**PCIe-Mini-CAN-FD**

**PCIe Mini CAN Controller**

**PCIexpress Mini**

**928-25-001-0210**

**Software MANUAL**

Revision 1.0

September 2020

**ALPHI TECHNOLOGY CORPORATION**

**1898 E. Southern Ave**

**Tempe, AZ 85282 USA**

**Tel : (480) 838-2428**

**Fax: (480) 838-4477**

**NOTICE**

The information in this document has been carefully checked and is believed to be entirely reliable. While all reasonable efforts to ensure accuracy have been taken in the preparation of this manual, ALPHI TECHNOLOGY assumes no responsibility resulting from omissions or errors in this manual, or from the use of information contained herein.

ALPHI TECHNOLOGY reserves the right to make any changes, without notice, to this or any of ALPHI TECHNOLOGY’s products to improve reliability, performance, function or design.

ALPHI TECHNOLOGY does not assume any liability arising out of the application or use of any product or circuit described herein; nor does ALPHI TECHNOLOGY convey any license under its patent rights or the rights of others.

**ALPHI TECHNOLOGY CORPORATION**

**All Rights Reserved**

This document shall not be duplicated, nor its contents used

for any purpose, unless express permission has been granted in advance.

# Table of Contents

Hierarchical Index 2

Class Index 3

File Index 4

Class Documentation 8

AlphiBoard 8

AlteraDma 16

AlteraSpi 18

BoardVersion 22

IrigDecoder::IrigDate 24

IrigDecoder 25

LinearAddress 26

MINIPCIE\_DEV\_CTX 27

MINIPCIE\_INT\_RESULT 28

ParallelInput 30

PcieCra 33

PCIeMini\_CAN\_FD 37

TCAN4550 41

TCAN4x5x\_Device\_Interrupt\_Enable 56

TCAN4x5x\_Device\_Interrupts 61

TCAN4x5x\_MCAN\_CCCR\_Config 67

TCAN4x5x\_MCAN\_Data\_Timing\_Raw 71

TCAN4x5x\_MCAN\_Data\_Timing\_Simple 73

TCAN4x5x\_MCAN\_Interrupt\_Enable 75

TCAN4x5x\_MCAN\_Interrupts 81

TCAN4x5x\_MCAN\_Nominal\_Timing\_Raw 87

TCAN4x5x\_MCAN\_Nominal\_Timing\_Simple 89

TCAN4x5x\_MCAN\_RX\_Header 91

TCAN4x5x\_MCAN\_SID\_Filter 94

TCAN4x5x\_MCAN\_TX\_Header 96

TCAN4x5x\_MCAN\_XID\_Filter 98

TCAN4x5x\_MRAM\_Config 100

TcanInterface 103

TransferDesc 106

File Documentation 108

C:/Alphi/PCIeMiniSoftware/include/AlphiBoard. 108

C:/Alphi/PCIeMiniSoftware/include/AlphiDll.h 111

C:/Alphi/PCIeMiniSoftware/include/AlphiErrorCodes.h 112

C:/Alphi/PCIeMiniSoftware/include/AlteraDma.h 117

C:/Alphi/PCIeMiniSoftware/include/AlteraSpi.h 121

C:/Alphi/PCIeMiniSoftware/include/IrigDecoder.h 123

C:/Alphi/PCIeMiniSoftware/include/ParallelInput.h 124

C:/Alphi/PCIeMiniSoftware/include/PcieCra.h 125

C:/Alphi/PCIeMiniSoftware/include/PCIeMini\_CAN\_FD.h 126

C:/Alphi/PCIeMiniSoftware/include/TCAN4550.h 128

C:/Alphi/PCIeMiniSoftware/include/TCAN4x5x\_Data\_Structs.h 131

C:/Alphi/PCIeMiniSoftware/include/TCAN4x5x\_Reg.h 136

C:/Alphi/PCIeMiniSoftware/include/TCAN4x5x\_SPI.h 164

dllmain.cpp 165

PCIe\_Mini\_CAN\_FD.cpp 166

TCAN4550.cpp 167

TCAN4x5x\_SPI.cpp 168

x64/Debug/PCIeMini\_CAN\_FD\_lib.vcxproj.FileListAbsolute.txt 169

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/AlphiBoard.cpp 170

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/AlphiBoard\_dma.cpp 171

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/AlphiBoard\_irq.cpp 172

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/AlteraDma.cpp 173

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/AlteraSpi.cpp 174

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/PcieCra.cpp 175

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/PCIeMini\_error.cpp 176

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/TestProgram.cpp 177

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/x64/Debug/CodeAnalysisResultManifest.txt 178

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/x64/Debug/PCIeMini\_lib.vcxproj.FileListAbsolute.txt 179

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/x64/Release/PCIeMini\_lib.vcxproj.FileListAbsolute.txt 180

Index 181

# Hierarchical Index

## Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

AlphiBoard

PCIeMini\_CAN\_FD

AlteraDma

AlteraSpi

TcanInterface

BoardVersion

IrigDecoder::IrigDate

IrigDecoder 25

LinearAddress 26

MINIPCIE\_DEV\_CTX 27

MINIPCIE\_INT\_RESULT 28

ParallelInput 30

PcieCra 33

TCAN4550 41

TCAN4x5x\_Device\_Interrupt\_Enable 56

TCAN4x5x\_Device\_Interrupts 61

TCAN4x5x\_MCAN\_CCCR\_Config

TCAN4x5x\_MCAN\_Data\_Timing\_Raw

TCAN4x5x\_MCAN\_Data\_Timing\_Simple

TCAN4x5x\_MCAN\_Interrupt\_Enable

TCAN4x5x\_MCAN\_Interrupts

TCAN4x5x\_MCAN\_Nominal\_Timing\_Raw

TCAN4x5x\_MCAN\_Nominal\_Timing\_Simple

TCAN4x5x\_MCAN\_RX\_Header

TCAN4x5x\_MCAN\_SID\_Filter

TCAN4x5x\_MCAN\_TX\_Header

TCAN4x5x\_MCAN\_XID\_Filter

TCAN4x5x\_MRAM\_Config

TransferDesc

# Class Index

## Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

**AlphiBoard (Base class implementing a PCIe board and the Jungo driver )**

**AlteraDma (Low level SPI interface to the SPI hardware )**

**AlteraSpi (Low level SPI interface to the SPI hardware )**

**BoardVersion (Board Hardware identification and version )**

**IrigDecoder::IrigDate**

**IrigDecoder**

**LinearAddress (Memory Segment Descriptor )**

**MINIPCIE\_DEV\_CTX (Minipcie Device Information Structure )** 27

**MINIPCIE\_INT\_RESULT (Interrupt result information structure )**

**ParallelInput (Alphi Avalon Pio controller class )**

**PcieCra (PCIe CRA module controller class )**

**PCIeMini\_CAN\_FD (PCIeMini\_CAN\_FD controller board object )**

**TCAN4550**

**TCAN4x5x\_Device\_Interrupt\_Enable (Struct containing the device interrupt enable bit field )**

**TCAN4x5x\_Device\_Interrupts (Struct containing the device interrupt bit field )**

**TCAN4x5x\_MCAN\_CCCR\_Config (Struct containing the bit fields of the MCAN CCCR register )**

**TCAN4x5x\_MCAN\_Data\_Timing\_Raw (Used to setup the timing parameters of the MCAN module This is the raw MCAN form of the struct which takes in the same values as the actual Bosch MCAN core )**

**TCAN4x5x\_MCAN\_Data\_Timing\_Simple (Used to setup the data timing parameters of the MCAN module This is a simplified struct, requiring only the prescaler value (1:x), number of time quanta before and after the sample point )**

**TCAN4x5x\_MCAN\_Interrupt\_Enable (Struct containing the MCAN interrupt enable bit field )**

**TCAN4x5x\_MCAN\_Interrupts (Struct containing the MCAN interrupt bit field )**

**TCAN4x5x\_MCAN\_Nominal\_Timing\_Raw (Used to setup the nominal timing parameters of the MCAN module This is the raw MCAN form of the struct which takes in the same values as the actual Bosch MCAN core )**

**TCAN4x5x\_MCAN\_Nominal\_Timing\_Simple (Used to setup the nominal timing parameters of the MCAN module This is a simplified struct, requiring only the prescaler value (1:x), number of time quanta before and after the sample point )**

**TCAN4x5x\_MCAN\_RX\_Header (CAN message header )**

**TCAN4x5x\_MCAN\_SID\_Filter (Standard ID filter struct )**

**TCAN4x5x\_MCAN\_TX\_Header (CAN message header for transmitted messages )**

**TCAN4x5x\_MCAN\_XID\_Filter (Extended ID filter struct )**

**TCAN4x5x\_MRAM\_Config (Defines the number of MRAM elements and the size of the elements )**

**TcanInterface (This class implements the TCAN4550 SPI interface )**

**TransferDesc (Structure containing the details of a DMA transaction )**

# File Index

## File List

Here is a list of all files with brief descriptions:

**C:/Alphi/PCIeMiniSoftware/include/AlphiBoard.h (Base PCIe board class with Jungo driver and Altera PCIe hardware )**

**C:/Alphi/PCIeMiniSoftware/include/AlphiDll.h (Utility DLL definitions )**

**C:/Alphi/PCIeMiniSoftware/include/AlphiErrorCodes.h (Description of the Error Codes used by the libraries )**

**C:/Alphi/PCIeMiniSoftware/include/AlteraDma.h (Description of the low-level access routines to the SPI )**

**C:/Alphi/PCIeMiniSoftware/include/AlteraSpi.h (Description of the low-level access routines to the SPI )**

**C:/Alphi/PCIeMiniSoftware/include/IrigDecoder.h (Irig Decoder class to get time )**

**C:/Alphi/PCIeMiniSoftware/include/ParallelInput.h (Description of the Alphi Parallel Input class )**

**C:/Alphi/PCIeMiniSoftware/include/PcieCra.h (PCIe interface CRA class )**

**C:/Alphi/PCIeMiniSoftware/include/PCIeMini\_CAN\_FD.h (Definitition of the PCIeMini\_CAN\_FD board class )**

**C:/Alphi/PCIeMiniSoftware/include/TCAN4550.h (This file contains TCAN4550 functions )**

**C:/Alphi/PCIeMiniSoftware/include/TCAN4x5x\_Data\_Structs.h (This file contains the TCAN4x5x data structures )**

**C:/Alphi/PCIeMiniSoftware/include/TCAN4x5x\_Reg.h (This file contains the register definitions for the TCAN4x5x Family )**

**C:/Alphi/PCIeMiniSoftware/include/TCAN4x5x\_SPI.h (This file is responsible for abstracting the lower-level microcontroller SPI read and write functions )**

**dllmain.cpp**

**PCIe\_Mini\_CAN\_FD.cpp (Implementation of the PCIeMini\_CAN\_FD board class )**

**TCAN4550.cpp (This file contains TCAN4550 functions, and relies on the TCAN4x5x\_SPI abstraction functions Additional Feature Sets of TCAN4550 vs TCAN4x5x: )**

**TCAN4x5x\_SPI.cpp (This file is responsible for abstracting the lower-level microcontroller SPI read and write functions )**

**C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/AlphiBoard.cpp (Implementation of the base PCIe board class with Jungo driver and Altera PCIe hardware )**

**C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/AlphiBoard\_dma.cpp (Implementation of the DMAs for the base PCIe board class with Jungo driver and Altera PCIe hardware )**

**C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/AlphiBoard\_irq.cpp (Implementation of the interrupts for the base PCIe board class with Jungo driver and Altera PCIe hardware )**

**C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/AlteraDma.cpp (Implementation of the DMA block controller )**

**C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/AlteraSpi.cpp (Implementation of the low-level access routines to the SPI )**

**C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/PcieCra.cpp (PCIe interface CRA class )**

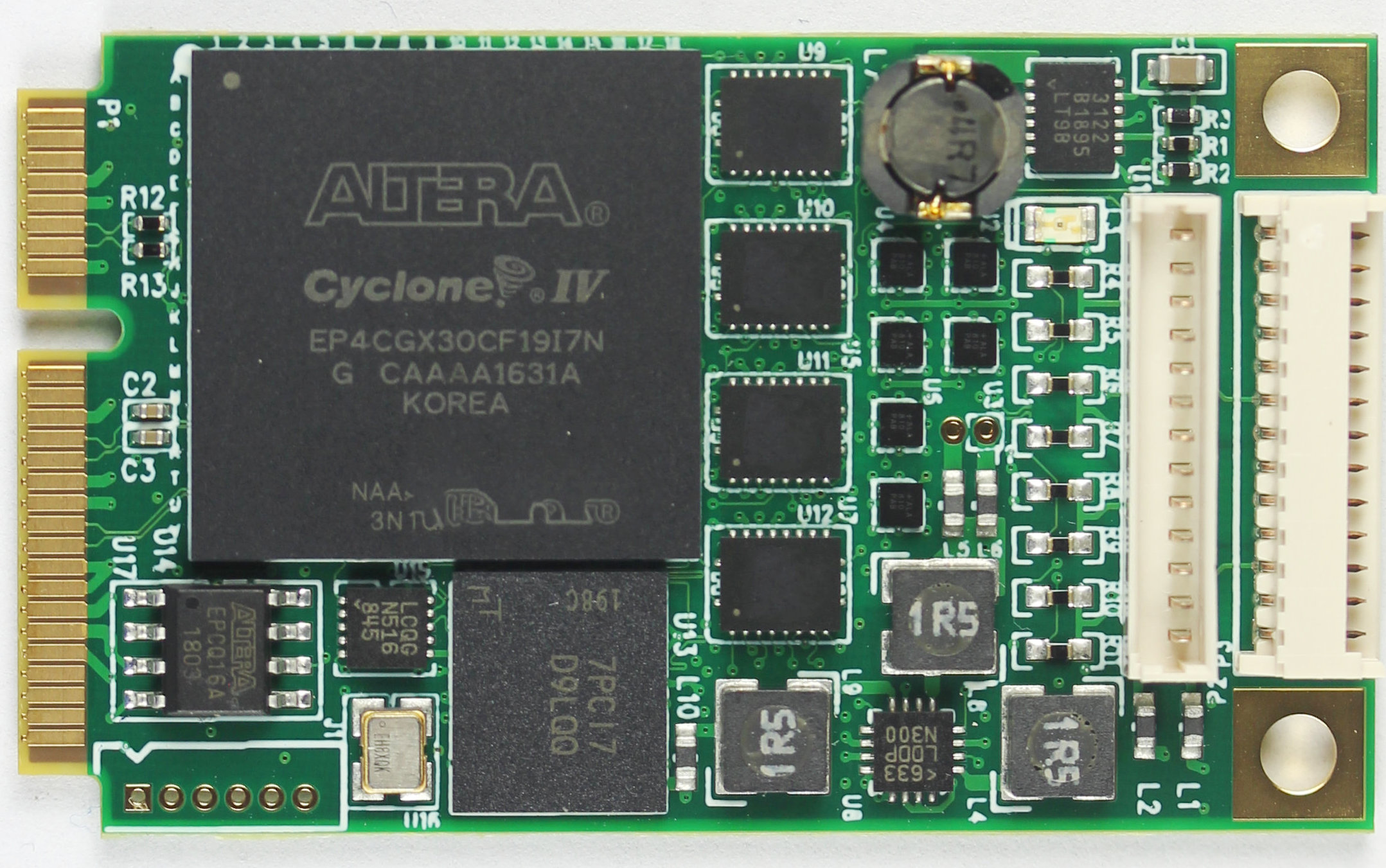
**C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/PCIeMini\_error.cpp (Error code to string conversion )**

**C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/TestProgram.cpp (Utility program class for the test programs )**

**Features**

**CAN Bus**

* Four channel CAN controller



* Supports classic CAN and CAN FD
* Complies with
* ISO11898-1:2015
* M\_CAN Revision 3.2.1.1
* Meets requirements of ISO11898‑2:2016
* Data rates from 40Kbps to 8Mbps
* Software programmable speed selection
* Independent Software controlled
  + 120 Ohm Termination
  + Internal External Power Source
* CAN bus robustness
* ±42V bus fault protection
* Failsafe mode
* Internal dominant state timeout
* Timeout watchdog
* IRIG-B input
* ±15KV ESD protection
* PCI Express x1 interface

**Block Diagram and Operational Overview**

The ***PCIe-Mini-CAN-FD*** is PCIexpress controller with an integrated CAN FD transceiver supporting data rates up to 8 Mbps. The CAN FD controller meets the specifications of the ISO11898-1:2015 high speed controller area network (CAN) data link layer and meets the physical layer requirements of the ISO11898–2:2016 high speed CAN specification.

