Vector CAN Driver

!

!

	!	

Document Information

History

Author	Date	Version	Remarks
!	- -		! ! ! ! ! ! !
!	- -		! ! ! ! ! !
!	- -		! ! ! !
!	- -		! ! ! ! ! ! ! ! ! ! ! ! ! !
!	- -		! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !
!	- -		! ! ! ! !
!	- -		! ! ! ! ! ! ! ! !

! - ! ! ! !

Please note

! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! !

Contents

!-!   !   . . . .

Illustrations

Tables

1 Introduction

! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! TechnicalReference_CANDriver.pdf ! !
! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! !
! ! !

! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! !

2   Important References

! ! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! !

Driver Version	Supported Compilers	Supported Derivatives	Hardware Manufacturer Document			Version
!	! ! !	!	! ! ! ! !	! !	!	! !
			! ! ! ! ! !		!!	!
		!¹	! ! ! ! ! !		!	! !
			! ! ! ! ! !			!
			! ! ! ! ! !			!
			! ! ! ! !			!

! - ! ! !

Driver Version: ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !
             ! ! ! ! ! ! ! ! !
Supported Compilers: ! ! ! ! ! ! ! !
Supported Derivatives: ! ! ! ! ! ! ! ! ! !
Hardware Manufacturer Document: ! ! ! ! ! ! ! ! ! !
Version: ! ! ! ! ! ! ! ! ! ! ! ! !

3 Usage of Controller Features

3.1   [#hw_comObj] - Communication Objects

!　　　　!　!　　　　!　!　　　!　　　　!　!　　　　　!

```
┌──────────────────────┐
│     [0] RX Buffer      │ ⎫
└──────────────────────┘ ⎪  Rx FullCAN
           │             ⎬  nRXMBmax receive buffers
┌──────────────────────┐ ⎪  shared over all channels
│   [max 127] RX Buffer  │ ⎭
└──────────────────────┘

┌──────────────────────┐
│     [128] RX FIFO      │ ⎫
└──────────────────────┘ ⎪  Rx BasicCAN
           │             ⎬  8 receive FIFOs
┌──────────────────────┐ ⎪  shared over all channels
│     [135] RX FIFO      │ ⎭
└──────────────────────┘

┌──────────────────────┐
│     [160] TX Buffer    │ ⎫
└──────────────────────┘ ⎪  Tx Normal, LowLevel, FullCAN
           │             ⎬  16 transmit buffers
┌──────────────────────┐ ⎪  physical channel CAN0
│     [175] TX Buffer    │ ⎭
└──────────────────────┘

┌──────────────────────┐
│     [176] TX Buffer    │ ⎫
└──────────────────────┘ ⎪  Tx Normal, LowLevel, FullCAN
           │             ⎬  16 transmit buffers
┌──────────────────────┐ ⎪  physical channel CAN1
│     [191] TX Buffer    │ ⎭
└──────────────────────┘
          ..

┌──────────────────────┐
│     [272] TX Buffer    │ ⎫
└──────────────────────┘ ⎪  Tx Normal, LowLevel, FullCAN
           │             ⎬  16 transmit buffers
┌──────────────────────┐ ⎪  physical channel CAN7
│     [287] TX Buffer    │ ⎭
└──────────────────────┘
```

!　-　　　　　!　　　!

> ℹ️ **Note**
>
> !　-　!　　　　!　!　　　　　!　　　　!　　　　　!-!　　!　　　!　　　!　　　　!　!　!
> !　　　　　!　　!　!　!　　　　　!　　　!　!　!　!　　　!　!　　　　　!　　　　　!
> !　!　　　　!　　!　!　　　!　!　　　!　!　　　!　!　　　!　!
> !　　　　!　　　!　!　　　!　!　　　!　　　!　　　!　!

Obj number	Hw object type	Log object type	No. of objects	Comment
-	!	!	-	
-	!		-	
-!	!	!	-	
-	!		-	

Obj number	Hw object type	Log object type	No. of objects	Comment
-				
				!CanTransmit()!
				CanMsgTransmit()!
-			-	CanTransmit()!
-			-	

! - ! ! !

© !   !   !   !   !   ! !

⚠ **Caution**

! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! - ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! !
! !

3.2   Acceptance Filters

! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! !
!

⚠ **Caution**

! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! !
! ! ! !

4 [#hw_sleep] - SleepMode and WakeUp

! ! ! ! ! ! ! ! ! !CanSleep()! !
! ! ! ! ! ! !! ! ! !CanWakeUp()! !
! ! ! ! ! ! ! ! ! ! ! ! !

! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! !CanSleep()!
!CanWakeUp()! ! ! ! !kCanNotSupported

4.1 Sleep

! !CanSleep()! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! ! !
CanResetBusSleep()

! ! ! ! ! ! ! ! ! ! ! ! ! he
return value `kCanOk` is always expected. However, if the function returns `kCanFailed`
call `CanSleep()`! ! ! - ! !

