery	0 – 255
ExcessTemperatureOfBattery	uint8	Excess temperature of battery	0 – 255
InsulationStateOfBattery	uint8	Insulation state of battery	0 – 255
ConnectionStateOfOutputConnectorOfBattery	uint8	Connection state of output connector of batter	0 – 255
ChargingPermissible	uint8	Charging permissible	0 – 255

Table 5-7 vCanCcGbt_BSMDataType

vCanCcGbt_BMVSingleType

Struct Element Name	C-Type	Description	Value Range
IsActive	boolean	Is element valid	TRUE The element is valid
			FALSE The element is not valid
BatteryVoltage	uint16	Battery voltage	0 – 65.535
BatteryGroupNumber	uint8	Battery group number	0 – 255

Table 5-8 vCanCcGbt_BMVSingleType

vCanCcGbt_BMVDataType

Struct Element Name	C-Type	Description	Value Range
BMVSignleBattery	vCanCcGbt_BMVSingleType	Single battery state	256 elements

Table 5-9 vCanCcGbt_BMVDataType

vCanCcGbt_BMTSingleType

Struct Element Name	C-Type	Description	Value Range
IsActive	boolean	Is element valid	TRUE The element is valid
			FALSE The element is not valid
BatteryTemperature	uint8	Battery Temperature	0 – 255

vCanCcGbt_BMTDataType

Struct Element Name	C-Type	Description	Value Range
BMTSignleTemperature	vCanCcGbt_BMTSingleType	Single temperature	128 elements

Table 5-10 vCanCcGbt_BMTDataType

vCanCcGbt_BSPSingleType

Struct Element Name	C-Type	Description	Value Range
IsActive	boolean	Is element valid	TRUE The element is valid
			FALSE The element is not valid
BatteryReserved	uint8	Battery reserved	0 – 255

Table 5-11 vCanCcGbt_BSPSingleType

vCanCcGbt_BSPDataType

Struct Element Name	C-Type	Description	Value Range
BSPSignleBatteryReserved	vCanCcGbt_BSPSingleType	Single battery reserved	16 elements

Table 5-12 vCanCcGbt_BSPDataType

vCanCcGbt_BSTDataType

Struct Element Name	C-Type	Description	Value Range
FaultCauseForBMSSuspendingCharging	uint16	Fault cause for BMS suspending the charging	0 – 65.535
CauseForBMSSuspendingCharging	uint8	Cause for BMS suspending the charging	0 – 255

Struct Element Name	C-Type	Description	Value Range
ErrorCauseForBMSSuspendingCharging	uint8	Error cause for BMS suspending the charging	0 – 255

Table 5-13 vCanCcGbt_BSTDataType

vCanCcGbt_BSDDataType

Struct Element Name	C-Type	Description	Value Range
MinimumVoltageSingleBattery	uint16	Minimum voltage of single battery	0 – 65.535
MaximumVoltageSingleBattery	uint16	Maximum voltage of single battery	0 – 65.535
SuspendStateOfCharge	uint8	Suspend state of charge	0 – 255
MinimumTemperatureBattery	uint8	Minimum temperature of battery	0 – 255
MaximumTemperatureBattery	uint8	Maximum temperature of battery	0 – 255

Table 5-14 vCanCcGbt_BSDDataType

5.2 Services provided by vCanCcGbt

5.2.1 vCanCcGbt_InitMemory

Prototype	
`void vCanCcGbt_InitMemory (void)`	
Parameter	
void	none
Return code	
void	none
Functional Description	
Sets the global vCanCcGbt state to uninitialized.	
Particularities and Limitations	
This function should be used if the vCanCcGbt is not initialized by startup code. Module is initialized. -	
Call context	
> TASK > This function is Synchronous > This function is Non-Reentrant	

Table 5-15 vCanCcGbt_InitMemory

5.2.2 vCanCcGbt_Init

Prototype	
void **vCanCcGbt_Init** (const vCanCcGbt_ConfigType *config)	
Parameter	
config [in]	Pointer to configuration data structure.
Return code	
void	none
Functional Description	
Initializes the vCanCcGbt.	
Particularities and Limitations	
The config parameter is only required if the configuration is variant or changed at post-build time. The module must be in the uninitialized state. -	
Call context	
> TASK > This function is Synchronous > This function is Non-Reentrant	