The ***PCIe-Mini-CAN-FD*** provides an interface between the CAN bus and the system processor that support both classical CAN and CAN FD

The ***PCIe-Mini-CAN-FD*** provides CAN FD transceiver functionality – differential transmit‑receive capability from the bus. It also supports local wake up (LWU) and bus wake up (WUP).

The ***PCIe-Mini-CAN-FD*** device includes many protection features providing device and CAN bus robustness. These features include failsafe mode, internal dominant state timeout, wide bus operating range and a timeout watchdog as examples.



# Class Documentation

## AlphiBoard Class Reference

Base class implementing a PCIe board and the Jungo driver.

#include <AlphiBoard.h>

Inheritance diagram for AlphiBoard:

### Public Member Functions

**AlphiBoard** (UINT16 vendorId, UINT16 deviceId)

**~AlphiBoard** (void)

*Destructor.*

**HRESULT** **Open** (int brdNbr)

*Open a board.*

DWORD **reset** ()

*reset some of the board resources*

uint32\_t **getFpgaID** ()

*Get the FPGA ID of.*

time\_t **getFpgaTimeStamp** ()

*Return the timestamp corresponding to when the FPGA was compiled.*

void **setVerbose** (int **verbose**)

*set the verbose flag*

bool **IsValidDevice** (const CHAR \*sFunc)

*Validate a WDC device handle.*

DWORD **hookInterruptServiceRoutine** (uint32\_t mask, **MINIPCIE\_INT\_HANDLER** uicr, void \*userData)

*Setup the interrupt of the board.*

DWORD **hookInterruptServiceRoutine** (**MINIPCIE\_INT\_HANDLER** uicr)

*Set an interrupt handling routine.*

DWORD **getIntResults** (**MINIPCIE\_INT\_RESULT** \*intResult)

DWORD **unhookInterruptServiceRoutine** ()

*Disable the board interrupt.*

DWORD **enableInterrupts** (uint16\_t mask=0xffff)

*Enable PCIe interrupts.*

DWORD **disableInterrupts** ()

*Disable PCIe interrupts.*

DWORD **Close** ()

*Close a device handle.*

volatile void \* **getBar0Address** (size\_t offset)

*Return a pointer to an object in BAR 0.*

volatile void \* **getBar2Address** (size\_t offset)

*Return a pointer to an object in BAR 2.*

volatile void \* **getBar3Address** (size\_t offset)

*Return a pointer to an object in BAR 3.*

bool **DMARoutine** (DWORD dwDMABufSize, uint32\_t u32LocalAddr, bool fPolling, bool fToDev, **TransferDesc** \*tfrDesc)

**PCIeMini\_status** **DMAOpen** (uint32\_t u32LocalAddr, DWORD dwDMABufSize, bool fToDev, **TransferDesc** \*tfrDesc)

*Allocates and locks a contiguous DMA buffer.*

void **DMAClose** (bool fPolling)

*Frees a previously allocated contiguous DMA buffer.*

void **DMATransfer** (**TransferDesc** \*tfrDesc, bool fPolling)

virtual void **hwDMAStart** (**TransferDesc** \*tfrDesc)

virtual bool **hwDMAWaitForCompletion** (**TransferDesc** \*tfrDesc, bool fPolling)

virtual bool **hwDMAInterruptEnable** (**MINIPCIE\_INT\_HANDLER** **MyDmaIntHandler**, void \*pDMA)

virtual void **hwDMAInterruptDisable** ()

virtual void **hwDMAProgram** (WD\_DMA\_PAGE \*Page, DWORD dwPages, bool fToDev, uint32\_t u32LocalAddr, **TransferDesc** \*tfrDesc)

### Static Public Member Functions

static void **MsSleep** (int ms)

*Millisecond Delay Function.*

### Public Attributes

**LinearAddress** **bar0**

*Memory descriptor for the BAR0 in user memory.*

**LinearAddress** **bar2**

*Memory descriptor for the BAR2 in user memory.*

**LinearAddress** **bar3**

*Memory descriptor for the BAR3 in user memory.*

**PcieCra** \* **cra**

*PCIe Interface instance.*

**BoardVersion** \* **sysid**

*Board identification.*

WD\_DMA \* **pDma**

*Jungo DMA structure.*

int **verbose**

*Flag used by various functions to determine the amount of messages to generate.*

DWORD **libStatus**

*Status returned when trying to open the Jungo library. If it is not WD\_STATUS\_SUCCESS, the initialization failed.*

### Detailed Description

Base class implementing a PCIe board and the Jungo driver.

### Constructor & Destructor Documentation

#### AlphiBoard::AlphiBoard (UINT16 *vendorId*, UINT16 *deviceId*)

#### AlphiBoard::~AlphiBoard (void )

Destructor.

Will close the connection to the board if needed.

### Member Function Documentation

#### DWORD AlphiBoard::Close ()

Close a device handle.

##### Returns

status, a Jungo status code

#### DWORD AlphiBoard::disableInterrupts ()

Disable PCIe interrupts.

Disable the generation of PCIe interrupts by the PCIe interface, and the reception by the Windows driver.

##### Return values

|  |  |
| --- | --- |
| *Status* | code |

#### void AlphiBoard::DMAClose (bool *fPolling*)

Frees a previously allocated contiguous DMA buffer.

#### PCIeMini\_status AlphiBoard::DMAOpen (uint32\_t *u32LocalAddr*, DWORD *dwDMABufSize*, bool *fToDev*, TransferDesc \* *tfrDesc*)

Allocates and locks a contiguous DMA buffer.

##### Parameters

|  |  |
| --- | --- |
| *fToDev* | true means DMA to device, false means DMA from device. |
| *dwDMABufSize* | Size of the DMA buffer allocated in user space. |
| *tfrDesc* | Pointer to a transfer information structure. |
| *u32LocalAddr* | Local FPGA address of the DMA source or destination inside the board. |

#### bool AlphiBoard::DMARoutine (DWORD *dwDMABufSize*, uint32\_t *u32LocalAddr*, bool *fPolling*, bool *fToDev*, TransferDesc \* *tfrDesc*)

#### void AlphiBoard::DMATransfer (TransferDesc \* *tfrDesc*, bool *fPolling*)

#### DWORD AlphiBoard::enableInterrupts (uint16\_t *mask* = 0xffff)

Enable PCIe interrupts.

Enable the generation of PCIe interrupts by the board's PCIe interface. Enable the reception of PCIe interrupts by the Windows driver.

##### Parameters

|  |  |
| --- | --- |
| *mask* | Optional bit map of which local interrupt line is enabled (board dependent.) If not used, default to 0xffff - all local interrupts allowed. |

##### Return values

|  |  |
| --- | --- |
| *Status* | code |

#### volatile void \* AlphiBoard::getBar0Address (size\_t *offset*)

Return a pointer to an object in BAR 0.

##### Parameters

|  |  |
| --- | --- |
| *offset* | Offset in BAR0 |

##### Return values

|  |  |
| --- | --- |
| *Pointer* | to the object |

#### volatile void \* AlphiBoard::getBar2Address (size\_t *offset*)

Return a pointer to an object in BAR 2.

##### Parameters

|  |  |
| --- | --- |
| *offset* | Offset in BAR2 |

##### Return values

|  |  |
| --- | --- |
| *Pointer* | to the object |

#### volatile void \* AlphiBoard::getBar3Address (size\_t *offset*)

Return a pointer to an object in BAR 3.

BAR3 is used on a few board to locate dual-ported RAM.

##### Parameters

|  |  |
| --- | --- |
| *offset* | Offset in BAR3 |

##### Return values

|  |  |
| --- | --- |
| *Pointer* | to the object |

#### uint32\_t AlphiBoard::getFpgaID ()

Get the FPGA ID of.

##### Returns

The FPGA ID.

#### time\_t AlphiBoard::getFpgaTimeStamp ()

Return the timestamp corresponding to when the FPGA was compiled.

##### Returns

a timestamp.

#### DWORD AlphiBoard::getIntResults (MINIPCIE\_INT\_RESULT \* *intResult*)

#### DWORD AlphiBoard::hookInterruptServiceRoutine (MINIPCIE\_INT\_HANDLER *uicr*)

Set an interrupt handling routine.

##### Parameters

|  |  |
| --- | --- |
| *uicr* | user callback routine typedef void (\_\_stdcall \*UsersIntCompletionRoutine)(void \*, uint32\_t); |

##### Returns

ERRCODE\_NO\_ERROR if successful.

#### DWORD AlphiBoard::hookInterruptServiceRoutine (uint32\_t *mask*, MINIPCIE\_INT\_HANDLER *uicr*, void \* *userData*)

Setup the interrupt of the board.

Specify and interrupt service routine and enable the interrupts.

##### Parameters

|  |  |
| --- | --- |
| *mask* | board dependent interrupt mask. |
| *uicr* | pointer to the interrupt service routine. |
| *userData* | Value sent to the interrupt service routine as parameter. |

##### Returns

WD\_STATUS\_SUCCESS when the operation succeeded WD\_INVALID\_PARAMETER if the board is not opened WD\_OPERATION\_FAILED if the board does not have an interrupt resource WD\_OPERATION\_ALREADY\_DONE if there is already an isr active for the interrupt.

#### void AlphiBoard::hwDMAInterruptDisable ()[virtual]

Reimplemented in **PCIeMini\_CAN\_FD** (*p.38*).

#### bool AlphiBoard::hwDMAInterruptEnable (MINIPCIE\_INT\_HANDLER *MyDmaIntHandler*, void \* *pDMA*)[virtual]

Reimplemented in **PCIeMini\_CAN\_FD** (*p.38*).

#### void AlphiBoard::hwDMAProgram (WD\_DMA\_PAGE \* *Page*, DWORD *dwPages*, bool *fToDev*, uint32\_t *u32LocalAddr*, TransferDesc \* *tfrDesc*)[virtual]

Reimplemented in **PCIeMini\_CAN\_FD** (*p.38*).

#### void AlphiBoard::hwDMAStart (TransferDesc \* *tfrDesc*)[virtual]

Reimplemented in **PCIeMini\_CAN\_FD** (*p.38*).

#### bool AlphiBoard::hwDMAWaitForCompletion (TransferDesc \* *tfrDesc*, bool *fPolling*)[virtual]

Reimplemented in **PCIeMini\_CAN\_FD** (*p.39*).

#### bool AlphiBoard::IsValidDevice (const CHAR \* *sFunc*)

Validate a WDC device handle.

##### Parameters

|  |  |
| --- | --- |
| *sFunc* | C-string with name of the function e.g. "IntEnable" |

##### Return values

|  |  |
| --- | --- |
| *true* | if the device context exists. |

#### static void AlphiBoard::MsSleep (int *ms*)[inline], [static]

Millisecond Delay Function.

#### HRESULT AlphiBoard::Open (int *brdNbr*)

Open a board.

Establishes a connection to a board.

##### Parameters

|  |  |
| --- | --- |
| *brdNbr* | the board index to open. |

##### Returns

WD\_DEVICE\_NOT\_FOUND if there is no board corresponding to the number

#### DWORD AlphiBoard::reset ()[inline]

reset some of the board resources

#### void AlphiBoard::setVerbose (int *vb*)

set the verbose flag

The verbose value is used to send more information to the log file or console. It is only partially implemented.

##### Parameters

|  |  |
| --- | --- |
| *vb* | Verbosity level. |

#### DWORD AlphiBoard::unhookInterruptServiceRoutine ()

Disable the board interrupt.

##### Parameters

|  |  |
| --- | --- |
| *mask* | board dependent interrupt mask. |
| *uicr* | pointer to the interrupt service routine. |

##### Returns

WD\_STATUS\_SUCCESS when the operation succeeded WD\_INVALID\_PARAMETER if the board is not opened WD\_OPERATION\_FAILED if the board does not have an interrupt resource WD\_OPERATION\_ALREADY\_DONE if there the interrupt is already disabled.

### Member Data Documentation

#### LinearAddress AlphiBoard::bar0

Memory descriptor for the BAR0 in user memory.

#### LinearAddress AlphiBoard::bar2

Memory descriptor for the BAR2 in user memory.

#### LinearAddress AlphiBoard::bar3

Memory descriptor for the BAR3 in user memory.

#### PcieCra\* AlphiBoard::cra

PCIe Interface instance.

#### DWORD AlphiBoard::libStatus

Status returned when trying to open the Jungo library. If it is not WD\_STATUS\_SUCCESS, the initialization failed.

#### WD\_DMA\* AlphiBoard::pDma

Jungo DMA structure.

#### BoardVersion\* AlphiBoard::sysid

Board identification.

#### int AlphiBoard::verbose

Flag used by various functions to determine the amount of messages to generate.

#### The documentation for this class was generated from the following files:

C:/Alphi/PCIeMiniSoftware/include/**AlphiBoard.h**

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/**AlphiBoard.cpp**

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/**AlphiBoard\_dma.cpp**

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/**AlphiBoard\_irq.cpp**

## AlteraDma Class Reference

Low level SPI interface to the SPI hardware.

#include <AlteraDma.h>

### Public Member Functions

**AlteraDma** (volatile void \*dmaAddress)

*Constructor.*

int **prepare** (void \*data, uint32\_t len, **alt\_rxchan\_done** \*done, void \*handle)

int **space** ()

int **send** (const void \*from, uint32\_t len, **alt\_txchan\_done** \*done, void \*handle)

int **tx\_ioctl** (int req, void \*arg)

int **rx\_ioctl** (int req, void \*arg)

int **ioctl** (int req, void \*arg)

void **launch\_bidir** (**TransferDesc** \*t)

void **launch\_txonly** (**TransferDesc** &t)

void **launch\_rxonly** (**TransferDesc** &t)

void **reset** ()

char \* **statusToString** (char \*buffer)

char \* **controlToString** (char \*buffer)

void **print** (const char \*title=0)

void **irq** (void \*context, uint32\_t id)

uint32\_t **getStatus** ()

uint32\_t **getLength** ()

### Detailed Description

Low level SPI interface to the SPI hardware.

### Constructor & Destructor Documentation

#### AlteraDma::AlteraDma (volatile void \* *dmaAddress*)

Constructor.

Initialise and register the transmit and receive channels for a given physical DMA device.