4.2 Internal Wakeup

! !CanWakeUp()! ! ! ! ! ! ! ! ! ! !
!

! ! !kCanOk! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! ! !
! !kCanLoopEnterOperationMode! ! ! ! ! ! ! !
! !kCanFailed! ! ! ! ! ! ! !CanWakeup()!
! ! - ! !

4.3 External Wakeup

! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! ! !!
! ! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! !

! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! !ApplCanPreWakeUp()! ! ! ! !
! ! ! ! ! ! ! ! ! !ApplCanWakeUp()!

Caution

!   !   !   ! !   ! ! !   !   !   !   !   !
   !   !   !   !   ! !   !   !   !   !   ! ! ! ! !   !
      !   !   ! !   !   ! !   !   ! !   !   ! !   ! !
              !   ! !   !   ! !   !   !   ! !   ! !   ! !
    !   !   !   !   !   !   !   !   !   !   ! !

   !   !   ! !   !   ! !   !   !   !   ! !   !   !
          ! !   !   !   !   !   ! ! !   !   ! ! !
      !   !   ! !   !   !   !   !   ! ! !   !
      !   !   ! !

   !   !   !   !   !   ! ! !   ! ! !   !   !
     ! ! !   ! !   !   ! ! !   !   !   !   !   !   !
        !   ! !   !   !   !   !   ! !   ! ! !   !

```
#define C_ENABLE_EXTERNAL_WAKEUP_SUPPRESSION
```

Caution

    !   !   !   !   !   !   !   !   !   !   ! !   !
      ! !   !   ! !   ! !   !   !   !   ! !
      ! !   !   !   !   !   ! !   !   !   !
          !   !   !   !   !   !   !   !   !   !
   !   !   ! !   !   !   !   !   !   !   !   !   !   !   !
      !   !   ! !   ! ! ! !   !   !   ! ! !   !
          !   ! ! !   !   !   !   !   !   !   !   ! ! !
   !   ! !   !

5 [#hw_loop] - Hardware Loop Check

⚠️ **Caution**

kCanLoopRamInit

`CanInitPowerOn()`

`CanInitPowerOn()`

kCanLoopInit

`CanInitPowerOn()`

-
-
-
-
-

`CanInitPowerOn()`

kCanLoopBusOffRecovery

! ! ! ! ! ! ! !CanInit()! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! !
!
! ! ! ! ! ! ! ! ! ! ! !
! ! ! - ! ! ! ! ! ! ! !!
! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! !

kCanLoopEnterResetMode

! ! ! ! ! ! ! !CanInit()! ! ! ! ! ! !
! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! !CanInitPowerOn()

kCanLoopEnterOperationMode

! ! ! ! ! ! ! ! CanInit()!CanStart()! !
CanWakeUp()! ! ! ! ! ! ! ! ! ! ! ! !
! !
! ! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! !CanStart()! !CanWakeUp()! ! !
! !

kCanLoopRxFcProcess

! ! ! ! ! ! ! !CanFullCanMsgReceived()! ! !
! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! !!
! ! ! ! !! ! ! ! ! ! ! ! ! ! !
!CanFullCanMsgReceived()! ! ! ! ! !!

6 [#hw_busoff] - Bus off

!   ! ! !   !   !   !   !   !   ! !   !   ! !   !
        !   !   ! ! !   !   ! !   ! !   - !   !
        !   ! !   !   !   ! !   ! !   !   !
    !   !   ! ! !   !CanResetBusOffStart/-End()!   !   ! ! !
! !CanInit()!   !   !   ! !CanResetBusOffEnd()! !   !   ! !
        !   ! !   !   !   ! !   !kCanLoopBusOffRecovery! !
    !   !CanInit()!   !   !

7 CAN Driver Features

7.1   [#hw_feature] - Feature List

	Standard	HighEnd
Initialization		
Power-On Initialization	■	■
Re-Initialization	■	■
Transmission		
Transmit Request	■	■
Transmit Request Queue	■	■
Internal data copy mechanism	■	■
Pretransmit functions	■	■
Common confirmation function	■	■
Confirmation flag	■	■
Confirmation function	■	■
Offline Mode	■	■
Partial Offline Mode	■	■
Passive-Mode	■	■
Tx Observe mode	■	■
Dynamic TxObjects ID	■	■
DLC	■	■
Data-Ptr	■	■
Full CAN Tx Objects	■	■
Cancellation in Hardware	■	■
Low Level Message Transmit	-	■
Reception		
Receive function	■	■
Search algorithms Linear	■	■
Table	-	-
Index	■	■
Hash	■	■
Range specific precopy functions (min. 2, typ.4)	4	4
DLC check	■	■
Internal data copy mechanism	■	■
Generic precopy function	■	■
Precopy function	■	■
Indication flag	■	■
Indication function	■	■
Message not matched function	■	■

Overrun Notification	■	■
Full CAN overrun notification	-	-
Multiple Basic CAN	-	■
Rx Queue [2]	-	■
Bus off		
Notification function	■	■
Nested Recovery functions	■	■
Sleep Mode		
Mode Change	■	■
Preparation	■	■
Notification function	■	■
Special Features		
Status	■	■
Security Level	■	■
Assertions	■	■
Hardware loop check	■	■
Stop Mode	■	■
Support of OSEK operating system	■	■
Polling Mode Tx		

7.2 Description of Hardware-related Features

7.2.1 [#hw_status] - Status

! ! ! ! ! ! ! ! ! ! !