Table 5-16 vCanCcGbt_Init

5.2.3 vCanCcGbt_DeInit

Prototype	
void **vCanCcGbt_DeInit** (void)	
Parameter	
void	none
Return code	
void	none
Functional Description	
Resets the vCanCcGbt module to the uninitialized state.	
Particularities and Limitations	
The module must be in the initialized state. The module is not truly shut down before all services and callback functions have terminated.	
Call context	
> TASK > This function is Synchronous > This function is Non-Reentrant	

Table 5-17 vCanCcGbt_DeInit

5.2.4 vCanCcGbt_GetVersionInfo

<table>
<tr><th colspan="2">Prototype</th></tr>
<tr><td colspan="2">void <b>vCanCcGbt_GetVersionInfo</b> (Std_VersionInfoType *versionInfo)</td></tr>
<tr><th colspan="2">Parameter</th></tr>
<tr><td>VersionInfo [out]</td><td>Pointer to the location where the version information shall be stored.</td></tr>
<tr><th colspan="2">Return code</th></tr>
<tr><td>void</td><td>none</td></tr>
<tr><th colspan="2">Functional Description</th></tr>
<tr><td colspan="2">Returns the version information of the vCanCcGbt.</td></tr>
<tr><th colspan="2">Particularities and Limitations</th></tr>
<tr><td colspan="2">The VersionInfo parameter must not be NULL.
-</td></tr>
<tr><th colspan="2">Call context</th></tr>
<tr><td colspan="2">TASK|ISR
This function is Synchronous
This function is Non-Reentrant</td></tr>
</table>

Table 5-18 vCanCcGbt_GetVersionInfo

5.2.5 vCanCcGbt_ChargerPlugged

<table>
<tr><th colspan="2">Prototype</th></tr>
<tr><td colspan="2">void <b>vCanCcGbt_ChargerPlugged</b> (boolean isPlugged)</td></tr>
<tr><th colspan="2">Parameter</th></tr>
<tr><td>isPlugged [in]</td><td></td></tr>
<tr><th colspan="2">Return code</th></tr>
<tr><td>void</td><td>none</td></tr>
<tr><th colspan="2">Functional Description</th></tr>
<tr><td colspan="2">Function to set the state of the physical connetion to the Charger.</td></tr>
<tr><th colspan="2">Particularities and Limitations</th></tr>
<tr><td colspan="2">Module is initialized.
-</td></tr>
<tr><th colspan="2">Call context</th></tr>
<tr><td colspan="2">TASK|ISR
This function is Synchronous
This function is Non-Reentrant</td></tr>
</table>

Table 5-19 vCanCcGbt_ChargerPlugged

5.2.6 vCanCcGbt_SuspendCharging

Prototype	
`void` **`vCanCcGbt_SuspendCharging`** `(boolean isSuspended)`	
Parameter	
isSuspended [in]	
Return code	
void	none
Functional Description	
Function to indicate the vCanCcGbt that the charging is finished.	
Particularities and Limitations	
Module is initialized. -	
Call context	
> TASK|ISR > This function is Synchronous > This function is Non-Reentrant	

Table 5-20 vCanCcGbt_SuspendCharging

5.3 Services used by vCanCcGbt

In the following table services provided by other components, which are used by the vCanCcGbt are listed. For details about prototype and functionality refer to the documentation of the providing component.

Component	API
Default Error Tracer	Det_ReportError
PDU Router	PduR_vCanCcGbtTransmit
EcuM	EcuM_BswErrorHook
VStdLib	VStdLib_MemCpy
VStdLib	VStdLib_MemSet

Table 5-21 Services used by the vCanCcGbt

5.4 Callback Functions

This chapter describes the callback functions that are implemented by the vCanCcGbt and can be invoked by other modules. The prototypes of the callback functions are provided in the header file `vCanCcGbt_Cbk.h` by the vCanCcGbt.