##### Parameters

|  |  |
| --- | --- |
| *dmaAddress* | Address of the DMA controller in user space. |

### Member Function Documentation

#### char\* AlteraDma::controlToString (char \* *buffer*)[inline]

#### uint32\_t AlteraDma::getLength ()[inline]

#### uint32\_t AlteraDma::getStatus ()[inline]

#### int AlteraDma::ioctl (int *req*, void \* *arg*)

#### void AlteraDma::irq (void \* *context*, uint32\_t *id*)

#### void AlteraDma::launch\_bidir (TransferDesc \* *t*)

#### void AlteraDma::launch\_rxonly (TransferDesc & *t*)

#### void AlteraDma::launch\_txonly (TransferDesc & *t*)

#### int AlteraDma::prepare (void \* *data*, uint32\_t *len*, alt\_rxchan\_done \* *done*, void \* *handle*)

#### void AlteraDma::print (const char \* *title* = 0)[inline]

#### void AlteraDma::reset ()[inline]

#### int AlteraDma::rx\_ioctl (int *req*, void \* *arg*)

#### int AlteraDma::send (const void \* *from*, uint32\_t *len*, alt\_txchan\_done \* *done*, void \* *handle*)

#### int AlteraDma::space ()

#### char\* AlteraDma::statusToString (char \* *buffer*)[inline]

#### int AlteraDma::tx\_ioctl (int *req*, void \* *arg*)

#### The documentation for this class was generated from the following files:

C:/Alphi/PCIeMiniSoftware/include/**AlteraDma.h**

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/**AlteraDma.cpp**

## AlteraSpi Class Reference

Low level SPI interface to the SPI hardware.

#include <AlteraSpi.h>

Inheritance diagram for AlteraSpi:

### Public Member Functions

**AlteraSpi** (volatile void \*addr, uint8\_t width=1)

*Constructor.*

int **sendSpiCommand** (uint32\_t slave, uint32\_t write\_length, const uint32\_t \*write\_data, uint32\_t read\_length, uint32\_t \*read\_data, uint32\_t flags)

*Send an SPI command.*

volatile uint32\_t **getRxData** ()

*Get the content of the receive data register.*

volatile uint32\_t **getStatus** ()

void **resetStatus** ()

void **setTxData** (uint32\_t data)

void **setControl** (uint32\_t data)

void **selectSlave** (volatile uint32\_t data)

### Static Public Attributes

static const uint32\_t **status\_ROE\_mask** = 0x0008

*Receive - overrun error.*

static const uint32\_t **status\_TOE\_mask** = 0x0010

*Transmitter-overrun error.*

static const uint32\_t **status\_TMT\_mask** = 0x0020

*Transmitter shift-register empty.*

static const uint32\_t **status\_TRDY\_mask** = 0x0040

*Transmitter ready.*

static const uint32\_t **status\_RRDY\_mask** = 0x0080

*Receiver ready.*

### Protected Attributes

volatile uint32\_t \* **base**

uint8\_t **wordSize**

int **rxData\_index** = 0

int **txData\_index** = 1

int **status\_Index** = 2

int **control\_index** = 3

int **slaveSelect\_index** = 5

### Detailed Description

Low level SPI interface to the SPI hardware.

### Constructor & Destructor Documentation

#### AlteraSpi::AlteraSpi (volatile void \* *addr*, uint8\_t *width* = 1)

Constructor.

Called only when the board is opened.

##### Parameters

|  |  |
| --- | --- |
| *addr* | Pointer to the device in user space. |
| *width* | Word size in byte, default is 1. |

### Member Function Documentation

#### volatile uint32\_t AlteraSpi::getRxData ()[inline]

Get the content of the receive data register.

##### Return values

|  |  |
| --- | --- |
| *Content* | of the receive data register |

#### volatile uint32\_t AlteraSpi::getStatus ()[inline]

#### void AlteraSpi::resetStatus ()[inline]

#### void AlteraSpi::selectSlave (volatile uint32\_t *data*)[inline]

#### int AlteraSpi::sendSpiCommand (uint32\_t *slave*, uint32\_t *write\_length*, const uint32\_t \* *write\_data*, uint32\_t *read\_length*, uint32\_t \* *read\_data*, uint32\_t *flags*)

Send an SPI command.

This is a very simple routine which performs one SPI master transaction. It would be possible to implement a more efficient version using interrupts and sleeping threads but this is probably not worthwhile initially.

##### Parameters

|  |  |
| --- | --- |
| *slave* | Slave number select 0-31 |
| *write\_length* | Number of bytes to send |
| *write\_data* | A pointer to the buffer containing the data to write |
| *read\_length* | Number of bytes to receive |
| *read\_data* | A pointer to the buffer where the received data is going |
| *flags* | A bit mask, only ALT\_AVALON\_SPI\_COMMAND\_TOGGLE\_SS\_N is used. |

##### Return values

|  |  |
| --- | --- |
| *Number* | of bytes read - in SPI read and write are simultaneous so it cannot be 0. |

#### void AlteraSpi::setControl (uint32\_t *data*)[inline]

#### void AlteraSpi::setTxData (uint32\_t *data*)[inline]

### Member Data Documentation

#### volatile uint32\_t\* AlteraSpi::base[protected]

#### int AlteraSpi::control\_index = 3[protected]

#### int AlteraSpi::rxData\_index = 0[protected]

#### int AlteraSpi::slaveSelect\_index = 5[protected]

#### int AlteraSpi::status\_Index = 2[protected]

#### const uint32\_t AlteraSpi::status\_ROE\_mask = 0x0008[static]

Receive - overrun error.

The ROE bit is set to 1 if new data is received while the rxdata register is full(that is, while the RRDY bit is 1).In this case, the new data overwrites the old. Writing to the status register clears the ROE bit to 0.

#### const uint32\_t AlteraSpi::status\_RRDY\_mask = 0x0080[static]

Receiver ready.

The RRDY bit is set to 1 when the rxdata register is full.

#### const uint32\_t AlteraSpi::status\_TMT\_mask = 0x0020[static]

Transmitter shift-register empty.

In master mode, the TMT bit is set to 0 when a transaction is in progress and set to 1 when the shift register is empty.

#### const uint32\_t AlteraSpi::status\_TOE\_mask = 0x0010[static]

Transmitter-overrun error.

The TOE bit is set to 1 if new data is written to the txdata register while it is still full (that is, while the TRDY bit is 0). In this case, the new data is ignored. Writing to the status register clears the TOE bit to 0.

#### const uint32\_t AlteraSpi::status\_TRDY\_mask = 0x0040[static]

Transmitter ready.

The TRDY bit is set to 1 when the txdata register is empty.

#### int AlteraSpi::txData\_index = 1[protected]

#### uint8\_t AlteraSpi::wordSize[protected]

#### The documentation for this class was generated from the following files:

C:/Alphi/PCIeMiniSoftware/include/**AlteraSpi.h**

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/**AlteraSpi.cpp**

## BoardVersion Class Reference

Board Hardware identification and version.

#include <AlphiBoard.h>

### Public Member Functions

**BoardVersion** (volatile uint32\_t \*addr)

*constructor*

uint32\_t **getVersion** ()

*Version, if there is one programmed on the board hardware. Typically 0.*

time\_t **getTimeStamp** ()

*Date when the board firmware was compiled.*

### Detailed Description

Board Hardware identification and version.

### Constructor & Destructor Documentation

#### BoardVersion::BoardVersion (volatile uint32\_t \* *addr*)

constructor

This constructor reads the chip register to initialize the data. It is called by the open and should not be called by the user.

##### Parameters

|  |  |
| --- | --- |
| *addr* | Offset to the sysid controller in the BAR2 address space |

### Member Function Documentation

#### time\_t BoardVersion::getTimeStamp ()

Date when the board firmware was compiled.

Return FPGA time stamp.

Date and time when the board firmware was compiled, it can be used to identify the version of the hardware.

#### uint32\_t BoardVersion::getVersion ()

Version, if there is one programmed on the board hardware. Typically 0.

Return the board type.

#### The documentation for this class was generated from the following files:

C:/Alphi/PCIeMiniSoftware/include/**AlphiBoard.h**

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/**AlphiBoard.cpp**

## IrigDecoder::IrigDate Struct Reference

#include <IrigDecoder.h>

### Public Attributes

int **tm\_sec**

int **tm\_min**

int **tm\_hour**

int **tm\_yday**

int **tm\_year**

### Member Data Documentation

#### int IrigDecoder::IrigDate::tm\_hour

#### int IrigDecoder::IrigDate::tm\_min

#### int IrigDecoder::IrigDate::tm\_sec

#### int IrigDecoder::IrigDate::tm\_yday

#### int IrigDecoder::IrigDate::tm\_year

#### The documentation for this struct was generated from the following file:

C:/Alphi/PCIeMiniSoftware/include/**IrigDecoder.h**

## IrigDecoder Class Reference

#include <IrigDecoder.h>

### Classes

struct **IrigDate**

### Public Member Functions

**IrigDecoder** (volatile void \*addr)

uint32\_t **getTimeRaw** ()

void **getTime** (**IrigDate** \*ttm)

uint32\_t **getDayRaw** ()

void **getDay** (**IrigDate** \*ttm)

uint32\_t **getSecond** ()

void **getIrigDate** (struct tm \*t)

### Constructor & Destructor Documentation

#### IrigDecoder::IrigDecoder (volatile void \* *addr*)[inline]

### Member Function Documentation

#### void IrigDecoder::getDay (IrigDate \* *ttm*)[inline]

#### uint32\_t IrigDecoder::getDayRaw ()[inline]

#### void IrigDecoder::getIrigDate (struct tm \* *t*)[inline]

#### uint32\_t IrigDecoder::getSecond ()[inline]

#### void IrigDecoder::getTime (IrigDate \* *ttm*)[inline]

#### uint32\_t IrigDecoder::getTimeRaw ()[inline]

#### The documentation for this class was generated from the following file:

C:/Alphi/PCIeMiniSoftware/include/**IrigDecoder.h**

## LinearAddress Struct Reference

Memory Segment Descriptor.

#include <AlphiBoard.h>

### Public Attributes

void \* **Address**

*Linear address.*

size\_t **Length**

*Length of the mapping.*

### Detailed Description

Memory Segment Descriptor.

### Member Data Documentation

#### void\* LinearAddress::Address

Linear address.

#### size\_t LinearAddress::Length

Length of the mapping.

#### The documentation for this struct was generated from the following file:

C:/Alphi/PCIeMiniSoftware/include/**AlphiBoard.h**

## MINIPCIE\_DEV\_CTX Struct Reference

Minipcie Device Information Structure.

#include <AlphiBoard.h>

### Public Attributes

**MINIPCIE\_INT\_HANDLER** **funcDiagIntHandler**

*Interrupt handler routine.*

**MINIPCIE\_EVENT\_HANDLER** **funcDiagEventHandler**

*Event handler routine.*

void \* **userData**

*Data passed to the interrupt routine.*

### Detailed Description

Minipcie Device Information Structure.

### Member Data Documentation

#### MINIPCIE\_EVENT\_HANDLER MINIPCIE\_DEV\_CTX::funcDiagEventHandler

Event handler routine.

#### MINIPCIE\_INT\_HANDLER MINIPCIE\_DEV\_CTX::funcDiagIntHandler

Interrupt handler routine.

#### void\* MINIPCIE\_DEV\_CTX::userData

Data passed to the interrupt routine.

#### The documentation for this struct was generated from the following file:

C:/Alphi/PCIeMiniSoftware/include/**AlphiBoard.h**

## MINIPCIE\_INT\_RESULT Struct Reference

Interrupt result information structure.

#include <AlphiBoard.h>

### Public Attributes

DWORD **dwCounter**

*Number of interrupts received.*

DWORD **dwLost**

*Number of interrupts not yet handled.*

WD\_INTERRUPT\_WAIT\_RESULT **waitResult**

*See WD\_INTERRUPT\_WAIT\_RESULT values in windrvr.h.*

DWORD **dwEnabledIntType**

*Interrupt type that was actually enabled (MSI/MSI-X/Level Sensitive/Edge-Triggered)*

DWORD **dwLastMessage**

### Detailed Description

Interrupt result information structure.

### Member Data Documentation

#### DWORD MINIPCIE\_INT\_RESULT::dwCounter

Number of interrupts received.

#### DWORD MINIPCIE\_INT\_RESULT::dwEnabledIntType

Interrupt type that was actually enabled (MSI/MSI-X/Level Sensitive/Edge-Triggered)

#### DWORD MINIPCIE\_INT\_RESULT::dwLastMessage

Message data of the last received MSI/MSI-X (Windows Vista and higher); N/A to line-based interrupts)

#### DWORD MINIPCIE\_INT\_RESULT::dwLost

Number of interrupts not yet handled.

#### WD\_INTERRUPT\_WAIT\_RESULT MINIPCIE\_INT\_RESULT::waitResult

See WD\_INTERRUPT\_WAIT\_RESULT values in windrvr.h.

#### The documentation for this struct was generated from the following file:

C:/Alphi/PCIeMiniSoftware/include/**AlphiBoard.h**

## ParallelInput Class Reference

Alphi Avalon Pio controller class.

#include <ParallelInput.h>

### Public Member Functions

**ParallelInput** (volatile void \*addr)

**PCIeMini\_status** **reset** ()

*Reset the PIO.*

uint32\_t **getData** ()

uint32\_t **getIrqEnable** ()

*Retrieve the interrupt mask.*

uint32\_t **setIrqEnable** (uint32\_t mask)

*Enable bits in the interrupt mask.*

uint32\_t **setIrqDisable** (uint32\_t mask)

*Disable bits in the interrupt mask.*

uint32\_t **getIrqStatus** ()

uint32\_t **resetIrq** ()

**PCIeMini\_status** **clearIrqStatus** (uint32\_t mask)

### Public Attributes

volatile uint32\_t \* **base**

int **data\_index** = 0

int **polarity\_index** = 1

*polarity: when set to 1, the interrupt is requested if the corresponding bit is low.*

int **edgeReg\_Index** = 2

*edge register: when a bit is set to 1, the interrupt is generated on an edge*

int **irqStatus\_index** = 3

int **irqEnable\_index** = 4

int **direction\_index** = 5

int **dataOut\_index** = 6

int **irqDelay\_index** = 7

### Static Public Attributes

static const uint16\_t **CAP\_INPUT** = 0x01

static const uint16\_t **CAP\_OUTPUT** = 0x02

static const uint16\_t **CAP\_INPUT\_OUTPUT** = 0x03

### Detailed Description

Alphi Avalon Pio controller class.

### Constructor & Destructor Documentation

#### ParallelInput::ParallelInput (volatile void \* *addr*)[inline]

### Member Function Documentation

#### PCIeMini\_status ParallelInput::clearIrqStatus (uint32\_t *mask*)[inline]

#### uint32\_t ParallelInput::getData ()[inline]

#### uint32\_t ParallelInput::getIrqEnable ()[inline]

Retrieve the interrupt mask.

Returns 1 for the bits corresponding to input bits able to generate interrupts. On output-only devices, it will return 0.

##### Return values

|  |  |
| --- | --- |
| *A* | 32-bit bit map of which bit can generate interrupts. |

#### uint32\_t ParallelInput::getIrqStatus ()[inline]

#### PCIeMini\_status ParallelInput::reset ()[inline]

Reset the PIO.

Whenever supported, set the direction register to all input, the data register to 0, and disable interrupts.

##### Return values

|  |  |
| --- | --- |
| *Always* | success |

#### uint32\_t ParallelInput::resetIrq ()[inline]

#### uint32\_t ParallelInput::setIrqDisable (uint32\_t *mask*)[inline]

Disable bits in the interrupt mask.

Returns 1 for the bits corresponding to input bits able to generate interrupts. On output-only devices, it will return 0.

##### Parameters

|  |  |
| --- | --- |
| *mask* | a bit mask of which bits to disable |

##### Return values

|  |  |
| --- | --- |
| *A* | 32-bit bit map of which bit can generate interrupts. |

#### uint32\_t ParallelInput::setIrqEnable (uint32\_t *mask*)[inline]

Enable bits in the interrupt mask.

Returns 1 for the bits corresponding to input bits able to generate interrupts. On output-only devices, it will return 0.