Status	Support
CanHwIsOk(state)	▪
CanHwIsWarning(state)	▪
CanHwIsPassive(state)	▪
CanHwIsBusOff(state)	▪
CanHwIsWakeup(state)	▪
CanHwIsSleep(state)	▪
CanHwIsStart(state)	▪
CanHwIsStop(state)	▪
CanIsOnline(state)	▪
CanIsOffline(state)	▪

! - ! !

7.2.2 [#hw_stop] - Stop Mode

The service function `CanStop()` calls `CanInit()` and then puts the CAN Controller into channel reset mode, where it is disconnected from the bus. If the function is called during CAN communication, the reception or transmission is terminated before it is completed. This mode can be left by calling `CanStart()`.　!　!!!　!!　　! ! ! ! ! ! ! he return value `kCanOk` is always expected. However, if the functions return `kCanFailed` (e.g. caused by a hardware loop cancellation) call `CanStop()`, respectively `CanStart()` again or　-　! !

7.2.3 [#hw_int] - Control of CAN Interrupts

! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! !

! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! !

! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! !

Caution
!　!　　!　!　!　　　　　!**exclusive**!　!　!　!　!　　!　!
!　!　!　!　　　!　!　!　　!　!　!　!　　　　!　!
　　!　!　!　!　!　　　!　!　　!　!　!

7.2.4 [#hw_cancel] - Cancel in Hardware

	Yes	No
!　!CanTxTask()!　!　!　　!　!　!　　　　!　!　　!　!　　　!　　!　　!　!　!		■

!　　　　!　!　　　!　!CanCancelTransmit()!
CanCancelMessageTransmit()

	Yes	No
ApplCanTxConfirmation()!　!　!　　!　!　　　!　　　　!　　　! 　!　　!　!　!　　!　!　　!　!　!　!!　!　!　　! 　!　!　　!　!　　!　!!　!　　　!　!　!　!	■	

7.2.5 Remote Frames
!　　!　!　!　!　!　　!　!　!　　　　　!　　!　!　!　!
!　!　!　　　!

7.2.6 CAN RAM Check
!　!　!　　!　!　!　!　!　!　　!　　!　!　　　!　　　　!
!　!　!　　　!CanInit()!　!　　!　!　!　　!　　!　!　!　!
!　!　　!　!　!!　　!　!　　!　　!　!　　!　!　!　!　!
!　　!　　!　　!　!　!　　!　!　!　!　!　　!　!　　!
!　!　　!　　!　!　!　　!ApplCanCorruptMailbox()!　!　　　!
　!　!　!　!　　　!　!　　!　!

!　!　!　!　　　!　!　!　!　　!　!　!　!　!　!
ApplCanMemCheckFailed()!　!　!　!　!　　!　　!　!　!　!　　　!
　!　　!　　!　!　!　!　!　　!　　　!　!　!　!　!!　!
!　　!　!　!　!　　!　!　!　Hardware-　!　　!　!　!　　!　!　　!　!
　!　!　　　　!　!　!　　!　!　!　!　　　!　!　!
!　!　!　!CanInitPowerOn()

Caution
!　!　　　!　　　!　!　!　　!　!　　!　　!　!　!　!　!
!　!　　!　!　　!　　!　　!　!　　　!　　!!　!
　!　　!　!!　!　!!　　　!　!　!　　!　!　!　!

! ! - ! !ApplCanCorruptMailbox()! ! ! ! ! !
! ! ! ! !-! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! !

Switch	Value	Description
		! ! !ApplCanCorruptMailbox()! ! ! ! ! ! ! ! !! !

7.2.7 Extended CAN RAM Check

! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! !

! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! !
CanInitPowerOn()! ! ! ! ! ! ! ! !! - ! !
! ! ! ! ! ! ! ! ! ! ! ! ! !
! !

! ! ! ! ! ! ! ! ! ! ! !
CanInit()! ! ! ! ! ! ! ! ! !! !
! ! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! !! ! !
! ! ! ! ! ! - ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! - ! ! !
! ! ! ! !! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! - !ApplCanMemCheckFailed() !
!

! ! ! ! ! ! ! ! ! ! ! !

- ! ! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! !
ApplCanGlobalMemCheckFailed()! ! ! !CanInitPowerOn()! ! !
! !kCanEnableCommunication! ! ! ! ! !
! ! ! ! !!! !kCanDisableCommunication! ! ! ! !
! ! ! ! ! CanInitPowerOn()! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! !
CanInitPowerOn()! ! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! !CanInitPowerOn()! !
! ! ! ! ! ! ! ! ! !

- ! ! ! ! ! ! ! ! ! ! ! !
ApplCanCorruptRegisters()! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! ! - !
!ApplCanMemCheckFailed()! ! ! ! ! ! !
! ! !

- ! ! ! ! ! ! ! ! ! ! ! ! ! - ! !
  ApplCanCorruptMailbox() ! ! ! ! ! ! ! ! ! !
  ! ! ! ! ! ! ! ! ! ! !
  ! ! ! ! ! ! ! ! ! ! ! ! !
  ! ! ! ! ! ! ! ! ! ! ! ! !
  !ApplCanMemCheckFailed(),! ! ! ! ! !

- ! ! ! ! ! ! ! ! ! ! ! - ! !
  ! !CanEnableChannelCommunication()

- ! ! ! ! ! ! ! ! ! ! - ! ! ! !
  CanEnableMailboxCommunication()! ! ! ! ! ! ! ! !
  ! !

- ! ! ! ! ! ! ! ! ! ! ! - ! !
  ! !CanInit()! ! ! !CanResetBusOffEnd()! ! ! !
  ! ! ! ! !

> ⚠️ **Caution**
> ! ! ! ! - ! ! ! ! ! ! ! ! ! ! !
> CanEnableChannelCommunication()!CanEnableMailboxCommunication()!
> !CanInitPowerOn()

! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! !! ! ! ! ! ! ! ! ! !
! ! ! ! ! !

Switch	Value	Description
		! ! ! ! ! !

7.2.8 RSCAN ECC Configuration

! ! ! ! ! ! ! ! ! ! ! ! ! ! !
! - ! !ApplCanEccConfiguration()! ! ! ! ! ! !
! !CanInitPowerOn()! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! !-! ! ! ! ! ! ! ! ! !

! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! ! !

Switch	Value	Description
		! ! !ApplCanEccConfiguration()

7.2.9 RSCAN RAM Test

CanInitPowerOn()

ApplCanGlobalMemCheckFailed()

kCanEnableCommunication

kCanDisableCommunication

CanInitPowerOn()

CanInitPowerOn()

8   [#hw_assert] – Assertions

！　　！　　　　　！　！　　　！　　　！　！　　　　！　　！

9 API

9.1 Category

Single Receive Channels (SRC)	
◾	! ! ! ! ! ! ! ! !
Multiple Receive Channel (MRC)	
◾	! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !CanOnline()! ! ! CanOnline(channel)! ! ! ! ! ! ! ! ! ! !CanTransmit(txHandle)

! - ! !

9.2 RSCAN ECC Configuration

! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! !

ApplCanEccConfiguration

Prototype			
! !	void		(void)
! !	void		(void)
Parameter			
–	–		
Return code			
–	–		
Functional Description			
! ! ! ! !CanInitPowerOn()! ! ! ! ! ! ! ! ! ! !			
Particularities and Limitations			
! ! !C_ENABLE_ECC_CALLOUT! !			

9.3 (Extended) CAN RAM Check

! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! !

ApplCanMemCheckFailed

Prototype		
! !	vuint8	(void)
! !	vuint8	(CanChannelHandle channel)
Parameter		
CanChannelHandle channel	! ! ! ! ! ! ! ! ! ! ! !	
Return code		

vuint8

ApplCanCorruptMailbox

Prototype	
! !	void (CanObjectHandle hwObjHandle)
! !	void (CanChannelHandle channel, CanObjectHandle hwObjHandle)
Parameter	
CanChannelHandle channel	! ! ! ! ! ! ! ! ! ! ! !
CanObjectHandle hwObjHandle	! ! ! ! ! ! ! !
Return code	
–	–
Functional Description	
! ! ! ! !CanInit()! ! ! ! ! ! ! ! !
Particularities and Limitations	
Call context !	
Configuration ! ! ! ! ! !	

! !

ApplCanCorruptRegisters

Prototype	
! !	void (void)
! !	void (CanChannelHandle channel)
Parameter	
CanChannelHandle channel	! ! ! ! ! ! ! ! ! ! !
Return code	
–	–
Functional Description	
! ! ! ! ! ! ! ! ! ! ! ! ! ! !	
Particularities and Limitations	
Call context ! ! ! ! ! ! ! ! !CanInit()! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !CanInit() ! ! ! ! ! ! CanResetBusOffEnd() ! ! ! !CanEnableChannelCommunication()	
Configuration ! ! ! ! ! !	