5.4.1 vCanCcGbt_RxIndication

Prototype	
`void` **`vCanCcGbt_RxIndication`** `(PduIdType rxPduId, const PduInfoType *pduInfoPtr)`	
Parameter	
RxPduId [in]	ID of the received N-PDU.
PduInfoPtr [in]	Contains the length (SduLength) of the received N-PDU and a pointer to a buffer (SduDataPtr) containing the N-PDU and MetaData.
Return code	
void	none
Functional Description	
Indicates the reception of an N-PDU.	
Particularities and Limitations	
Module is initialized. Indication of a received PDU from a lower layer communication interface module.	
Call context	
> TASK|ISR > This function is Synchronous > This function is Reentrant	

Table 5-22 vCanCcGbt_RxIndication

5.4.2 vCanCcGbt_TxConfirmation

Prototype	
`void` **`vCanCcGbt_TxConfirmation`** `(PduIdType txPduId)`	
Parameter	
txPduId [in]	ID of the N-PDU that has been transmitted.
Return code	
void	none
Functional Description	
Confirms the successful transmission of an N-PDU.	
Particularities and Limitations	
Module is initialized. The lower layer communication interface module confirms the transmission of a PDU, or the failure to transmit a PDU.	
Call context	
> TASK|ISR > This function is Synchronous > This function is Reentrant	

Table 5-23 vCanCcGbt_TxConfirmation

5.4.3 vCanCcGbt_CopyTxData

Prototype	
`BufReq_ReturnType` **`vCanCcGbt_CopyTxData`** `(PduIdType id, PduInfoType *info, RetryInfoType *retry, PduLengthType *availableDataPtr)`	
Parameter	
id [in]	PDU id
info [in]	Contains the length of the next data block that should be send and a pointer to the buffer.
retry [in]	
availableDataPtr [in]	
Return code	
BufReq_ReturnType	BUFREQ_OK BUFREQ_E_NOT_OK
Functional Description	
This function indicats the request for new data for a TP transmission.	
Particularities and Limitations	
Module is initialized. -	
Call context	
> TASK|ISR > This function is Synchronous > This function is Non-Reentrant	

Table 5-24 vCanCcGbt_CopyTxData

5.4.4 vCanCcGbt_TpTxConfirmation

Prototype	
`void` **`vCanCcGbt_TpTxConfirmation`** `(PduIdType txPduId, Std_ReturnType result)`	
Parameter	
txPduId [in]	ID of the PDU that has been transmitted.
result [in]	Result of the transmssion.
Return code	
void	none
Functional Description	
Confirms the successful transmission of an TP PDU.	
Particularities and Limitations	
Module is initialized. -	

Call context
> TASK|ISR > This function is Synchronous > This function is Non-Reentrant

Table 5-25 vCanCcGbt_TpTxConfirmation

5.4.5 vCanCcGbt_StartOfReception

Prototype	
`BufReq_ReturnType` **`vCanCcGbt_StartOfReception`** `(PduIdType pduId, PduInfoType *info, PduLengthType tpSduLength, PduLengthType *bufferSizePtr)`	
Parameter	
pduId [in]	Identification of the N-SDU.
info [in]	Pointer to a PduInfoType structure containing the MetaData and MetaDataLength.
tpSduLength [in]	Total length of the N-SDU to be received.
bufferSizePtr [in]	Available receive buffer in the receiving module. This parameter will be used to compute the block size in the transport protocol module.
Return code	
BufReq_ReturnType	BUFREQ_OK: Connection has been accepted. bufferSizePtr indicates the available receive buffer; reception is continued. If no buffer of the requested size is available, a receive buffer size of 0 shall be indicated by bufferSizePtr. BUFREQ_E_NOT_OK: Connection has been rejected; reception is aborted. bufferSizePtr remains unchanged. BUFREQ_E_OVFL: No buffer of the required length can be provided; reception is aborted. bufferSizePtr remains unchanged.
Functional Description	
This function is called at the start of receiving a TP PDU.	
Particularities and Limitations	
Module is initialized. -	
Call context	
> TASK|ISR > This function is Synchronous > This function is Non-Reentrant	