##### Parameters

|  |  |
| --- | --- |
| *mask* | a bit mask of which bits to enable |

##### Return values

|  |  |
| --- | --- |
| *A* | 32-bit bit map of which bit can generate interrupts. |

### Member Data Documentation

#### volatile uint32\_t\* ParallelInput::base

#### const uint16\_t ParallelInput::CAP\_INPUT = 0x01[static]

#### const uint16\_t ParallelInput::CAP\_INPUT\_OUTPUT = 0x03[static]

#### const uint16\_t ParallelInput::CAP\_OUTPUT = 0x02[static]

#### int ParallelInput::data\_index = 0

#### int ParallelInput::dataOut\_index = 6

#### int ParallelInput::direction\_index = 5

#### int ParallelInput::edgeReg\_Index = 2

edge register: when a bit is set to 1, the interrupt is generated on an edge

#### int ParallelInput::irqDelay\_index = 7

#### int ParallelInput::irqEnable\_index = 4

#### int ParallelInput::irqStatus\_index = 3

#### int ParallelInput::polarity\_index = 1

polarity: when set to 1, the interrupt is requested if the corresponding bit is low.

#### The documentation for this class was generated from the following file:

C:/Alphi/PCIeMiniSoftware/include/**ParallelInput.h**

## PcieCra Class Reference

PCIe CRA module controller class.

#include <PcieCra.h>

### Public Member Functions

**PcieCra** (volatile void \*cra\_addr)

*constructor*

void **reset** ()

*Reset the CRA PCIe interface.*

uint32\_t **getIrqStatus** ()

*return the interrupt status of the local IRQ lines*

void **setIrqEnableMask** (uint32\_t mask)

*Enable/disable the interrupts.*

uint32\_t **getIrqEnableMask** ()

*return the interrupt enable mask*

**PCIeMini\_status** **setTxsAvlAddress** (uint32\_t txs\_addr, uint64\_t pageSize, uint16\_t nbrOfEntries)

*Set the local Avalon address for the PCIe txs port.*

**PCIeMini\_status** **getMappedAddress** (uint64\_t pcieAddress, int tableEntry, uint32\_t \*localAddress)

*Calculate the DMA address through the txs.*

int **setTrEntry** (int entryNbr, bool is64bitAddress, uint64\_t pcieAddress)

*program an entry in the translation table*

### Detailed Description

PCIe CRA module controller class.

This is a limited software interface to the CRA module of the PCIe adapter that:

allows to enable/disable the interrupt requests to the PCIe bus and check the status of the local interrupt request lines.

allows some DMA to/from the PC

##### Parameters

|  |  |
| --- | --- |
| *txs\_addr* | Address of the txs interface of the PCIe interface chip, in the Avalon address space. It is used to program the DMA controller. |

### Constructor & Destructor Documentation

#### PcieCra::PcieCra (volatile void \* *cra\_addr*)

constructor

This constructor should be only called when the board is opened.

##### Parameters

|  |  |
| --- | --- |
| *cra\_addr* | Address of the CRA in user space. |

### Member Function Documentation

#### uint32\_t PcieCra::getIrqEnableMask ()

return the interrupt enable mask

#### uint32\_t PcieCra::getIrqStatus ()

return the interrupt status of the local IRQ lines

In order for the PCIe interface to request an interrupt on the PCIe bus, the bit needs to be set in this register, and the corresponding bit should be set in the interrupt mask register.

#### PCIeMini\_status PcieCra::getMappedAddress (uint64\_t *pcieAddress*, int *tableEntry*, uint32\_t \* *localAddress*)

Calculate the DMA address through the txs.

##### Parameters

|  |  |
| --- | --- |
| *pcieAddress* | PCIe address of the PC memory |
| *tableEntry* | entry number in the translation table |
| *localAddress* | Address to program in the DMA to access the txs port of the PCIe controller |

#### void PcieCra::reset ()

Reset the CRA PCIe interface.

Disable interrupts.

#### void PcieCra::setIrqEnableMask (uint32\_t *mask*)

Enable/disable the interrupts.

##### Parameters

|  |  |
| --- | --- |
| *mask* | bit mask of enabled interrupts |

#### int PcieCra::setTrEntry (int *entryNbr*, bool *is64bitAddress*, uint64\_t *pcieAddress*)

program an entry in the translation table

##### Parameters

|  |  |
| --- | --- |
| *entryNbr* | Index of the entry to set up. |
| *is64bitAddress* | True if the address of the target location is a 64-bit address |
| *pcieAddress* | Address of the target location |

#### PCIeMini\_status PcieCra::setTxsAvlAddress (uint32\_t *txs\_addr*, uint64\_t *pageSize*, uint16\_t *nbrOfEntries*)

Set the local Avalon address for the PCIe txs port.

For example, if the core is configured with an address translation table with the following attributes :

Number of Address Pages—16

* Size of Address Pages—1 MByte
* PCI Express Address Size—64 bits then the values in Figure 4–12 are :

N = 20 (due to the 1 MByte page size)

* Q = 16 (number of pages)
* M = 24 (20 + 4 bit page selection)
* P = 64 In this case, the Avalon address is interpreted as follows :

Bits[31:24] select the TX slave module port from among other slaves connected to the same master by the system interconnect fabric.The decode is based on the base addresses assigned in Qsys.

* Bits[23:20] select the address translation table entry.
* Bits[63:20] of the address translation table entry become PCI Express address bits [63:20].
* Bits[19:0] are passed throughand become PCI Express address bits[19:0]. The address translation table can be hardwired or dynamically configured at run time.When the IP core is parameterized for dynamic address translation, the address translation table is implemented in memoryand can be accessed through the CRA slave module.This access mode is useful in a typical PCI Express system where address allocation occurs after BIOS initialization.
* Number of Address Pages—2
* Size of Address Pages—16 MByte
* PCI Express Address Size—64 bits then the values in Figure 4–12 are :

N = 24 (due to the 16 MByte page size)

* Q = 2 (number of pages)
* M = 25 (24 + 1 bit page selection)
* P = 64 In this case, the Avalon address is interpreted as follows :

Bits[31:24] select the TX slave module port from among other slaves connected to the same master by the system interconnect fabric.The decode is based on the base addresses assigned in Qsys.

* Bits[24] select the address translation table entry.
* Bits[63:25] of the address translation table entry become PCI Express address bits [63:20].
* Bits[23:0] are passed throughand become PCI Express address bits[19:0]. The address translation table can be hardwired or dynamically configured at run time.When the IP core is parameterized for dynamic address translation, the address translation table is implemented in memoryand can be accessed through the CRA slave module.This access mode is useful in a typical PCI Express system where address allocation occurs after BIOS initialization.

##### Parameters

|  |  |
| --- | --- |
| *txs\_addr* | Local Avalon Address of the txs area |
| *nbrOfEntries* | Number of table entries used to calculate the bit pattern |
| *pageSize* | size of each translation page (this is currently ignored, hard coded) |

#### The documentation for this class was generated from the following files:

C:/Alphi/PCIeMiniSoftware/include/**PcieCra.h**

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/**PcieCra.cpp**

## PCIeMini\_CAN\_FD Class Reference

**PCIeMini\_CAN\_FD** controller board object.

#include <PCIeMini\_CAN\_FD.h>

Inheritance diagram for PCIeMini\_CAN\_FD:

### Public Member Functions

**PCIeMini\_CAN\_FD** ()

**PCIeMini\_status** **open** (int brdNbr)

*Open: connect to an actual board.*

**PCIeMini\_status** **close** ()

*Close the connection to a board object and free the resources.*

**PCIeMini\_status** **reset** ()

*Reset the board controllers.*

void **hwDMAStart** (**TransferDesc** \*tfrDesc)

bool **hwDMAWaitForCompletion** (**TransferDesc** \*tfrDesc, bool fPolling)

bool **hwDMAInterruptEnable** (**MINIPCIE\_INT\_HANDLER** **MyDmaIntHandler**, void \*pDMA)

void **hwDMAInterruptDisable** ()

void **hwDMAProgram** (WD\_DMA\_PAGE \*Page, DWORD dwPages, bool fToDev, uint32\_t u32LocalAddr, **TransferDesc** \*tfrDesc)

*program the local devices (DMA and CRA) for the DMA*

### Public Attributes

**TCAN4550** \* **can** [**nbrOfCanInterfaces**]

CanFdNiosComm \* **canNios**

AlteraPio \* **controlRegister**

*Interface to the board control register.*

AlteraPio \* **ledPio**

*Interface to the board control register.*

**ParallelInput** \* **input0**

**ParallelInput** \* **input1**

**IrigDecoder** \* **irig**

**AlteraDma** \* **dma**

volatile uint32\_t \* **dpr**

volatile uint16\_t \* **mddr**

### Static Public Attributes

static const uint8\_t **nbrOfCanInterfaces** = 4

static const uint32\_t **dpr\_offset** = 0x4000

static const uint32\_t **dpr\_length** = 0x400

### Additional Inherited Members

### Detailed Description

**PCIeMini\_CAN\_FD** controller board object.

### Constructor & Destructor Documentation

#### PCIeMini\_CAN\_FD::PCIeMini\_CAN\_FD ()

The constructor does not take any parameter. The board is not actually usable until the open method connects it to real hardware.

### Member Function Documentation

#### PCIeMini\_status PCIeMini\_CAN\_FD::close ()

Close the connection to a board object and free the resources.

##### Returns

ERRCODE\_NO\_ERROR if successful.

#### void PCIeMini\_CAN\_FD::hwDMAInterruptDisable ()[virtual]

Reimplemented from **AlphiBoard** (*p.13*).

#### bool PCIeMini\_CAN\_FD::hwDMAInterruptEnable (MINIPCIE\_INT\_HANDLER *MyDmaIntHandler*, void \* *pDMA*)[virtual]

Reimplemented from **AlphiBoard** (*p.13*).

#### void PCIeMini\_CAN\_FD::hwDMAProgram (WD\_DMA\_PAGE \* *Page*, DWORD *dwPages*, bool *fToDev*, uint32\_t *u32LocalAddr*, TransferDesc \* *tfrDesc*)[virtual]

program the local devices (DMA and CRA) for the DMA

##### Parameters

|  |  |
| --- | --- |
| *fToDev* | When true DMA to device, when false DMA from device. |

Reimplemented from **AlphiBoard** (*p.13*).

#### void PCIeMini\_CAN\_FD::hwDMAStart (TransferDesc \* *tfrDesc*)[virtual]

Reimplemented from **AlphiBoard** (*p.13*).

#### bool PCIeMini\_CAN\_FD::hwDMAWaitForCompletion (TransferDesc \* *tfrDesc*, bool *fPolling*)[virtual]

Reimplemented from **AlphiBoard** (*p.13*).

#### PCIeMini\_status PCIeMini\_CAN\_FD::open (int *brdNbr*)

Open: connect to an actual board.

##### Parameters

|  |  |
| --- | --- |
| *brdNbr* | The board number is actually system dependent but if you have only one board, it should be 0. |

##### Returns

ERRCODE\_NO\_ERROR if successful.

#### PCIeMini\_status PCIeMini\_CAN\_FD::reset ()

Reset the board controllers.

##### Returns

ERRCODE\_NO\_ERROR if successful.

### Member Data Documentation

#### TCAN4550\* PCIeMini\_CAN\_FD::can[nbrOfCanInterfaces]

#### CanFdNiosComm\* PCIeMini\_CAN\_FD::canNios

#### AlteraPio\* PCIeMini\_CAN\_FD::controlRegister

Interface to the board control register.

#### AlteraDma\* PCIeMini\_CAN\_FD::dma

#### volatile uint32\_t\* PCIeMini\_CAN\_FD::dpr

#### const uint32\_t PCIeMini\_CAN\_FD::dpr\_length = 0x400[static]

#### const uint32\_t PCIeMini\_CAN\_FD::dpr\_offset = 0x4000[static]

#### ParallelInput\* PCIeMini\_CAN\_FD::input0

#### ParallelInput\* PCIeMini\_CAN\_FD::input1

#### IrigDecoder\* PCIeMini\_CAN\_FD::irig

#### AlteraPio\* PCIeMini\_CAN\_FD::ledPio

Interface to the board control register.

#### volatile uint16\_t\* PCIeMini\_CAN\_FD::mddr

#### const uint8\_t PCIeMini\_CAN\_FD::nbrOfCanInterfaces = 4[static]

#### The documentation for this class was generated from the following files:

C:/Alphi/PCIeMiniSoftware/include/**PCIeMini\_CAN\_FD.h**

**PCIe\_Mini\_CAN\_FD.cpp**

## TCAN4550 Class Reference

#include <TCAN4550.h>

### Public Member Functions

**TCAN4550** (volatile void \*addr, AlteraPio \*rstPio, **ParallelInput** \*status0, uint8\_t nbr)

void **reset** ()

void **enableIrq** ()

void **disableIrq** ()

bool **MCAN\_EnableProtectedRegisters** (void)

*Enable Protected MCAN Registers.*

bool **MCAN\_DisableProtectedRegisters** (void)

*Disable Protected MCAN Registers.*

bool **MCAN\_ConfigureCCCRRegister** (**TCAN4x5x\_MCAN\_CCCR\_Config** \*cccr)

*Configure the MCAN CCCR Register.*

void **MCAN\_ReadCCCRRegister** (**TCAN4x5x\_MCAN\_CCCR\_Config** \*cccrConfig)

*Read the MCAN CCCR configuration register.*

void **MCAN\_ReadDataTimingFD\_Simple** (**TCAN4x5x\_MCAN\_Data\_Timing\_Simple** \*dataTiming)

*Reads the MCAN data time settings, using the simple struct.*

void **MCAN\_ReadDataTimingFD\_Raw** (**TCAN4x5x\_MCAN\_Data\_Timing\_Raw** \*dataTiming)

*Reads the MCAN data time settings, using the raw MCAN struct.*

bool **MCAN\_ConfigureDataTiming\_Simple** (**TCAN4x5x\_MCAN\_Data\_Timing\_Simple** \*dataTiming)

*Writes the MCAN data time settings, using the simple data timing struct.*

bool **MCAN\_ConfigureDataTiming\_Raw** (**TCAN4x5x\_MCAN\_Data\_Timing\_Raw** \*dataTiming)

*Writes the MCAN data time settings, using the raw MCAN data timing struct.*

void **MCAN\_ReadNominalTiming\_Simple** (**TCAN4x5x\_MCAN\_Nominal\_Timing\_Simple** \*nomTiming)

*Reads the MCAN nominal/arbitration time settings, using the simple timing struct.*

void **MCAN\_ReadNominalTiming\_Raw** (**TCAN4x5x\_MCAN\_Nominal\_Timing\_Raw** \*nomTiming)

*Reads the MCAN nominal/arbitration time settings, using the raw MCAN timing struct.*

bool **MCAN\_ConfigureNominalTiming\_Simple** (**TCAN4x5x\_MCAN\_Nominal\_Timing\_Simple** \*nomTiming)

*Writes the MCAN nominal timing settings, using the simple nominal timing struct.*

bool **MCAN\_ConfigureNominalTiming\_Raw** (**TCAN4x5x\_MCAN\_Nominal\_Timing\_Raw** \*nomTiming)

*Writes the MCAN nominal timing settings, using the raw MCAN nominal timing struct.*

bool **MRAM\_Configure** (**TCAN4x5x\_MRAM\_Config** \*MRAMConfig)

*Configures the MRAM registers.*

void **MRAM\_Clear** (void)

*Clear (Zero-fill) the contents of MRAM.*

void **MCAN\_ReadInterrupts** (**TCAN4x5x\_MCAN\_Interrupts** \*ir)