ApplCanGlobalMemCheckFailed

Prototype		
! !	vuint8	(void)
! !	vuint8	(void)
Parameter		
–	–	
Return code		
vuint8	! ! ! !	
	! ! ! !	

Functional Description

! - !
! !
! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! !

Particularities and Limitations

Call context ! ! ! ! ! ! ! ! !CanInitPowerOn()
Configuration ! ! ! ! ! ! !

Important note: ! ! ! ! ! ! !!kCanEnableCommunication! ! ! !
! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! !

! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !
! !

CanEnableChannelCommunication

Prototype	
! !	void (vuint8 suppressRamCheck)
! !	void (CanChannelHandle channel, vuint8 suppressRamCheck)
Parameter	
CanChannelHandle channel	! ! ! ! ! ! ! ! ! -
vuint8 suppressRamCheck	! ! ! ! ! ! - ! ! ! ! ! ! ! ! ! ! ! - ! ! ! ! !
Return code	
–	–
Functional Description	
! ! - ! ! ! ! ! ! ! ! ! ! `!CanInit()!` ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! - ! ! ! ! ! !	
Particularities and Limitations	
Restriction ! ! ! ! ! ! ! `!CanInit()!` **Call context** ! ! ! ! ! ! ! **Configuration** ! ! ! ! ! ! !	

CanEnableMailboxCommunication

Prototype	
! !	`vuint8` (CanObjectHandle `hwObjHandle)`
! !	`vuint8` (CanChannelHandle channel, CanObjectHandle hwObjHandle)`
Parameter	
CanChannelHandle channel	! ! ! ! ! ! ! ! ! ! ! -
CanObjectHandle hwObjHandle	! ! ! ! ! ! ! -
Return code	
vuint8	! ! ! ! ! - ! ! ! ! ! ! !hwObjHandle! !
Functional Description	
	! ! - !-! ! ! ! ! ! ! ! !
Particularities and Limitations	
Call context ! ! ! ! ! ! ! **Configuration** ! ! ! ! ! ! !	

9.4 CAN Interrupt Handling by Application

! ! - ! ! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! !

Prototype	
! !	void (void)
! !	void (CanChannelHandle channel)
Parameter	
CanChannelHandle channel	! ! ! ! ! ! ! ! ! ! ! !
Return code	
–	–
Functional Description	
! ! ! ! !CanCanInterruptDisable()	
Particularities and Limitations	
! ! !C_ENABLE_OSEK_CAN_INTCTRL! !	

Prototype	
! !	void (void)
! !	void (CanChannelHandle channel)
Parameter	
CanChannelHandle channel	! ! ! ! ! ! ! ! ! ! ! !
Return code	
–	–
Functional Description	
! ! ! ! !CanCanInterruptRestore()	
Particularities and Limitations	
! ! !C_ENABLE_OSEK_CAN_INTCTRL! !	

ApplCanWakeupInterruptDisable

Prototype	
! !	void (vuint8 channel)
! !	void (vuint8 channel)
Parameter	
vuint8 channel	! ! ! ! ! ! ! ! ! ! ! ! !
Return code	
–	–
Functional Description	
! ! ! ! !CanWakeup()! !CanInit()	
Particularities and Limitations	
! ! !C_ENABLE_OSEK_CAN_INTCTRL! !C_ENABLE_SLEEP_WAKEUP! ! ! ! ! ! !	

ApplCanWakeupInterruptEnable

Prototype	
! !	void (vuint8 channel)
! !	void (vuint8 channel)
Parameter	
vuint8 channel	! ! ! ! ! ! ! ! ! ! ! ! !
Return code	
–	–
Functional Description	
! ! ! ! !CanSleep()	
Particularities and Limitations	
! ! !C_ENABLE_OSEK_CAN_INTCTRL! !C_ENABLE_SLEEP_WAKEUP! ! ! ! ! ! !	

10 Implementations Hints

10.1 Important Notes

1.   !   !   !   !   !   !   !   !   !   !   !   !   !
         !   !   !   !   ! !   !   ! !   !   !   !   !
      !   !   !   !   !   ! !CanSetPassive!   !

   >   !   !   !CanTransmit!   !   -

   >   !   !CanSetPassive!   !   !   !   !

2.   ! !   !   ! !   !   !   ! !   ! !   !   !CanOffline()!
      !   ! !   !   !   ! !   !   -   !   !   !   !   ! !   !
         ! !   !   !   !   !CanInit()!   !   !CanOffline()!   !
      ! ! ! !   !   ! !   !   !   ! ! !   !   !CanInit()! !
      !   ! !   !

3. !   !   !   -   -   ! !   !   !   !   !   !   ! !   !   !
      !   !   ! !   !   ! !   !   !   !   !   !   !   ! !
      !   !   !   !   !   !

4.   !   !   !   !   !   ! !   ! !   -   -   !   !
      ! !   !   !   !CanGlobalInterruptDisable()!   !
   CanGlobalInterruptRestore()!   !   !   ! !   !   ! !   !   !
         !   ! !   !   ! !