Table 5-26 vCanCcGbt_StartOfReception

5.4.6 vCanCcGbt_CopyRxData

Prototype	
`BufReq_ReturnType` **`vCanCcGbt_CopyRxData`** `(PduIdType pduId, PduInfoType *info, PduLengthType *availableDataPtr)`	
Parameter	
pduId [in]	Identification of the received N-SDU.
info [in]	Provides the source buffer (SduDataPtr) and the number of bytes to be copied (SduLength).
bufferSizePtr [in]	Available receive buffer after data has been copied.
Return code	
BufReq_ReturnType	BUFREQ_OK: Data copied successfully. BUFREQ_E_NOT_OK: Data was not copied because an error occurred.
Functional Description	
This function is called to provide the received data of an N-SDU segment (N-PDU) to the upper layer.	
Particularities and Limitations	
Module is initialized. Each call to this function provides the next part of the N-SDU data. The size of the remaining data is written to the position indicated by bufferSizePtr.	
Call context	
> TASK|ISR > This function is Synchronous > This function is Non-Reentrant	

Table 5-27 vCanCcGbt_CopyRxData

5.4.7 vCanCcGbt_TpRxIndication

Prototype	
`void` **`vCanCcGbt_TpRxIndication`** `(PduIdType PduId, Std_ReturnType result)`	
Parameter	
PduId [in]	Identification of the received N-SDU.
result [in]	Result of the reception.
Return code	
void	none
Functional Description	
This function is called after a N-SDU has been received via the TP API.	
Particularities and Limitations	
Module is initialized. The result indicates whether the reception was successful or not.	
Call context	

> TASK|ISR > This function is Synchronous > This function is Non-Reentrant

Table 5-28 vCanCcGbt_TpRxIndication

5.5 Configurable Interfaces

5.5.1 Callout Functions

At its configurable interfaces the vCanCcGbt defines callout functions. The declarations of the callout functions are provided by the BSW module, i.e. the vCanCcGbt. It is the integrator's task to provide the corresponding function definitions. The definitions of the callouts can be adjusted to the system's needs. The vCanCcGbt callout function declarations are described in the following tables:

5.5.1.1 AppVCanCcGbt_ProvideChargerVersion

Prototype	
`void` **`AppVCanCcGbt_ProvideChargerVersion`** `(uint8 chargerRevision, uint16 chargerVersion)`	
Parameter	
chargerRevision [in]	Revision of the Charger
chargerVersion [in]	Version of the Charger
Return code	
void	none
Functional Description	
Indicates the reception of a CHM message.	
Particularities and Limitations	
Module is initialized. Handshake phase.	
Call context	
> TASK|ISR > This function is Synchronous > This function is Non-Reentrant	

Table 5-29 AppVCanCcGbt_ProvideChargerVersion

5.5.1.2 AppVCanCcGbt_ProvideChargerInformation

Prototype
`void` **`AppVCanCcGbt_ProvideChargerInformation`** `(uint32 chargerNo, uint8 chargerStationLocationCode[4])`

Parameter	
chargerNo [in]	Charger number
chargerStationLocationCode [in]	Charger station location code
Return code	
Void	none
Functional Description	
Indicates the reception of a CRM message.	
Particularities and Limitations	
Module is initialized. Handshake phase.	
Call context	
> TASK|ISR > This function is Synchronous > This function is Non-Reentrant	

Table 5-30 AppVCanCcGbt_ProvideChargerInformation

5.5.1.3 AppVCanCcGbt_ProvideHandshakeSuccess

Prototype	
void **AppVCanCcGbt_ProvideHandshakeSuccess** (void)	
Parameter	
Void	none
Return code	
void	none
Functional Description	
Indicates the success of the handshake phase.	
Particularities and Limitations	
Module is initialized. Handshake phase.	
Call context	
> TASK|ISR > This function is Synchronous > This function is Non-Reentrant	

Table 5-31 AppVCanCcGbt_ProvideHandshakeSuccess

5.5.1.4 AppVCanCcGbt_ProvideChargerTime

Prototype	
`void **AppVCanCcGbt_ProvideChargerTime** (uint8 second, uint8 minute, uint8 hour, uint8 day, uint8 month, uint16 year)`	
Parameter	
second [in]	Seconds of the Chargers time.
minute [in]	Minutes of the Chargers time.
hour [in]	Hour of the Chargers time.
day [in]	Day of the Chargers time.
month [in]	Month of the Chargers time.
year [in]	Year of the Chargers time.
Return code	
void	none
Functional Description	
Indicates the reception of a CTS message.	
Particularities and Limitations	
Module is initialized. Parameter exchange phase.	
Call context	
> TASK|ISR > This function is Synchronous > This function is Non-Reentrant	