*Read the MCAN interrupts.*

void **MCAN\_ClearInterrupts** (**TCAN4x5x\_MCAN\_Interrupts** \*ir)

*Clear the MCAN interrupts.*

void **MCAN\_ClearInterruptsAll** (void)

*Clear all MCAN interrupts.*

void **MCAN\_ReadInterruptEnable** (**TCAN4x5x\_MCAN\_Interrupt\_Enable** \*ie)

*Read the MCAN interrupt enable register.*

void **MCAN\_ConfigureInterruptEnable** (**TCAN4x5x\_MCAN\_Interrupt\_Enable** \*ie)

*Configures the MCAN interrupt enable register.*

uint8\_t **MCAN\_ReadNextFIFO** (**TCAN4x5x\_MCAN\_FIFO\_Enum** FIFODefine, **TCAN4x5x\_MCAN\_RX\_Header** \*header, uint8\_t dataPayload[])

*Read the next MCAN FIFO element.*

uint8\_t **MCAN\_ReadRXBuffer** (uint8\_t bufIndex, **TCAN4x5x\_MCAN\_RX\_Header** \*header, uint8\_t dataPayload[])

*Read the specified RX buffer element.*

uint32\_t **MCAN\_WriteTXBuffer** (uint8\_t bufIndex, **TCAN4x5x\_MCAN\_TX\_Header** \*header, uint8\_t dataPayload[])

*Write CAN message to the specified TX buffer.*

bool **MCAN\_TransmitBufferContents** (uint8\_t bufIndex)

*Transmit TX buffer contents of the specified tx buffer.*

bool **MCAN\_WriteSIDFilter** (uint8\_t filterIndex, **TCAN4x5x\_MCAN\_SID\_Filter** \*filter)

*Write MCAN Standard ID filter into MRAM.*

bool **MCAN\_WriteXIDFilter** (uint8\_t fifoIndex, **TCAN4x5x\_MCAN\_XID\_Filter** \*filter)

*Write MCAN Extended ID filter into MRAM.*

uint8\_t **MCAN\_DLCtoBytes** (uint8\_t inputDLC)

*Converts the CAN message DLC hex value to the number of bytes it corresponds to.*

uint8\_t **MCAN\_TXRXESC\_DataByteValue** (uint8\_t inputESCValue)

*Converts the MCAN ESC (Element Size) value to number of bytes that it corresponds to.*

uint16\_t **Device\_ReadDeviceVersion** (void)

*Read the TCAN4x5x device version register.*

void **Device\_ReadDeviceIdent** (uint32\_t \*id)

*Read the TCAN4x5x device identification.*

void **Device\_ReadInterrupts** (**TCAN4x5x\_Device\_Interrupts** \*ir)

*Read the device interrupts.*

void **Device\_ClearInterrupts** (**TCAN4x5x\_Device\_Interrupts** \*ir)

*Clear the device interrupts.*

void **Device\_ClearInterruptsAll** (void)

*Clear all device interrupts.*

void **Device\_ReadInterruptEnable** (**TCAN4x5x\_Device\_Interrupt\_Enable** \*ie)

*Read the device interrupt enable register.*

bool **Device\_ConfigureInterruptEnable** (**TCAN4x5x\_Device\_Interrupt\_Enable** \*ie)

*Configures the device interrupt enable register.*

bool **Device\_SetMode** (**TCAN4x5x\_Device\_Mode\_Enum** modeDefine)

*Sets the TCAN4x5x device mode.*

**TCAN4x5x\_Device\_Mode\_Enum** **Device\_ReadMode** (void)

*Reads the TCAN4x5x device mode.*

bool **Device\_EnableTestMode** (**TCAN4x5x\_Device\_Test\_Mode\_Enum** modeDefine)

*Sets the TCAN4x5x device test mode.*

bool **Device\_DisableTestMode** (void)

*Disables the TCAN4x5x device test mode.*

**TCAN4x5x\_Device\_Test\_Mode\_Enum** **Device\_ReadTestMode** (void)

*Reads the TCAN4x5x device test mode.*

bool **WDT\_Configure** (**TCAN4x5x\_WDT\_Timer\_Enum** WDTtimeout)

*Configure the watchdog.*

**TCAN4x5x\_WDT\_Timer\_Enum** **WDT\_Read** (void)

*Read the watchdog configuration.*

bool **WDT\_Enable** (void)

*Enable the watchdog timer.*

bool **WDT\_Disable** (void)

*Disable the watchdog timer.*

void **WDT\_Reset** (void)

*Reset the watchdog timer.*

### Public Attributes

uint8\_t **msgBufferOut** [**BUFF\_LEN**]

uint8\_t **msgBufferIn** [**BUFF\_LEN**]

int **msgLength**

**TcanInterface** \* **can**

**ParallelInput** \* **status**

uint8\_t **slaveNbr**

### Static Public Attributes

static const int **BUFF\_LEN** = 256

### Constructor & Destructor Documentation

#### TCAN4550::TCAN4550 (volatile void \* *addr*, AlteraPio \* *rstPio*, ParallelInput \* *status0*, uint8\_t *nbr*)[inline]

### Member Function Documentation

#### void TCAN4550::Device\_ClearInterrupts (TCAN4x5x\_Device\_Interrupts \* *ir*)

Clear the device interrupts.

Will attempt to clear any interrupts that are marked as a '1' in the passed **TCAN4x5x\_Device\_Interrupts** struct

##### Parameters

|  |  |
| --- | --- |
| *\*ir* | is a pointer to a **TCAN4x5x\_Device\_Interrupts** struct containing the interrupt bit fields that will be updated |

#### void TCAN4550::Device\_ClearInterruptsAll (void )

Clear all device interrupts.

Clears all device interrupts

#### bool TCAN4550::Device\_ConfigureInterruptEnable (TCAN4x5x\_Device\_Interrupt\_Enable \* *ie*)

Configures the device interrupt enable register.

Configures the device interrupt enable register based on the passed **TCAN4x5x\_Device\_Interrupt\_Enable** struct

##### Parameters

|  |  |
| --- | --- |
| *\*ie* | is a pointer to a **TCAN4x5x\_Device\_Interrupt\_Enable** struct containing the desired enabled interrupt bits |

##### Returns

true if configuration successfully done, false if not

#### bool TCAN4550::Device\_DisableTestMode (void )

Disables the TCAN4x5x device test mode.

##### Returns

true if disabling test mode was successful, false if not

#### bool TCAN4550::Device\_EnableTestMode (TCAN4x5x\_Device\_Test\_Mode\_Enum *modeDefine*)

Sets the TCAN4x5x device test mode.

Sets the TCAN4x5x device test mode based on the input modeDefine enum

##### Parameters

|  |  |
| --- | --- |
| *modeDefine* | is an TCAN4x5x\_Device\_Test\_Mode\_Enum enum |

##### Returns

true if configuration successfully done, false if not

#### void TCAN4550::Device\_ReadDeviceIdent (uint32\_t \* *id*)

Read the TCAN4x5x device identification.

##### Parameters

|  |  |
| --- | --- |
| *id* | an array of uint32\_t for the results |

##### Returns

The register value for the device version register

#### uint16\_t TCAN4550::Device\_ReadDeviceVersion (void )

Read the TCAN4x5x device version register.

##### Returns

The register value for the device version register

#### void TCAN4550::Device\_ReadInterruptEnable (TCAN4x5x\_Device\_Interrupt\_Enable \* *ie*)

Read the device interrupt enable register.

Reads the device interrupt enable register and updates the passed **TCAN4x5x\_Device\_Interrupt\_Enable** struct

##### Parameters

|  |  |
| --- | --- |
| *\*ie* | is a pointer to a **TCAN4x5x\_Device\_Interrupt\_Enable** struct containing the interrupt bit fields that will be updated |

#### void TCAN4550::Device\_ReadInterrupts (TCAN4x5x\_Device\_Interrupts \* *ir*)

Read the device interrupts.

Reads the device interrupts and updates a **TCAN4x5x\_Device\_Interrupts** struct that is passed to the function

##### Parameters

|  |  |
| --- | --- |
| *\*ir* | is a pointer to a **TCAN4x5x\_Device\_Interrupts** struct containing the interrupt bit fields that will be updated |

#### TCAN4x5x\_Device\_Mode\_Enum TCAN4550::Device\_ReadMode (void )

Reads the TCAN4x5x device mode.

Reads the TCAN4x5x device mode and returns a modeDefine enum

##### Returns

A TCAN4x5x\_Device\_Mode\_Enum enum of the current state

#### TCAN4x5x\_Device\_Test\_Mode\_Enum TCAN4550::Device\_ReadTestMode (void )

Reads the TCAN4x5x device test mode.

##### Returns

an TCAN4x5x\_Device\_Test\_Mode\_Enum of the current device test mode

#### bool TCAN4550::Device\_SetMode (TCAN4x5x\_Device\_Mode\_Enum *modeDefine*)

Sets the TCAN4x5x device mode.

Sets the TCAN4x5x device mode based on the input modeDefine enum

##### Parameters

|  |  |
| --- | --- |
| *modeDefine* | is an TCAN4x5x\_Device\_Mode\_Enum enum |

##### Returns

true if configuration successfully done, false if not

#### void TCAN4550::disableIrq ()[inline]

#### void TCAN4550::enableIrq ()[inline]

#### void TCAN4550::MCAN\_ClearInterrupts (TCAN4x5x\_MCAN\_Interrupts \* *ir*)

Clear the MCAN interrupts.

Will attempt to clear any interrupts that are marked as a '1' in the passed **TCAN4x5x\_MCAN\_Interrupts** struct

##### Parameters

|  |  |
| --- | --- |
| *\*ir* | is a pointer to a **TCAN4x5x\_MCAN\_Interrupts** struct containing the interrupt bit fields that will be updated |

#### void TCAN4550::MCAN\_ClearInterruptsAll (void )

Clear all MCAN interrupts.

Clears all MCAN interrupts

#### bool TCAN4550::MCAN\_ConfigureCCCRRegister (TCAN4x5x\_MCAN\_CCCR\_Config \* *cccrConfig*)

Configure the MCAN CCCR Register.

Configures the bits of the CCCR register to match the CCCR config struct

##### Warning

This function writes to protected MCAN registers

##### Note

Requires that protected registers have been unlocked using TCAN4x5x\_MCAN\_EnableProtectedRegisters() and TCAN4x5x\_MCAN\_DisableProtectedRegisters() be used to lock the registers after configuration

##### Parameters

|  |  |
| --- | --- |
| *\*cccrConfig* | is a pointer to a **TCAN4x5x\_MCAN\_CCCR\_Config** struct containing the configuration bits |

##### Returns

true if successfully enabled, otherwise return false

#### bool TCAN4550::MCAN\_ConfigureDataTiming\_Raw (TCAN4x5x\_MCAN\_Data\_Timing\_Raw \* *dataTiming*)

Writes the MCAN data time settings, using the raw MCAN data timing struct.

Writes the data timing information to MCAN using the input from the \*dataTiming pointer

##### Warning

This function writes to protected MCAN registers

##### Note

Requires that protected registers have been unlocked using TCAN4x5x\_MCAN\_EnableProtectedRegisters() and TCAN4x5x\_MCAN\_DisableProtectedRegisters() be used to lock the registers after configuration

##### Parameters

|  |  |
| --- | --- |
| *\*dataTiming* | is a pointer of a **TCAN4x5x\_MCAN\_Data\_Timing\_Raw** struct containing the raw data timing information |

##### Returns

true if successfully enabled, otherwise return false

#### bool TCAN4550::MCAN\_ConfigureDataTiming\_Simple (TCAN4x5x\_MCAN\_Data\_Timing\_Simple \* *dataTiming*)

Writes the MCAN data time settings, using the simple data timing struct.

Writes the data timing information to MCAN using the input from the \*dataTiming pointer

##### Warning

This function writes to protected MCAN registers

##### Note

Requires that protected registers have been unlocked using TCAN4x5x\_MCAN\_EnableProtectedRegisters() and TCAN4x5x\_MCAN\_DisableProtectedRegisters() be used to lock the registers after configuration

##### Parameters

|  |  |
| --- | --- |
| *\*dataTiming* | is a pointer of a **TCAN4x5x\_MCAN\_Data\_Timing\_Simple** struct containing the simplified data timing information |

##### Returns

true if successfully enabled, otherwise return false

#### void TCAN4550::MCAN\_ConfigureInterruptEnable (TCAN4x5x\_MCAN\_Interrupt\_Enable \* *ie*)

Configures the MCAN interrupt enable register.

Configures the MCAN interrupt enable register based on the passed **TCAN4x5x\_MCAN\_Interrupt\_Enable** struct Also enables MCAN interrupts out to the INT1 pin.

##### Parameters

|  |  |
| --- | --- |
| *\*ie* | is a pointer to a **TCAN4x5x\_MCAN\_Interrupt\_Enable** struct containing the desired enabled interrupt bits |

#### bool TCAN4550::MCAN\_ConfigureNominalTiming\_Raw (TCAN4x5x\_MCAN\_Nominal\_Timing\_Raw \* *nomTiming*)

Writes the MCAN nominal timing settings, using the raw MCAN nominal timing struct.

Writes the data timing information to MCAN using the input from the \*nomTiming pointer

##### Warning

This function writes to protected MCAN registers

##### Note

Requires that protected registers have been unlocked using TCAN4x5x\_MCAN\_EnableProtectedRegisters() and TCAN4x5x\_MCAN\_DisableProtectedRegisters() be used to lock the registers after configuration

##### Parameters

|  |  |
| --- | --- |
| *\*nomTiming* | is a pointer of a **TCAN4x5x\_MCAN\_Nominal\_Timing\_Raw** struct containing the raw MCAN nominal timing information |

##### Returns

true if successfully enabled, otherwise return false

#### bool TCAN4550::MCAN\_ConfigureNominalTiming\_Simple (TCAN4x5x\_MCAN\_Nominal\_Timing\_Simple \* *nomTiming*)

Writes the MCAN nominal timing settings, using the simple nominal timing struct.

Writes the data timing information to MCAN using the input from the \*nomTiming pointer

##### Warning

This function writes to protected MCAN registers

##### Note

Requires that protected registers have been unlocked using TCAN4x5x\_MCAN\_EnableProtectedRegisters() and TCAN4x5x\_MCAN\_DisableProtectedRegisters() be used to lock the registers after configuration

##### Parameters

|  |  |
| --- | --- |
| *\*nomTiming* | is a pointer of a **TCAN4x5x\_MCAN\_Nominal\_Timing\_Simple** struct containing the simplified nominal timing information |

##### Returns

true if successfully enabled, otherwise return false

#### bool TCAN4550::MCAN\_DisableProtectedRegisters (void )

Disable Protected MCAN Registers.

Attempts to disable CCCR.CCE and CCCR.INIT to disallow writes to protected registers

##### Returns

true if successfully enabled, otherwise return false

#### uint8\_t TCAN4550::MCAN\_DLCtoBytes (uint8\_t *inputDLC*)

Converts the CAN message DLC hex value to the number of bytes it corresponds to.

##### Parameters

|  |  |
| --- | --- |
| *inputDLC* | is the DLC value from/to a CAN message struct |

##### Returns

The number of bytes of data (0-64 bytes)

#### bool TCAN4550::MCAN\_EnableProtectedRegisters (void )

Enable Protected MCAN Registers.

Attempts to enable CCCR.CCE and CCCR.INIT to allow writes to protected registers, needed for MCAN configuration

##### Returns

true if successfully enabled, otherwise return false

#### void TCAN4550::MCAN\_ReadCCCRRegister (TCAN4x5x\_MCAN\_CCCR\_Config \* *cccrConfig*)

Read the MCAN CCCR configuration register.