5.   !   !   !   !   !   !   !   !   !   !   !   !   ! !   !
      !   !   !   !   !   !   ! !CanInitPowerOn()!   !   !   !
         !   !   !   ! !   !   !   !   !   ! !   !
         !   !   ! !   !   !   !   !   !   !   ! ! !
         !   !   !   !   !   ! !   !

6.

10.2 Interrupt Configuration

CanInitPoweron()

CAN interrupt request name	CAN interrupt request cause	Provided service routine	Remarks
		CanIsrRxFifo	Used for BasicCAN reception if ! B ! ! ! ! ! ! !
		CanIsrGlobalStatus	Used for Rx BasicCAN overrun if ! ! ! !
		CanIsrTx_n	Used for transmission on physical channel n if ! !is not enabled ! ! ! !
		-	This interrupt is not used.
		CanIsrStatus_n	Used for BusOff detection on physical channel n if Error Polling is not enabled.
		CanIsrWakeup_n	Used for wakeup detection on physical channel n if the sleep/wakeup functionality is enabled, the external ! ! ! ! ! ! ! not enabled.

C_ENABLE_DIRECT_INTERRUPT_BRANCH

```
#pragma ghs interrupt
void CommonIsr_Prio_x ( void )
{
  /* handling for other interrupts that are assigned to
     this priority and not handled by table reference   */

  /* CAN interrupts */
  if (MK bit of INTRCANGRECC == 0) CanIsrRxFifo();
  if (MK bit of INTRCANGERR == 0) CanIsrGlobalStatus();
  if (MK bit of INTRCAN1ERR == 0) CanIsrStatus_1();
  if (MK bit of INTRCAN1TRX == 0) CanIsrTx_1();
  if (MK bit of INTRCAN4ERR == 0) CanIsrStatus_4();
  if (MK bit of INTRCAN4TRX == 0) CanIsrTx_4();

  /* handling for other interrupts that are assigned to
     this priority and not handled by table reference   */
}
```

!C_ENABLE_ISRVOID!

> **Caution**
>
> !C_ENABLE_DIRECT_INTERRUPT_BRANCH!

10.2.1 Configuration of Interrupt Vectors with IAR compiler

```
#define C_CANISRRXFIFO_VECTOR        15
#define C_CANISRGLOBALSTATUS_VECTOR  14
#define C_CANISRTX_VECTOR_2          211
#define C_CANISRSTATUS_VECTOR_2      209
#define C_CANISRWAKEUP_VECTOR_2      31
#define C_CANISRTX_VECTOR_5          281
#define C_CANISRSTATUS_VECTOR_5      279
#define C_CANISRWAKEUP_VECTOR_5      36
```

Caution

! ! !   ! ! !   !   !   ! ! !   !   ! !
      !   !   ! ! !   !   !   ! ! !   !   ! !
  ! !   !

10.3   CAN Interrupt Handling by Application

! !   !   !   ! ! !   !   ! !   !   !   !   !   ! !
 !   ! ! ! !   !   ! !   !   !   ! !   !   !   !   !
    !C_ENABLE_OSEK_CAN_INTCTRL has to be defined via the user configuration file!
 !   !   !   !   !   ! ! !   ! !   ! ! !   !   ! ! ! ! !
      ! ! !   !   !   !   !   !   ! ! !   !   !

  !   ! ! !   ! ! ! ! !   !   ! ! !   !
    ! !   ! !CanInitPowerOn() !   !   ! !   ! !   !   !   !
    ! !   ! - ! ! ! !   !   !   !   !   !   !   !
 ! !   !

 !   ! ! !   !   !   !   ! !   !   !   !
  -   ! ! ! !   ! !   ! CanCanInterruptDisable()
CanCanInterruptRestore()! !   !   !   ! ! ! !   !   ! !
    !OsCanCanInterruptDisable()!   !   ! !   !   !   !   ! !
 ! !   !   !   !   ! ! !   ! ! !   !   !   ! ! ! !
  ! ! ! ! !   ! !   ! !   !OsCanCanInterruptRestore()!   ! !
   ! !   !   !   ! ! ! !   !   !   !   !   ! !
   !   ! ! !   !   !   ! ! !   !   ! ! !
     !   !   ! ! !   !   ! ! ! ! !   !   ! !   ! !
OsCanCanInterruptDisable() and be available not until the last nested call of
OsCanCanInterruptRestore().   ! !   ! ! ! !   !   ! ! !
   !   ! ! ! !   !   !   ! ! ! ! ! ! ! ! !   !
   ! !   !   !   ! ! ! ! !