Table 5-32 AppVCanCcGbt_ProvideChargerTime

5.5.1.5 AppVCanCcGbt_ProvideChargerMaximumOutput

Prototype	
`void **AppVCanCcGbt_ProvideChargerMaximumOutput** (uint16 maxOutputVoltage, uint16 minOutputVoltage, uint16 maxOutputCurrent, uint16 minOutputCurrent)`	
Parameter	
maxOutputVoltage [in]	Maximum voltage output
minOutputVoltage [in]	Minimum voltage output
maxOutputCurrent [in]	Maximum current output
minOutputCurrent [in]	Minimum current output
Return code	
void	none
Functional Description	
Indicates the reception of a CML message.	

Particularities and Limitations
Module is initialized. Parameter exchange phase.
Call context
> TASK|ISR > This function is Synchronous > This function is Non-Reentrant

Table 5-33 AppVCanCcGbt_ProvideChargerMaximumOutput

5.5.1.6 AppVCanCcGbt_ProvideParameterConfigurationSuccess

Prototype	
void **AppVCanCcGbt_ProvideParameterConfigurationSuccess** (uint8 chargerChargingState)	
Parameter	
chargerChargingState [in]	State of the Charger
Return code	
void	none
Functional Description	
Indicates the reception of a CRO message.	
Particularities and Limitations	
Module is initialized. Parameter exchange phase.	
Call context	
> TASK|ISR > This function is Synchronous > This function is Non-Reentrant	

Table 5-34 AppVCanCcGbt_ProvideParameterConfigurationSuccess

5.5.1.7 AppVCanCcGbt_ProvideChargerChargingState

Prototype	
void **AppVCanCcGbt_ProvideChargerChargingState** (uint16 voltgeOutput, uint16 currentOutput, uint16 chargingTime, uint8 chargingPermissible)	
Parameter	
voltgeOutput [in]	Voltage output
currentOutput [in]	Current output
chargingTime [in]	Charging time

chargingPermissible [in]	Charging permissible
Return code	
void	none
Functional Description	
Indicates the reception of a CCS message.	
Particularities and Limitations	
Module is initialized. Charging phase.	
Call context	
> TASK|ISR > This function is Synchronous > This function is Non-Reentrant	

Table 5-35 AppVCanCcGbt_ProvideChargerChargingState

5.5.1.8 AppVCanCcGbt_ProvideChargerSuspendInformation

Prototype

```
void AppVCanCcGbt_ProvideChargerSuspendInformation (uint8
causeChargerSuspendingCharging, uint16 faultCauseChargerSuspendingCharging,
uint8 errorCauseChargerSuspendingCharging)
```

Parameter	
causeChargerSuspendingCharging [in]	Charger cause for suspending the charging
faultCauseChargerSuspendingCharging [in]	Charger fault cause for suspending the charging
errorCauseChargerSuspendingCharging [in]	Charger error cause for suspending the charging
Return code	
void	none
Functional Description	
Indicates the reception of a CST message.	
Particularities and Limitations	
Module is initialized. Charging phase.	
Call context	
> TASK|ISR > This function is Synchronous > This function is Non-Reentrant	

Table 5-36 AppVCanCcGbt_ProvideChargerSuspendInformation

5.5.1.9 AppVCanCcGbt_ProvideChargingStatistics

Prototype	
void **AppVCanCcGbt_ProvideChargingStatistics** (uint16 cumulativeChargingTime, uint16 outputEnergy, uint32 chargerNumber)	
Parameter	
cumulativeChargingTime [in]	Cumulative charging time
outputEnergy [in]	Output energy
chargerNumber [in]	Charger number
Return code	
void	none
Functional Description	
Indicates the reception of a CSD message.	
Particularities and Limitations	
Module is initialized. Charge statistic phase.	
Call context	
> TASK|ISR > This function is Synchronous > This function is Non-Reentrant	