Reads the MCAN CCCR configuration register and updates the passed **TCAN4x5x\_MCAN\_CCCR\_Config** struct

##### Parameters

|  |  |
| --- | --- |
| *\*cccrConfig* | is a pointer to a **TCAN4x5x\_MCAN\_CCCR\_Config** struct containing the CCCR bit fields that will be updated |

#### void TCAN4550::MCAN\_ReadDataTimingFD\_Raw (TCAN4x5x\_MCAN\_Data\_Timing\_Raw \* *dataTiming*)

Reads the MCAN data time settings, using the raw MCAN struct.

Reads the MCAN data timing registers and updates the \*dataTiming struct

##### Parameters

|  |  |
| --- | --- |
| *\*dataTiming* | is a pointer of a **TCAN4x5x\_MCAN\_Data\_Timing\_Simple** struct containing the raw data timing information |

#### void TCAN4550::MCAN\_ReadDataTimingFD\_Simple (TCAN4x5x\_MCAN\_Data\_Timing\_Simple \* *dataTiming*)

Reads the MCAN data time settings, using the simple struct.

Reads the MCAN data timing registers and updates the \*dataTiming struct

##### Warning

This function writes to protected MCAN registers

##### Note

Requires that protected registers have been unlocked using TCAN4x5x\_MCAN\_EnableProtectedRegisters() and TCAN4x5x\_MCAN\_DisableProtectedRegisters() be used to lock the registers after configuration

##### Parameters

|  |  |
| --- | --- |
| *\*dataTiming* | is a pointer of a **TCAN4x5x\_MCAN\_Data\_Timing\_Simple** struct containing the simplified data timing information |

#### void TCAN4550::MCAN\_ReadInterruptEnable (TCAN4x5x\_MCAN\_Interrupt\_Enable \* *ie*)

Read the MCAN interrupt enable register.

Reads the MCAN interrupt enable register and updates the passed **TCAN4x5x\_MCAN\_Interrupt\_Enable** struct

##### Parameters

|  |  |
| --- | --- |
| *\*ie* | is a pointer to a **TCAN4x5x\_MCAN\_Interrupt\_Enable** struct containing the interrupt bit fields that will be updated |

#### void TCAN4550::MCAN\_ReadInterrupts (TCAN4x5x\_MCAN\_Interrupts \* *ir*)

Read the MCAN interrupts.

Reads the MCAN interrupts and updates a MCAN\_Interrupts struct that is passed to the function

##### Parameters

|  |  |
| --- | --- |
| *\*ir* | is a pointer to a MCAN\_Interrupts struct containing the interrupt bit fields that will be updated |

#### uint8\_t TCAN4550::MCAN\_ReadNextFIFO (TCAN4x5x\_MCAN\_FIFO\_Enum *FIFODefine*, TCAN4x5x\_MCAN\_RX\_Header \* *header*, uint8\_t *dataPayload*[])

Read the next MCAN FIFO element.

This function will read the next MCAN FIFO element specified and return the corresponding header information and data payload. The start address of the elment is automatically calculated by looking at the MCAN's register that says where the next element to read exists.

##### Parameters

|  |  |
| --- | --- |
| *FIFODefine* | is an TCAN4x5x\_MCAN\_FIFO\_Enum enum corresponding to either RXFIFO0 or RXFIFO1 |
| *\*header* | is a pointer to a **TCAN4x5x\_MCAN\_RX\_Header** struct containing the CAN-specific header information |
| *dataPayload[]* | is a byte array that will be updated with the read data |

##### Warning

dataPayload [] must be at least as big as the largest possible data payload, otherwise writing to out of bounds memory may occur

##### Returns

the number of bytes that were read from the TCAN4x5x and stored into dataPayload []

#### void TCAN4550::MCAN\_ReadNominalTiming\_Raw (TCAN4x5x\_MCAN\_Nominal\_Timing\_Raw \* *nomTiming*)

Reads the MCAN nominal/arbitration time settings, using the raw MCAN timing struct.

Reads the MCAN nominal timing registers and updates the \*nomTiming struct

##### Parameters

|  |  |
| --- | --- |
| *\*nomTiming* | is a pointer of a **TCAN4x5x\_MCAN\_Nominal\_Timing\_Raw** struct containing the raw MCAN nominal timing information |

#### void TCAN4550::MCAN\_ReadNominalTiming\_Simple (TCAN4x5x\_MCAN\_Nominal\_Timing\_Simple \* *nomTiming*)

Reads the MCAN nominal/arbitration time settings, using the simple timing struct.

Reads the MCAN nominal timing registers and updates the \*nomTiming struct

##### Parameters

|  |  |
| --- | --- |
| *\*nomTiming* | is a pointer of a **TCAN4x5x\_MCAN\_Nominal\_Timing\_Simple** struct containing the simplified nominal timing information |

#### uint8\_t TCAN4550::MCAN\_ReadRXBuffer (uint8\_t *bufIndex*, TCAN4x5x\_MCAN\_RX\_Header \* *header*, uint8\_t *dataPayload*[])

Read the specified RX buffer element.

This function will read the specified MCAN buffer element and return the corresponding header information and data payload. The start address of the element is automatically calculated.

##### Parameters

|  |  |
| --- | --- |
| *bufIndex* | is the RX buffer index to read from (starts at 0) |
| *\*header* | is a pointer to a **TCAN4x5x\_MCAN\_RX\_Header** struct containing the CAN-specific header information |
| *dataPayload[]* | is a byte array that will be updated with the read data |

##### Warning

dataPayload [] must be at least as big as the largest possible data payload, otherwise writing to out of bounds memory may occur

##### Returns

the number of bytes that were read from the TCAN4x5x and stored into dataPayload []

#### bool TCAN4550::MCAN\_TransmitBufferContents (uint8\_t *bufIndex*)

Transmit TX buffer contents of the specified tx buffer.

Writes the specified buffer index bit value into the TXBAR register to request a message to send

##### Parameters

|  |  |
| --- | --- |
| *bufIndex* | is the TX buffer index to write to (starts at 0) |

##### Warning

Function does NOT check if the buffer contents are valid

##### Returns

true if the request was queued, false if the buffer value was invalid (out of range)

#### uint8\_t TCAN4550::MCAN\_TXRXESC\_DataByteValue (uint8\_t *inputESCValue*)

Converts the MCAN ESC (Element Size) value to number of bytes that it corresponds to.

##### Parameters

|  |  |
| --- | --- |
| *inputESCValue* | is the value from an element size configuration register |

##### Returns

The number of bytes of data (8-64 bytes)

#### bool TCAN4550::MCAN\_WriteSIDFilter (uint8\_t *filterIndex*, TCAN4x5x\_MCAN\_SID\_Filter \* *filter*)

Write MCAN Standard ID filter into MRAM.

This function will write a standard ID MCAN filter to a specified filter element

##### Parameters

|  |  |
| --- | --- |
| *filterIndex* | is the SID filter index in MRAM to write to (starts at 0) |
| *\*filter* | is a pointer to a MCAN\_SID\_Filter struct containing the MCAN filter information |

##### Returns

true if write was successful, false if not

#### uint32\_t TCAN4550::MCAN\_WriteTXBuffer (uint8\_t *bufIndex*, TCAN4x5x\_MCAN\_TX\_Header \* *header*, uint8\_t *dataPayload*[])

Write CAN message to the specified TX buffer.

This function will write a CAN message to a specified TX buffer that can be transmitted at a later time with the **MCAN\_TransmitBufferContents()** function

##### Parameters

|  |  |
| --- | --- |
| *bufIndex* | is the TX buffer index to write to (starts at 0) |
| *\*header* | is a pointer to a MCAN\_TX\_Header struct containing the CAN-specific header information |
| *dataPayload[]* | is a byte array that contains the data payload |

##### Warning

dataPayload [] must be at least as big as the specified DLC size inside the \*header struct

##### Returns

the number of bytes that were read from the TCAN4x5x and stored into dataPayload []

#### bool TCAN4550::MCAN\_WriteXIDFilter (uint8\_t *filterIndex*, TCAN4x5x\_MCAN\_XID\_Filter \* *filter*)

Write MCAN Extended ID filter into MRAM.

This function will write an extended ID MCAN filter to a specified filter element

##### Parameters

|  |  |
| --- | --- |
| *filterIndex* | is the XID filter index in MRAM to write to (starts at 0) |
| *\*filter* | is a pointer to a MCAN\_XID\_Filter struct containing the MCAN filter information |

##### Returns

true if write was successful, false if not

#### void TCAN4550::MRAM\_Clear (void )

Clear (Zero-fill) the contents of MRAM.

Write 0s to every address in MRAM. Useful for initializing the MRAM to known values during initial configuration so that accidental ECC errors do not happen

#### bool TCAN4550::MRAM\_Configure (TCAN4x5x\_MRAM\_Config \* *MRAMConfig*)

Configures the MRAM registers.

Uses the \*MRAMConfig pointer to set up the various sections of the MRAM memory space. There are several different elements that may be configured in the MRAM, including their number of elements, as well as size of elements. This function will automatically generate the start addresses for each of the appropriate MRAM sections, attempting to place them immediately back-to-back. This function will check for over allocated memory conditions, and return false if this is found to be the case.

##### Warning

This function writes to protected MCAN registers

##### Note

Requires that protected registers have been unlocked using TCAN4x5x\_MCAN\_EnableProtectedRegisters() and TCAN4x5x\_MCAN\_DisableProtectedRegisters() be used to lock the registers after configuration

##### Parameters

|  |  |
| --- | --- |
| *\*MRAMConfig* | is a pointer of a **TCAN4x5x\_MRAM\_Config** struct containing the desired MRAM configuration |

##### Returns

true if successful, otherwise return false

#### void TCAN4550::reset ()[inline]

#### bool TCAN4550::WDT\_Configure (TCAN4x5x\_WDT\_Timer\_Enum *WDTtimeout*)

Configure the watchdog.

##### Parameters

|  |  |
| --- | --- |
| *WDTtimeout* | is an TCAN4x5x\_WDT\_Timer\_Enum enum of different times for the watch dog window |

##### Returns

true if successfully configured, or false otherwise

#### bool TCAN4550::WDT\_Disable (void )

Disable the watchdog timer.

##### Returns

true if successfully disabled, or false otherwise

#### bool TCAN4550::WDT\_Enable (void )

Enable the watchdog timer.

##### Returns

true if successfully enabled, or false otherwise

#### TCAN4x5x\_WDT\_Timer\_Enum TCAN4550::WDT\_Read (void )

Read the watchdog configuration.

##### Returns

an TCAN4x5x\_WDT\_Timer\_Enum enum of the currently configured time window

#### void TCAN4550::WDT\_Reset (void )

Reset the watchdog timer.

### Member Data Documentation

#### const int TCAN4550::BUFF\_LEN = 256[static]

#### TcanInterface\* TCAN4550::can

#### uint8\_t TCAN4550::msgBufferIn[BUFF\_LEN]

#### uint8\_t TCAN4550::msgBufferOut[BUFF\_LEN]

#### int TCAN4550::msgLength

#### uint8\_t TCAN4550::slaveNbr

#### ParallelInput\* TCAN4550::status

#### The documentation for this class was generated from the following files:

C:/Alphi/PCIeMiniSoftware/include/**TCAN4550.h**

**TCAN4550.cpp**

## TCAN4x5x\_Device\_Interrupt\_Enable Struct Reference

Struct containing the device interrupt enable bit field.

#include <TCAN4x5x\_Data\_Structs.h>

### Public Attributes

union {

  uint32\_t **word**

*Full register as single 32-bit word.*

  struct {

    uint8\_t **RESERVED1**: 8

*DEV\_IE[0:7] : RESERVED.*

    uint8\_t **CANDOMEN**: 1

*DEV\_IE[8] : CANDOM, Can bus stuck dominant.*

    uint8\_t **RESERVED2**: 1

*DEV\_IE[9] : RESERVED.*

    uint8\_t **CANTOEN**: 1

*DEV\_IE[10] : CANTO, CAN Timeout.*

    uint8\_t **RESERVED3**: 1

*DEV\_IE[11] : RESERVED.*

    uint8\_t **FRAME\_OVFEN**: 1

*DEV\_IE[12] : FRAME\_OVF, Frame Error Overflow (If Selective Wake is equipped)*

    uint8\_t **WKERREN**: 1

*DEV\_IE[13] : WKERR, Wake Error.*

    uint8\_t **LWUEN**: 1

*DEV\_IE[14] : LWU, Local Wake Up.*

    uint8\_t **CANINTEN**: 1

*DEV\_IE[15] : CANINT, CAN Bus Wake Up Interrupt.*

    uint8\_t **ECCERREN**: 1

*DEV\_IE[16] : ECCERR, MRAM ECC Error.*

    uint8\_t **RESERVED4**: 1

*DEV\_IE[17] : Reserved.*

    uint8\_t **WDTOEN**: 1

*DEV\_IE[18] : WDTO, Watchdog Time Out.*

    uint8\_t **TSDEN**: 1

*DEV\_IE[19] : TSD, Thermal Shut Down.*

    uint8\_t **PWRONEN**: 1

*DEV\_IE[20] : PWRON, Power On Interrupt.*

    uint8\_t **UVIOEN**: 1

*DEV\_IE[21] : UVIO, Undervoltage on UVIO.*

    uint8\_t **UVSUPEN**: 1

*DEV\_IE[22] : UVSUP, Undervoltage on VSUP and VCCOUT.*

    uint8\_t **SMSEN**: 1

*DEV\_IE[23] : SMS, Sleep Mode Status Flag. Set when sleep mode is entered due to WKERR, UVIO, or TSD faults.*

    uint8\_t **CANBUSBATEN**: 1

*DEV\_IE[24] : CANBUSBAT, CAN Shorted to VBAT.*

    uint8\_t **CANBUSGNDEN**: 1

*DEV\_IE[25] : CANBUSGND, CAN Shorted to GND.*

    uint8\_t **CANBUSOPENEN**: 1

*DEV\_IE[26] : CANBUSOPEN, CAN Open fault.*

    uint8\_t **CANLGNDEN**: 1

*DEV\_IE[27] : CANLGND, CANL GND.*

    uint8\_t **CANHBATEN**: 1

*DEV\_IE[28] : CANHBAT, CANH to VBAT.*

    uint8\_t **CANHCANLEN**: 1

*DEV\_IE[29] : CANHCANL, CANH and CANL shorted.*

    uint8\_t **CANBUSTERMOPENEN**: 1

*DEV\_IE[30] : CANBUSTERMOPEN, CAN Bus has termination point open.*

    uint8\_t **CANBUSNORMEN**: 1

*DEV\_IE[31] : CANBUSNOM, CAN Bus is normal flag.*

  }

};

### Detailed Description

Struct containing the device interrupt enable bit field.

### Member Data Documentation

#### union { ... }

#### uint8\_t TCAN4x5x\_Device\_Interrupt\_Enable::CANBUSBATEN

DEV\_IE[24] : CANBUSBAT, CAN Shorted to VBAT.

#### uint8\_t TCAN4x5x\_Device\_Interrupt\_Enable::CANBUSGNDEN

DEV\_IE[25] : CANBUSGND, CAN Shorted to GND.

#### uint8\_t TCAN4x5x\_Device\_Interrupt\_Enable::CANBUSNORMEN

DEV\_IE[31] : CANBUSNOM, CAN Bus is normal flag.

#### uint8\_t TCAN4x5x\_Device\_Interrupt\_Enable::CANBUSOPENEN

DEV\_IE[26] : CANBUSOPEN, CAN Open fault.

#### uint8\_t TCAN4x5x\_Device\_Interrupt\_Enable::CANBUSTERMOPENEN

DEV\_IE[30] : CANBUSTERMOPEN, CAN Bus has termination point open.