 !   !   ! !   ! ! !   !C_ENABLE_OSEK_CAN_INTCTRL
is defined   ! ! ! !   !   !   !   ! ! ! !   !   !
     ! !   !   ! ! !   ! !   ! lso there are two additional
application call-back functions.!   !   ! ! ! !   ! ! !   ! !
   ! !   !   ! ! !   !   !   ! ! ! ! !   !
     ! ! !   !

`!ApplCanWakeupInterruptEnable()`! !   !   ! !   ! !`CanSleep()`!

`OsCanCanInterruptDisable()`

!`ApplCanWakeupInterruptDisable()` is only invoked if wakeup processing is performed by interrupts in context of `CanWakeup()`, respectively the external wakeup handling and in `CanInit()`. This function has to disable   !   !-!   ! ! !

> **⚠ Caution**
>
> !`C_ENABLE_OSEK_CAN_INTCTRL` !   ! !   !   ! !   ! !
>
> ! !`OsCanCanInterruptDisable()` and `OsCanCanInterruptRestore()`.   ! !   ! ! !   ! ! !

11 Configuration

11.1 Configuration by GENy

! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! !

> **Note**
>
> ! ! ! ! ! ! ! ! ! ! ! ! !

11.1.1 Platform Settings

Setup Dialog

Preconfiguration		OK
Micro	Hw_Rh850Cpu	Cancel
Derivative	F1L	
Compiler	GreenHills	

! - ! !

Attribute	Supported Values	Description
		! ! - ! ! !
		! ! !
	! ! -	! ! ! !
	! ! -	! ! !

! - ! ! !

11.1.2 Component Settings

!　-　　　!　　　　!　　　!

Attribute	Supported Values	Description
!　　　!　!　　!	!-!	Enter the maximum number of physical CAN channels that are supported by the actually used derivative. This value is independent from the number of channels in the configuration but used to determine the available hardware resources. Only if this value is correct the tool can ensure valid configurations for the actual derivative. Depending on the selected derivative not all values may be available.
!　　!　　!	!	Enable this attribute to use the external clock input (clk_xincan) as CAN clock　　!　!　!　　　　!　!　　　!　　!　!　　!-!　!　!　!　　　!　　　　　!(This setting directly affects the DCS bit in the global configuration register.)

!　-　!　　　!　　　　!

11.1.3 Channel-specific Settings

! - ! ! !

Caution
! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! ! ‼ ! ⁷ ! !
! ! ! ! ! ! ! ! ! ! ! ! !
! ! ! ! ! ! ! !

Attribute	Supported Values	Description
	!	! ! ! ! ! !
! !	-	! !
!	-	! ! ! ! ! ! ! ! ! ! ! !
!	! ! !	! !Depending on the selected derivative and configuration ! ! ! ! ! ! ! ! ! !11.1.2) n ! ! ! ! !
! !	! !	Select the interrupt priority level of this CAN interrupts that are configured if the driver has interrupt control. Depending on the selected derivative not all levels may be available. See section 7.2.3 and chapter 10 for further information.
! ! !	! !	! !-! !
! !	! !	! Depending on the selected derivative n ! ! ! ! !
! ! !	! !	Select the maximum number of pending receive messages that are processed for each Rx BasicCAN object within one polling cycle respectively one interrupt occurrence. By adjusting this value it can be ensured that high FIFO loads will be evenly processed by the driver. Remaining messages are processed within the next polling cycle respectively the interrupt of the next received message on this channel. Select greater values if overruns occur.
! !	! ! ! ! ! !	! !

! - ! ! ! !

> **Note**
> ! ! ! ! ! ! ! ! ! ! ! ! !
> ! ! ! ! ! ! ! ! ! ! !
> ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !
> ! ! ! ! ! ! ! ! ! ! ! !
> ! ! ! ! ! ! ! ! ! ! ! ! !
> ! ! !

11.1.3.1 Acceptance Filter Configuration

Nr.	Acceptance Filter	Type	Mask	Code
1	X XXXX XXXX XXXX XXXX XXXX XXXX 0100	extended	0xC000000F	0x80000004
2	XX1 X0XX XXX0	standard	0xC0000141	0x00000100
3	XXX X1XX XXX0	standard	0xC0000041	0x00000040
4	XXX XXX1 XX00	standard	0xC0000013	0x00000010

Algorithm options
- Use FullCAN ☑   Quick:
- Minimum irrelevant messages ◉
- Maximum closed filters ○   Intensive:

Statistics

	Count	Rate [1/s]
Messages to receive:	47	470.0
Passing messages:	39	390.0
Irrelevant passing messages:	0	0.0
Full CAN messages:	8	80.0
Known messages:	115	1150.0
Max. search depth:	19	

Messages

ID	Name	Filter
0x4BA	RxMSG000004BA_0	fullCAN
0x500	RxMSG00000500_0	2
0x546	RxMSG00000546_0	3
0x58C	RxMSG0000058C_0	2
0x5D2	RxMSG000005D2_0	3
0x618	RxMSG00000618_0	4
0x65E	RxMSG0000065E_0	3
0x6A4	RxMSG000006A4_0	fullCAN
0x6EA	RxMSG000006EA_0	3
0x730	RxMSG00000730_0	2
0x00000064	RxMSG80000064_0	1
0x01AAAAF4	RxMSG81AAAAF4_0	1
0x03555584	RxMSG83555584_0	1

[OK] [Cancel] [Open filters] [Optimize]

!   -   !   !   !