Table 5-37 AppVCanCcGbt_ProvideChargingStatistics

5.5.1.10 AppVCanCcGbt_ProvideChargerErrors

Prototype	
void **AppVCanCcGbt_ProvideChargerErrors** (uint8 timeoutValueBRM, uint8 timeoutValueBCP, uint8 timeoutValueBRO, uint8 timeoutValueBCS, uint8 timeoutValueBSM, uint8 timeoutValueBST, uint8 timeoutValueBSD)	
Parameter	
timeoutValueBRM [in]	BRM error value
timeoutValueBCP [in]	BCP error value
timeoutValueBRO [in]	BRO error value
timeoutValueBCS [in]	BCS error value
timeoutValueBSM [in]	BSM error value
timeoutValueBST [in]	BST error value
timeoutValueBSD [in]	BSD error value
Return code	
void	none

Functional Description
Indicates the reception of a CEM message.
Particularities and Limitations
Module is initialized. Error phase.
Call context
> TASK|ISR > This function is Synchronous > This function is Non-Reentrant

Table 5-38 AppVCanCcGbt_ProvideChargerErrors

5.5.1.11 AppVCanCcGbt_ProvideBmsErrors

Prototype

```
void AppVCanCcGbt_ProvideBmsErrors (boolean crm00Timeout, boolean crmAATimeout,
boolean ctsTimeout, boolean croTimeout, boolean ccsTimeout, boolean cstTimeout,
boolean csdTimeout)
```

Parameter	
crm00Timeout [in]	CRM00 timeout
crmAATimeout [in]	CRMAA timeout
ctsTimeout [in]	CTS timeout
croTimeout [in]	CRO timeout
ccsTimeout [in]	CCS timeout
cstTimeout [in]	CST timeout
csdTimeout [in]	CSD timeout
Return code	
void	none

Functional Description
Indicates the detection of a timeout of the reception of a Charger message.
Particularities and Limitations
Module is initialized. Error phase.
Call context
> TASK|ISR > This function is Synchronous > This function is Non-Reentrant

Table 5-39 AppVCanCcGbt_ProvideBmsErrors

5.5.1.12 AppVCanCcGbt_GetMandatoryInitialChargingParameter

Prototype	
void **AppVCanCcGbt_GetMandatoryInitialChargingParameter** (vCanCcGbt_MandatoryInitialChargingParameterType *mandatoryInitialChargingParameter)	
Parameter	
chargingParameter [in]	Struct with input parameter from Application.
Return code	
void	none
Functional Description	
Function returns the mandatory initial application specific parameter needed for charging.	
Particularities and Limitations	
Module is initialized. Handshake phase.	
Call context	
> TASK|ISR > This function is Synchronous > This function is Non-Reentrant	

Table 5-40 AppVCanCcGbt_GetMandatoryInitialChargingParameter

5.5.1.13 AppVCanCcGbt_GetOptionalInitialChargingParameter

Prototype	
void **AppVCanCcGbt_GetOptionalInitialChargingParameter** (vCanCcGbt_OptionalInitialChargingParameterType *optionalInitialChargingParameter)	
Parameter	
optionalInitialChargingParameter [in]	Struct with input parameter from Application.
Return code	
void	none
Functional Description	
Function returns the optional initial application specific parameter needed for charging.	
Particularities and Limitations	
Module is initialized. Handshake phase.	
Call context	
> TASK|ISR > This function is Synchronous > This function is Non-Reentrant	

Table 5-41 AppVCanCcGbt_GetOptionalInitialChargingParameter

5.5.1.14 AppVCanCcGbt_GetParameterExchangeData

Prototype	
void **AppVCanCcGbt_GetParameterExchangeData** (vCanCcGbt_MandatoryParameterExchangeDataType *parameterExchangeData)	
Parameter	
optionalInitialChargingParameter [in]	Struct with input parameter from Application.
Return code	
void	none
Functional Description	
Function returns the optional initial application specific parameter needed for charging.	
Particularities and Limitations	
Module is initialized. Parameter exchange phase.	
Call context	
> TASK|ISR > This function is Synchronous > This function is Non-Reentrant	