#### uint8\_t TCAN4x5x\_Device\_Interrupt\_Enable::CANDOMEN

DEV\_IE[8] : CANDOM, Can bus stuck dominant.

#### uint8\_t TCAN4x5x\_Device\_Interrupt\_Enable::CANHBATEN

DEV\_IE[28] : CANHBAT, CANH to VBAT.

#### uint8\_t TCAN4x5x\_Device\_Interrupt\_Enable::CANHCANLEN

DEV\_IE[29] : CANHCANL, CANH and CANL shorted.

#### uint8\_t TCAN4x5x\_Device\_Interrupt\_Enable::CANINTEN

DEV\_IE[15] : CANINT, CAN Bus Wake Up Interrupt.

#### uint8\_t TCAN4x5x\_Device\_Interrupt\_Enable::CANLGNDEN

DEV\_IE[27] : CANLGND, CANL GND.

#### uint8\_t TCAN4x5x\_Device\_Interrupt\_Enable::CANTOEN

DEV\_IE[10] : CANTO, CAN Timeout.

#### uint8\_t TCAN4x5x\_Device\_Interrupt\_Enable::ECCERREN

DEV\_IE[16] : ECCERR, MRAM ECC Error.

#### uint8\_t TCAN4x5x\_Device\_Interrupt\_Enable::FRAME\_OVFEN

DEV\_IE[12] : FRAME\_OVF, Frame Error Overflow (If Selective Wake is equipped)

#### uint8\_t TCAN4x5x\_Device\_Interrupt\_Enable::LWUEN

DEV\_IE[14] : LWU, Local Wake Up.

#### uint8\_t TCAN4x5x\_Device\_Interrupt\_Enable::PWRONEN

DEV\_IE[20] : PWRON, Power On Interrupt.

#### uint8\_t TCAN4x5x\_Device\_Interrupt\_Enable::RESERVED1

DEV\_IE[0:7] : RESERVED.

#### uint8\_t TCAN4x5x\_Device\_Interrupt\_Enable::RESERVED2

DEV\_IE[9] : RESERVED.

#### uint8\_t TCAN4x5x\_Device\_Interrupt\_Enable::RESERVED3

DEV\_IE[11] : RESERVED.

#### uint8\_t TCAN4x5x\_Device\_Interrupt\_Enable::RESERVED4

DEV\_IE[17] : Reserved.

#### uint8\_t TCAN4x5x\_Device\_Interrupt\_Enable::SMSEN

DEV\_IE[23] : SMS, Sleep Mode Status Flag. Set when sleep mode is entered due to WKERR, UVIO, or TSD faults.

#### uint8\_t TCAN4x5x\_Device\_Interrupt\_Enable::TSDEN

DEV\_IE[19] : TSD, Thermal Shut Down.

#### uint8\_t TCAN4x5x\_Device\_Interrupt\_Enable::UVIOEN

DEV\_IE[21] : UVIO, Undervoltage on UVIO.

#### uint8\_t TCAN4x5x\_Device\_Interrupt\_Enable::UVSUPEN

DEV\_IE[22] : UVSUP, Undervoltage on VSUP and VCCOUT.

#### uint8\_t TCAN4x5x\_Device\_Interrupt\_Enable::WDTOEN

DEV\_IE[18] : WDTO, Watchdog Time Out.

#### uint8\_t TCAN4x5x\_Device\_Interrupt\_Enable::WKERREN

DEV\_IE[13] : WKERR, Wake Error.

#### uint32\_t TCAN4x5x\_Device\_Interrupt\_Enable::word

Full register as single 32-bit word.

#### The documentation for this struct was generated from the following file:

C:/Alphi/PCIeMiniSoftware/include/**TCAN4x5x\_Data\_Structs.h**

## TCAN4x5x\_Device\_Interrupts Struct Reference

Struct containing the device interrupt bit field.

#include <TCAN4x5x\_Data\_Structs.h>

### Public Attributes

union {

  uint32\_t **word**

*Full register as single 32-bit word.*

  struct {

    uint8\_t **VTWD**: 1

*DEV\_IR[0] VTWD: Global Voltage, Temp, or Watchdog (if equipped) Interrupt.*

    uint8\_t **M\_CAN\_INT**: 1

*DEV\_IR[1] M\_CAN\_INT: There are MCAN interrupts pending.*

    uint8\_t **SWERR**: 1

*DEV\_IR[2] : Selective Wake Error (If equipped)*

    uint8\_t **SPIERR**: 1

*DEV\_IR[3] : SPI Error.*

    uint8\_t **CBF**: 1

*DEV\_IR[4] : CBF, CAN Bus Fault.*

    uint8\_t **CANERR**: 1

*DEV\_IR[5] : CANERR, CAN Error.*

    uint8\_t **WKRQ**: 1

*DEV\_IR[6] : WKRQ, Wake Request.*

    uint8\_t **GLOBALERR**: 1

*DEV\_IR[7] : GLOBALERR, Global Error. Is the OR output of all interrupts.*

    uint8\_t **CANDOM**: 1

*DEV\_IR[8] : CANDOM, Can bus stuck dominant.*

    uint8\_t **RESERVED**: 1

*DEV\_IR[9] : RESERVED.*

    uint8\_t **CANTO**: 1

*DEV\_IR[10] : CANTO, CAN Timeout.*

    uint8\_t **RESERVED2**: 1

*DEV\_IR[11] : RESERVED.*

    uint8\_t **FRAME\_OVF**: 1

*DEV\_IR[12] : FRAME\_OVF, Frame Error Overflow (If Selective Wake is equipped)*

    uint8\_t **WKERR**: 1

*DEV\_IR[13] : WKERR, Wake Error.*

    uint8\_t **LWU**: 1

*DEV\_IR[14] : LWU, Local Wake Up.*

    uint8\_t **CANINT**: 1

*DEV\_IR[15] : CANINT, CAN Bus Wake Up Interrupt.*

    uint8\_t **ECCERR**: 1

*DEV\_IR[16] : ECCERR, MRAM ECC Error.*

    uint8\_t **RESERVED3**: 1

*DEV\_IR[17] : Reserved.*

    uint8\_t **WDTO**: 1

*DEV\_IR[18] : WDTO, Watchdog Time Out.*

    uint8\_t **TSD**: 1

*DEV\_IR[19] : TSD, Thermal Shut Down.*

    uint8\_t **PWRON**: 1

*DEV\_IR[20] : PWRON, Power On Interrupt.*

    uint8\_t **UVIO**: 1

*DEV\_IR[21] : UVIO, Undervoltage on UVIO.*

    uint8\_t **UVSUP**: 1

*DEV\_IR[22] : UVSUP, Undervoltage on VSUP and VCCOUT.*

    uint8\_t **SMS**: 1

*DEV\_IR[23] : SMS, Sleep Mode Status Flag. Set when sleep mode is entered due to WKERR, UVIO, or TSD faults.*

    uint8\_t **CANBUSBAT**: 1

*DEV\_IR[24] : CANBUSBAT, CAN Shorted to VBAT.*

    uint8\_t **CANBUSGND**: 1

*DEV\_IR[25] : CANBUSGND, CAN Shorted to GND.*

    uint8\_t **CANBUSOPEN**: 1

*DEV\_IR[26] : CANBUSOPEN, CAN Open fault.*

    uint8\_t **CANLGND**: 1

*DEV\_IR[27] : CANLGND, CANL GND.*

    uint8\_t **CANHBAT**: 1

*DEV\_IR[28] : CANHBAT, CANH to VBAT.*

    uint8\_t **CANHCANL**: 1

*DEV\_IR[29] : CANHCANL, CANH and CANL shorted.*

    uint8\_t **CANBUSTERMOPEN**: 1

*DEV\_IR[30] : CANBUSTERMOPEN, CAN Bus has termination point open.*

    uint8\_t **CANBUSNORM**: 1

*DEV\_IR[31] : CANBUSNOM, CAN Bus is normal flag.*

  }

};

### Detailed Description

Struct containing the device interrupt bit field.

### Member Data Documentation

#### union { ... }

#### uint8\_t TCAN4x5x\_Device\_Interrupts::CANBUSBAT

DEV\_IR[24] : CANBUSBAT, CAN Shorted to VBAT.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::CANBUSGND

DEV\_IR[25] : CANBUSGND, CAN Shorted to GND.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::CANBUSNORM

DEV\_IR[31] : CANBUSNOM, CAN Bus is normal flag.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::CANBUSOPEN

DEV\_IR[26] : CANBUSOPEN, CAN Open fault.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::CANBUSTERMOPEN

DEV\_IR[30] : CANBUSTERMOPEN, CAN Bus has termination point open.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::CANDOM

DEV\_IR[8] : CANDOM, Can bus stuck dominant.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::CANERR

DEV\_IR[5] : CANERR, CAN Error.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::CANHBAT

DEV\_IR[28] : CANHBAT, CANH to VBAT.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::CANHCANL

DEV\_IR[29] : CANHCANL, CANH and CANL shorted.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::CANINT

DEV\_IR[15] : CANINT, CAN Bus Wake Up Interrupt.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::CANLGND

DEV\_IR[27] : CANLGND, CANL GND.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::CANTO

DEV\_IR[10] : CANTO, CAN Timeout.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::CBF

DEV\_IR[4] : CBF, CAN Bus Fault.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::ECCERR

DEV\_IR[16] : ECCERR, MRAM ECC Error.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::FRAME\_OVF

DEV\_IR[12] : FRAME\_OVF, Frame Error Overflow (If Selective Wake is equipped)

#### uint8\_t TCAN4x5x\_Device\_Interrupts::GLOBALERR

DEV\_IR[7] : GLOBALERR, Global Error. Is the OR output of all interrupts.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::LWU

DEV\_IR[14] : LWU, Local Wake Up.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::M\_CAN\_INT

DEV\_IR[1] M\_CAN\_INT: There are MCAN interrupts pending.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::PWRON

DEV\_IR[20] : PWRON, Power On Interrupt.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::RESERVED

DEV\_IR[9] : RESERVED.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::RESERVED2

DEV\_IR[11] : RESERVED.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::RESERVED3

DEV\_IR[17] : Reserved.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::SMS

DEV\_IR[23] : SMS, Sleep Mode Status Flag. Set when sleep mode is entered due to WKERR, UVIO, or TSD faults.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::SPIERR

DEV\_IR[3] : SPI Error.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::SWERR

DEV\_IR[2] : Selective Wake Error (If equipped)

#### uint8\_t TCAN4x5x\_Device\_Interrupts::TSD

DEV\_IR[19] : TSD, Thermal Shut Down.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::UVIO

DEV\_IR[21] : UVIO, Undervoltage on UVIO.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::UVSUP

DEV\_IR[22] : UVSUP, Undervoltage on VSUP and VCCOUT.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::VTWD

DEV\_IR[0] VTWD: Global Voltage, Temp, or Watchdog (if equipped) Interrupt.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::WDTO

DEV\_IR[18] : WDTO, Watchdog Time Out.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::WKERR

DEV\_IR[13] : WKERR, Wake Error.

#### uint8\_t TCAN4x5x\_Device\_Interrupts::WKRQ

DEV\_IR[6] : WKRQ, Wake Request.

#### uint32\_t TCAN4x5x\_Device\_Interrupts::word

Full register as single 32-bit word.

#### The documentation for this struct was generated from the following file:

C:/Alphi/PCIeMiniSoftware/include/**TCAN4x5x\_Data\_Structs.h**

## TCAN4x5x\_MCAN\_CCCR\_Config Struct Reference

struct containing the bit fields of the MCAN CCCR register

#include <TCAN4x5x\_Data\_Structs.h>

### Public Attributes

union {

  uint32\_t **word**

*Full register as single 32-bit word.*

  struct {

    uint8\_t **reserved**: 2

*Reserved (0)*

    uint8\_t **ASM**: 1

*ASM: Restricted Operation Mode. The device can only listen to CAN traffic and acknowledge, but not send anything.*

    uint8\_t **reserved2**: 1

*Reserved (0)*

    uint8\_t **CSR**: 1

*CSR: Clock stop request.*

    uint8\_t **MON**: 1

*MON: Bus monitoring mode. The device may only listen to CAN traffic, and is not allowed to acknowledge or send error frames.*

    uint8\_t **DAR**: 1

*DAR: Disable automatic retransmission. If a transmission errors, gets a NACK, or loses arbitration, the MCAN controller will NOT try to transmit again.*

    uint8\_t **TEST**: 1

*TEST: MCAN Test mode enable.*

    uint8\_t **FDOE**: 1

*FDOE: Can FD mode enabled, master enable for CAN FD support.*

    uint8\_t **BRSE**: 1

*BRSE: Bit rate switch enabled for can FD. Master enable for bit rate switching support.*

    uint8\_t **reserved3**: 2

*Reserved (0)*

    uint8\_t **PXHD**: 1

*PXHD: Protocol exception handling disable*

*0 = Protocol exception handling enabled [default]*

*1 = protocol exception handling disabled.*

    uint8\_t **EFBI**: 1

*EFBI: Edge filtering during bus integration. 0 Disables this [default].*

    uint8\_t **TXP**: 1

*TXP: Transmit Pause Enable: Pause for 2 can bit times before next transmission.*

    uint8\_t **NISO**: 1

*NSIO: Non Iso Operation*

*0: CAN FD frame format according to ISO 11898-1:2015 [default]*

*1: CAN FD frame format according to Bosch CAN FD Spec v1.*

  }

};

### Detailed Description

struct containing the bit fields of the MCAN CCCR register

### Member Data Documentation

#### union { ... }

#### uint8\_t TCAN4x5x\_MCAN\_CCCR\_Config::ASM

ASM: Restricted Operation Mode. The device can only listen to CAN traffic and acknowledge, but not send anything.

#### uint8\_t TCAN4x5x\_MCAN\_CCCR\_Config::BRSE

BRSE: Bit rate switch enabled for can FD. Master enable for bit rate switching support.

#### uint8\_t TCAN4x5x\_MCAN\_CCCR\_Config::CSR

CSR: Clock stop request.

#### uint8\_t TCAN4x5x\_MCAN\_CCCR\_Config::DAR

DAR: Disable automatic retransmission. If a transmission errors, gets a NACK, or loses arbitration, the MCAN controller will NOT try to transmit again.

#### uint8\_t TCAN4x5x\_MCAN\_CCCR\_Config::EFBI

EFBI: Edge filtering during bus integration. 0 Disables this [default].

#### uint8\_t TCAN4x5x\_MCAN\_CCCR\_Config::FDOE

FDOE: Can FD mode enabled, master enable for CAN FD support.

#### uint8\_t TCAN4x5x\_MCAN\_CCCR\_Config::MON

MON: Bus monitoring mode. The device may only listen to CAN traffic, and is not allowed to acknowledge or send error frames.

#### uint8\_t TCAN4x5x\_MCAN\_CCCR\_Config::NISO

NSIO: Non Iso Operation

0: CAN FD frame format according to ISO 11898-1:2015 [default]

1: CAN FD frame format according to Bosch CAN FD Spec v1.

#### uint8\_t TCAN4x5x\_MCAN\_CCCR\_Config::PXHD

PXHD: Protocol exception handling disable

0 = Protocol exception handling enabled [default]

1 = protocol exception handling disabled.

#### uint8\_t TCAN4x5x\_MCAN\_CCCR\_Config::reserved

Reserved (0)

#### uint8\_t TCAN4x5x\_MCAN\_CCCR\_Config::reserved2

Reserved (0)

#### uint8\_t TCAN4x5x\_MCAN\_CCCR\_Config::reserved3

Reserved (0)

#### uint8\_t TCAN4x5x\_MCAN\_CCCR\_Config::TEST

TEST: MCAN Test mode enable.