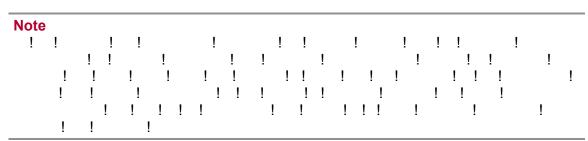| Attribute	Supported Values	Description
!	! ! ! ! !	! ! ! ! ! ! ! - !
	!	! !
! !	!	! ! ! ! ! ! ! ! ! ! ! !
!	-	! ! ! ! ! ! !
	-	! !

!   - !   !   !   !

© !   !   !   !   !   ! !

! !   !   !

11.1.3.1.1 Additional information if the feature "Multiple BasicCAN objects" is used

Nr.	Acceptance Filter	Type	Mask	Code	
1	000 0001 0000	standard	0xE00007FF	0x00000010	BasicCAN 0
2	0 0000 0000 0000 0000 0010 00XX XXX1	extended	0xFFFFFFC1	0x80000201	
3	0 0000 0000 0011 0000 0XXX XXXX XXXX	extended	0xFFFFF800	0x80030000	BasicCAN 1
4	X10 XXX0 XXXX	standard	0xE0000310	0x00000200	
5	XX1 0001 0X01	standard	0xE00001FB	0x00000111	BasicCAN 2
6	XXX 0001 0011	standard	0xE00000FF	0x00000013	

Note

11.1.3.2 Bustiming Configuration

! - ! !

Attribute	Supported Values	Description
Clock	CAN clock	Set the clock frequency that is provided to the CAN cell for baudrate generation Consider the setting of the ! ! ! ! ! ! !
Baudrate	baudrate	Set the baudrate to be used for this channel.
CAN BTR register	register value	Enter the value for the ! ! ! ! ! ! ! ! .
Calculate	-	Calculate possible ! ! !settings out of the entered baudrate or vice versa.
CAN_BTR, Sample, BTL cycles, SJW	-	! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !

! - ! ! !

11.2 Manual Configuration

!   !   !   !   !   ! !   !   !   !   !   ! !
! ! !   !   !

- !`C_DISABLE_NESTED_INTERRUPTS`! !`C_ENABLE_NESTED_INTERRUPTS`! !
  ! !   ! ! !   !   See section 10.1 for further information.

- Define `C_ENABLE_DIRECT_INTERRUPT_BRANCH`!   ! !   !   !
  `C_ENABLE_ISRVOID`! !   !   !   ! ! !   ! !   !   !
          ! !   !   ! !   !   !

- !`C_ENABLE_OSEK_CAN_INTCTRL` to prohibit write accesses by the driver within
  the interrupt controller address space. See section 10.3 for further information.

- Define `C_ENABLE_EXTERNAL_WAKEUP_SUPPRESSION`! !   ! !   !   !
              !   !   ! ! !   !

- See sections 7.2.6 to 7.2.9 for options that control different RAM test features.

- See section 10.2.1 for information on how to set up the interrupt vector table when
  using IAR compiler.

12 Known Issues / Limitations

1. ! ! ! ! ! ! ! ! ! ! ! ! ! !
   ! ! ! ! ! ! ! !

2. ! ! ! ! ! ! ! ! ! ! ! !
   ! ! ! ! ! ! ! !

3. ! ! ! ! ! ! ! ! ! ! ! !
   ! ! ! ! ! ! ! ! ! ! ! ! !
   ! ! ! ! ! !

4. ! ! ! ! ! ! ! ! ! ! !
   ! ! ! ! ! ! ! ! ! !

5. ! ! ! ! ! ! ! ! ! !
   ! ! ! ! ! ! ! ! ! ! ! ! ! !
   ! ! ! ! ! ! ! ! !

6. ! ! ! ! ! ! ! ! ! ! ! ! ! !
   ! ! ! ! ! ! ! ! ! ! ! ! ! !
   ! ! ! ! ! ! ! ! ! ! ! ! ! !
   ! ! ! ! ! ! ! ! ! ! ! !
   ! ! ! ! ! ! ! ! ! ! ! ! ! ! !
   ! ! ! ! ! ! ! ! ! !

7. ! ! ! ! ! ! ! ! ! ! ! ! ! !
   ! ! !

   ! ! ! ! ! ! ! ! ! ! ! ! !
   ! ! !

13 Contact

! ! ! ! ! !

>
>
> !
>
> !
>

_____