Table 5-42 AppVCanCcGbt_GetParameterExchangeData

5.5.1.15 AppVCanCcGbt_IsBmsReadyForCharging

Prototype	
boolean **AppVCanCcGbt_IsBmsReadyForCharging** (void)	
Parameter	
void	none
Return code	
boolean	TRUE BMS is ready for charging phase FALSE BMS is not read for charging phase
Functional Description	
Function returns weather the BMS is ready for charging phase or not.	
Particularities and Limitations	
Module is initialized. -	
Call context	
> TASK|ISR	

> This function is Synchronous > This function is Non-Reentrant

Table 5-43 AppVCanCcGbt_IsBmsReadyForCharging

5.5.1.16 AppVCanCcGbt_GetChargingMode

Prototype	
`Std_ReturnType` **`AppVCanCcGbt_GetChargingMode`** `(vCanCcGbt_BCLDataType *bclData)`	
Parameter	
bclData [in]	Data buffer for BCL message data
Return code	
Std_ReturnType	E_OK - New data are available. E_NOT_OK - No new data are available.
Functional Description	
Requests the data for a BCL message.	
Particularities and Limitations	
Module is initialized. Parameter exchange phase.	
Call context	
> TASK|ISR > This function is Synchronous > This function is Non-Reentrant	

Table 5-44 AppVCanCcGbt_GetChargingMode

5.5.1.17 AppVCanCcGbt_GetChargingStatus

Prototype	
`Std_ReturnType` **`AppVCanCcGbt_GetChargingStatus`** `(vCanCcGbt_BCSDataType *bcsData)`	
Parameter	
bcsData [in]	Data buffer for BCS message data
Return code	
Std_ReturnType	E_OK - New data are available. E_NOT_OK - No new data are available.
Functional Description	
Requests the data for a BCS message.	
Particularities and Limitations	
Module is initialized. Charging phase.	
Call context	
> TASK|ISR	

> This function is Synchronous > This function is Non-Reentrant

Table 5-45 AppVCanCcGbt_GetChargingStatus

5.5.1.18 AppVCanCcGbt_GetBatteryStatus

Prototype	
`Std_ReturnType` **`AppVCanCcGbt_GetBatteryStatus`** `(vCanCcGbt_BSMDataType *bsmData)`	
Parameter	
bsmData [in]	Data buffer for BSM message data
Return code	
Std_ReturnType	E_OK - New data are available. E_NOT_OK - No new data are available.
Functional Description	
Requests the data for a BSM message.	
Particularities and Limitations	
Module is initialized. Charging phase.	
Call context	
> TASK|ISR > This function is Synchronous > This function is Non-Reentrant	

Table 5-46 AppVCanCcGbt_GetBatteryStatus

5.5.1.19 AppVCanCcGbt_GetBatteryVoltages

Prototype	
`Std_ReturnType` **`AppVCanCcGbt_GetBatteryVoltages`** `(vCanCcGbt_BMVDataType *bmvData)`	
Parameter	
bmvData [in]	Data buffer for BMV message data
Return code	
Std_ReturnType	E_OK - At least one data field is active. E_NOT_OK - No data field is active.
Functional Description	
Requests the data for a BMV message.	
Particularities and Limitations	
Module is initialized. Charging phase.	
Call context	
> TASK|ISR	

> This function is Synchronous > This function is Non-Reentrant

Table 5-47 AppVCanCcGbt_GetBatteryVoltages

5.5.1.20 AppVCanCcGbt_GetBatteryTemperatures

Prototype	
Std_ReturnType **AppVCanCcGbt_GetBatteryTemperatures** (vCanCcGbt_BMTDataType *bmtData)	
Parameter	
bmtData [in]	Data buffer for BMT message data
Return code	
Std_ReturnType	E_OK - At least one data field is active. E_NOT_OK - No data field is active.
Functional Description	
Requests the data for a BMT message.	
Particularities and Limitations	
Module is initialized. Charging phase.	
Call context	
> TASK|ISR > This function is Synchronous > This function is Non-Reentrant	

Table 5-48 AppVCanCcGbt_GetBatteryTemperatures

5.5.1.21 AppVCanCcGbt_GetReservedBatteries

Prototype	
Std_ReturnType **AppVCanCcGbt_GetReservedBatteries** (vCanCcGbt_BSPDataType *bspData)	
Parameter	
bspData [in]	Data buffer for BSP message data
Return code	
Std_ReturnType	E_OK - At least one data field is active. E_NOT_OK - No data field is active.
Functional Description	
Requests the data for a BSP message.	
Particularities and Limitations	
Module is initialized. Charging phase.	

<table>
<tr><th>Call context</th></tr>
<tr><td>&gt; TASK|ISR
> This function is Synchronous
> This function is Non-Reentrant</td></tr>
</table>

Table 5-49 AppVCanCcGbt_GetReservedBatteries

5.5.1.22 AppVCanCcGbt_GetSuspendReason

<table>
<tr><th colspan="2">Prototype</th></tr>
<tr><td colspan="2">Std_ReturnType <b>AppVCanCcGbt_GetSuspendReason</b> (vCanCcGbt_BSTDataType *bstData)</td></tr>
<tr><th colspan="2">Parameter</th></tr>
<tr><td>bstData [in]</td><td>Data buffer for BST message data</td></tr>
<tr><th colspan="2">Return code</th></tr>
<tr><td>Std_ReturnType</td><td>E_OK - New data are available. E_NOT_OK - No new data are available.</td></tr>
<tr><th colspan="2">Functional Description</th></tr>
<tr><td colspan="2">Requests the data for a BST message.</td></tr>
<tr><th colspan="2">Particularities and Limitations</th></tr>
<tr><td colspan="2">Module is initialized.
Charging phase.</td></tr>
<tr><th colspan="2">Call context</th></tr>
<tr><td colspan="2">&gt; TASK|ISR
> This function is Synchronous
> This function is Non-Reentrant</td></tr>
</table>

Table 5-50 AppVCanCcGbt_GetSuspendReason

5.5.1.23 AppVCanCcGbt_GetChargingStatistics

<table>
<tr><th colspan="2">Prototype</th></tr>
<tr><td colspan="2">Std_ReturnType <b>AppVCanCcGbt_GetChargingStatistics</b> (vCanCcGbt_BSDDataType *bsdData)</td></tr>
<tr><th colspan="2">Parameter</th></tr>
<tr><td>bsdData [in]</td><td>Data buffer for BSD message data</td></tr>
<tr><th colspan="2">Return code</th></tr>
<tr><td>Std_ReturnType</td><td>E_OK - New data are available. E_NOT_OK - No new data are available.</td></tr>
<tr><th colspan="2">Functional Description</th></tr>
<tr><td colspan="2">Requests the data for a BSD message.</td></tr>
<tr><th colspan="2">Particularities and Limitations</th></tr>
<tr><td colspan="2">Module is initialized.</td></tr>
</table>

Charge statistic phase.
Call context
> TASK|ISR > This function is Synchronous > This function is Non-Reentrant

Table 5-51 AppVCanCcGbt_GetChargingStatistics

6 Configuration

The vCanCcGbt module can be configured through the Vector configuration and generation tool DaVinci Configurator.

6.1 Configuration Variants

The vCanCcGbt supports the configuration variants

- `VARIANT-PRE-COMPILE`
- `VARIANT-POST-BUILD-LOADABLE`
- `VARIANT-POST-BUILD-SEELCTABLE`

The configuration classes of the vCanCcGbt parameters depend on the supported configuration variants. For their definitions please see the vCanCcGbt_bswmd.arxml file.

6.2 Configuration of Post-Build

The configuration of post-build loadable is described in [4].

7 Glossary and Abbreviations

7.1 Glossary

Term	Description
DaVinci Configurator	Generation tool for MICROSAR components.

Table 7-1 Glossary

7.2 Abbreviations

Abbreviation	Description
API	Application Programming Interface
AUTOSAR	Automotive Open System Architecture
BSW	Basis Software
DET	Development Error Tracer
ECU	Electronic Control Unit
ISR	Interrupt Service Routine
MICROSAR	Microcontroller Open System Architecture (the Vector AUTOSAR solution)
RTE	Runtime Environment
SRS	Software Requirement Specification
SWC	Software Component
SWS	Software Specification

Table 7-2 Abbreviations

8 Contact

Visit our website for more information on

- News
- Products
- Demo software
- Support
- Training data
- Addresses

www.vector.com