#### uint8\_t TCAN4x5x\_MCAN\_CCCR\_Config::TXP

TXP: Transmit Pause Enable: Pause for 2 can bit times before next transmission.

#### uint32\_t TCAN4x5x\_MCAN\_CCCR\_Config::word

Full register as single 32-bit word.

#### The documentation for this struct was generated from the following file:

C:/Alphi/PCIeMiniSoftware/include/**TCAN4x5x\_Data\_Structs.h**

## TCAN4x5x\_MCAN\_Data\_Timing\_Raw Struct Reference

Used to setup the timing parameters of the MCAN module This is the raw MCAN form of the struct which takes in the same values as the actual Bosch MCAN core.

#include <TCAN4x5x\_Data\_Structs.h>

### Public Attributes

uint8\_t **DataBitRatePrescaler**: 5

*DBRP: The prescaler value from the MCAN system clock. Interpreted by MCAN as the value is this field + 1*

*Valid range is: 0 to 31.*

uint8\_t **DataTimeSeg1andProp**: 5

*DTSEG1: Data time segment 1 + prop segment value. Interpreted by MCAN as the value in this field + 1*

*Valid values are: 0 to 31.*

uint8\_t **DataTimeSeg2**: 4

*DTSEG2: Data time segment 2. Interpreted by MCAN as the value is this field + 1*

*Valid values are: 0 to 15.*

uint8\_t **DataSyncJumpWidth**: 4

*DSJW: Data Resynchronization jump width. Interpreted by MCAN as the value is this field + 1*

*Valid values are: 0 to 15.*

uint8\_t **TDCOffset**: 7

*TDCO: Transmitter delay compensation offset*

*Valid values are 0 to 127 mtq.*

uint8\_t **TDCFilter**: 7

*TDCFilter: Transmitter delay compensation Filter Window Length*

*Valid values are 0 to 127 mtq.*

### Detailed Description

Used to setup the timing parameters of the MCAN module This is the raw MCAN form of the struct which takes in the same values as the actual Bosch MCAN core.

### Member Data Documentation

#### uint8\_t TCAN4x5x\_MCAN\_Data\_Timing\_Raw::DataBitRatePrescaler

DBRP: The prescaler value from the MCAN system clock. Interpreted by MCAN as the value is this field + 1

Valid range is: 0 to 31.

#### uint8\_t TCAN4x5x\_MCAN\_Data\_Timing\_Raw::DataSyncJumpWidth

DSJW: Data Resynchronization jump width. Interpreted by MCAN as the value is this field + 1

Valid values are: 0 to 15.

#### uint8\_t TCAN4x5x\_MCAN\_Data\_Timing\_Raw::DataTimeSeg1andProp

DTSEG1: Data time segment 1 + prop segment value. Interpreted by MCAN as the value in this field + 1

Valid values are: 0 to 31.

#### uint8\_t TCAN4x5x\_MCAN\_Data\_Timing\_Raw::DataTimeSeg2

DTSEG2: Data time segment 2. Interpreted by MCAN as the value is this field + 1

Valid values are: 0 to 15.

#### uint8\_t TCAN4x5x\_MCAN\_Data\_Timing\_Raw::TDCFilter

TDCFilter: Transmitter delay compensation Filter Window Length

Valid values are 0 to 127 mtq.

#### uint8\_t TCAN4x5x\_MCAN\_Data\_Timing\_Raw::TDCOffset

TDCO: Transmitter delay compensation offset

Valid values are 0 to 127 mtq.

#### The documentation for this struct was generated from the following file:

C:/Alphi/PCIeMiniSoftware/include/**TCAN4x5x\_Data\_Structs.h**

## TCAN4x5x\_MCAN\_Data\_Timing\_Simple Struct Reference

Used to setup the data timing parameters of the MCAN module This is a simplified struct, requiring only the prescaler value (1:x), number of time quanta before and after the sample point.

#include <TCAN4x5x\_Data\_Structs.h>

### Public Attributes

uint8\_t **DataBitRatePrescaler**: 6

*Prescaler value, interpreted as 1:x*

*Valid range is: 1 to 32.*

uint8\_t **DataTqBeforeSamplePoint**: 6

*DTQBSP: Number of time quanta before sample point*

*Valid values are: 2 to 33.*

uint8\_t **DataTqAfterSamplePoint**: 5

*DTQASP: Number of time quanta after sample point*

*Valid values are: 1 to 16.*

### Detailed Description

Used to setup the data timing parameters of the MCAN module This is a simplified struct, requiring only the prescaler value (1:x), number of time quanta before and after the sample point.

### Member Data Documentation

#### uint8\_t TCAN4x5x\_MCAN\_Data\_Timing\_Simple::DataBitRatePrescaler

Prescaler value, interpreted as 1:x

Valid range is: 1 to 32.

#### uint8\_t TCAN4x5x\_MCAN\_Data\_Timing\_Simple::DataTqAfterSamplePoint

DTQASP: Number of time quanta after sample point

Valid values are: 1 to 16.

#### uint8\_t TCAN4x5x\_MCAN\_Data\_Timing\_Simple::DataTqBeforeSamplePoint

DTQBSP: Number of time quanta before sample point

Valid values are: 2 to 33.

#### The documentation for this struct was generated from the following file:

C:/Alphi/PCIeMiniSoftware/include/**TCAN4x5x\_Data\_Structs.h**

## TCAN4x5x\_MCAN\_Interrupt\_Enable Struct Reference

Struct containing the MCAN interrupt enable bit field.

#include <TCAN4x5x\_Data\_Structs.h>

### Public Attributes

union {

  uint32\_t **word**

*Full register as single 32-bit word.*

  struct {

    uint8\_t **RF0NE**: 1

*IE[0] RF0NE: Rx FIFO 0 new message.*

    uint8\_t **RF0WE**: 1

*IE[1] RF0WE: Rx FIFO 0 watermark reached.*

    uint8\_t **RF0FE**: 1

*IE[2] RF0FE: Rx FIFO 0 full.*

    uint8\_t **RF0LE**: 1

*IE[3] RF0LE: Rx FIFO 0 message lost.*

    uint8\_t **RF1NE**: 1

*IE[4] RF1NE: Rx FIFO 1 new message.*

    uint8\_t **RF1WE**: 1

*IE[5] RF1WE: RX FIFO 1 watermark reached.*

    uint8\_t **RF1FE**: 1

*IE[6] RF1FE: Rx FIFO 1 full.*

    uint8\_t **RF1LE**: 1

*IE[7] RF1LE: Rx FIFO 1 message lost.*

    uint8\_t **HPME**: 1

*IE[8] HPME: High priority message.*

    uint8\_t **TCE**: 1

*IE[9] TCE: Transmission completed.*

    uint8\_t **TCFE**: 1

*IE[10] TCFE: Transmission cancellation finished.*

    uint8\_t **TFEE**: 1

*IE[11] TFEE: Tx FIFO Empty.*

    uint8\_t **TEFNE**: 1

*IE[12] TEFNE: Tx Event FIFO new entry.*

    uint8\_t **TEFWE**: 1

*IE[13] TEFWE: Tx Event FIFO watermark reached.*

    uint8\_t **TEFFE**: 1

*IE[14] TEFFE: Tx Event FIFO full.*

    uint8\_t **TEFLE**: 1

*IE[15] TEFLE: Tx Event FIFO element lost.*

    uint8\_t **TSWE**: 1

*IE[16] TSWE: Timestamp wraparound.*

    uint8\_t **MRAFE**: 1

*IE[17] MRAFE: Message RAM access failure.*

    uint8\_t **TOOE**: 1

*IE[18] TOOE: Time out occured.*

    uint8\_t **DRXE**: 1

*IE[19] DRXE: Message stored to dedicated RX buffer.*

    uint8\_t **BECE**: 1

*IE[20] BECE: MRAM Bit error corrected.*

    uint8\_t **BEUE**: 1

*IE[21] BEUE: MRAM Bit error uncorrected.*

    uint8\_t **ELOE**: 1

*IE[22] ELOE: Error logging overflow.*

    uint8\_t **EPE**: 1

*IE[23] EPE: Error\_passive status changed.*

    uint8\_t **EWE**: 1

*IE[24] EWE: Error\_warning status changed.*

    uint8\_t **BOE**: 1

*IE[25] BOE: Bus\_off status changed.*

    uint8\_t **WDIE**: 1

*IE[26] WDIE: MRAM Watchdog Interrupt.*

    uint8\_t **PEAE**: 1

*IE[27] PEAE Protocol Error in arbitration phase (nominal bit time used)*

    uint8\_t **PEDE**: 1

*IE[28] PEDE: Protocol error in data phase (data bit time is used)*

    uint8\_t **ARAE**: 1

*IE[29] ARAE: Access to reserved address.*

    uint8\_t **reserved**: 2

*IE[30:31] Reserved, not writable.*

  }

};

### Detailed Description

Struct containing the MCAN interrupt enable bit field.

### Member Data Documentation

#### union { ... }

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::ARAE

IE[29] ARAE: Access to reserved address.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::BECE

IE[20] BECE: MRAM Bit error corrected.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::BEUE

IE[21] BEUE: MRAM Bit error uncorrected.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::BOE

IE[25] BOE: Bus\_off status changed.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::DRXE

IE[19] DRXE: Message stored to dedicated RX buffer.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::ELOE

IE[22] ELOE: Error logging overflow.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::EPE

IE[23] EPE: Error\_passive status changed.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::EWE

IE[24] EWE: Error\_warning status changed.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::HPME

IE[8] HPME: High priority message.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::MRAFE

IE[17] MRAFE: Message RAM access failure.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::PEAE

IE[27] PEAE Protocol Error in arbitration phase (nominal bit time used)

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::PEDE

IE[28] PEDE: Protocol error in data phase (data bit time is used)

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::reserved

IE[30:31] Reserved, not writable.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::RF0FE

IE[2] RF0FE: Rx FIFO 0 full.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::RF0LE

IE[3] RF0LE: Rx FIFO 0 message lost.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::RF0NE

IE[0] RF0NE: Rx FIFO 0 new message.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::RF0WE

IE[1] RF0WE: Rx FIFO 0 watermark reached.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::RF1FE

IE[6] RF1FE: Rx FIFO 1 full.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::RF1LE

IE[7] RF1LE: Rx FIFO 1 message lost.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::RF1NE

IE[4] RF1NE: Rx FIFO 1 new message.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::RF1WE

IE[5] RF1WE: RX FIFO 1 watermark reached.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::TCE

IE[9] TCE: Transmission completed.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::TCFE

IE[10] TCFE: Transmission cancellation finished.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::TEFFE

IE[14] TEFFE: Tx Event FIFO full.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::TEFLE

IE[15] TEFLE: Tx Event FIFO element lost.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::TEFNE

IE[12] TEFNE: Tx Event FIFO new entry.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::TEFWE

IE[13] TEFWE: Tx Event FIFO watermark reached.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::TFEE

IE[11] TFEE: Tx FIFO Empty.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::TOOE

IE[18] TOOE: Time out occured.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::TSWE

IE[16] TSWE: Timestamp wraparound.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::WDIE

IE[26] WDIE: MRAM Watchdog Interrupt.

#### uint32\_t TCAN4x5x\_MCAN\_Interrupt\_Enable::word

Full register as single 32-bit word.

#### The documentation for this struct was generated from the following file:

C:/Alphi/PCIeMiniSoftware/include/**TCAN4x5x\_Data\_Structs.h**

## TCAN4x5x\_MCAN\_Interrupts Struct Reference

Struct containing the MCAN interrupt bit field.

#include <TCAN4x5x\_Data\_Structs.h>

### Public Attributes

union {

  uint32\_t **word**

*Full register as single 32-bit word.*

  struct {

    uint8\_t **RF0N**: 1

*IR[0] RF0N: Rx FIFO 0 new message.*

    uint8\_t **RF0W**: 1

*IR[1] RF0W: Rx FIFO 0 watermark reached.*

    uint8\_t **RF0F**: 1

*IR[2] RF0F: Rx FIFO 0 full.*

    uint8\_t **RF0L**: 1

*IR[3] RF0L: Rx FIFO 0 message lost.*

    uint8\_t **RF1N**: 1

*IR[4] RF1N: Rx FIFO 1 new message.*

    uint8\_t **RF1W**: 1

*IR[5] RF1W: RX FIFO 1 watermark reached.*

    uint8\_t **RF1F**: 1

*IR[6] RF1F: Rx FIFO 1 full.*

    uint8\_t **RF1L**: 1

*IR[7] RF1L: Rx FIFO 1 message lost.*

    uint8\_t **HPM**: 1

*IR[8] HPM: High priority message.*

    uint8\_t **TC**: 1

*IR[9] TC: Transmission completed.*

    uint8\_t **TCF**: 1

*IR[10] TCF: Transmission cancellation finished.*

    uint8\_t **TFE**: 1

*IR[11] TFE: Tx FIFO Empty.*

    uint8\_t **TEFN**: 1

*IR[12] TEFN: Tx Event FIFO new entry.*

    uint8\_t **TEFW**: 1

*IR[13] TEFW: Tx Event FIFO water mark reached.*

    uint8\_t **TEFF**: 1

*IR[14] TEFF: Tx Event FIFO full.*

    uint8\_t **TEFL**: 1

*IR[15] TEFL: Tx Event FIFO element lost.*

    uint8\_t **TSW**: 1

*IR[16] TSW: Timestamp wrapped around.*

    uint8\_t **MRAF**: 1

*IR[17] MRAF: Message RAM access failure.*

    uint8\_t **TOO**: 1

*IR[18] TOO: Time out occurred.*

    uint8\_t **DRX**: 1

*IR[19] DRX: Message stored to dedicated RX buffer.*

    uint8\_t **BEC**: 1

*IR[20] BEC: MRAM Bit error corrected.*

    uint8\_t **BEU**: 1

*IR[21] BEU: MRAM Bit error uncorrected.*

    uint8\_t **ELO**: 1

*IR[22] ELO: Error logging overflow.*

    uint8\_t **EP**: 1

*IR[23] EP: Error\_passive status changed.*

    uint8\_t **EW**: 1

*IR[24] EW: Error\_warning status changed.*

    uint8\_t **BO**: 1

*IR[25] BO: Bus\_off status changed.*

    uint8\_t **WDI**: 1

*IR[26] WDI: MRAM Watchdog Interrupt.*

    uint8\_t **PEA**: 1

*IR[27] PEA Protocol Error in arbitration phase (nominal bit time used)*

    uint8\_t **PED**: 1

*IR[28] PED: Protocol error in data phase (data bit time is used)*

    uint8\_t **ARA**: 1

*IR[29] ARA: Access to reserved address.*

    uint8\_t **reserved**: 2

*IR[30:31] Reserved, not writable.*

  }

};

### Detailed Description

Struct containing the MCAN interrupt bit field.

### Member Data Documentation

#### union { ... }

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::ARA

IR[29] ARA: Access to reserved address.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::BEC

IR[20] BEC: MRAM Bit error corrected.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::BEU

IR[21] BEU: MRAM Bit error uncorrected.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::BO

IR[25] BO: Bus\_off status changed.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::DRX

IR[19] DRX: Message stored to dedicated RX buffer.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::ELO

IR[22] ELO: Error logging overflow.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::EP

IR[23] EP: Error\_passive status changed.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::EW

IR[24] EW: Error\_warning status changed.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::HPM

IR[8] HPM: High priority message.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::MRAF

IR[17] MRAF: Message RAM access failure.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::PEA

IR[27] PEA Protocol Error in arbitration phase (nominal bit time used)

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::PED

IR[28] PED: Protocol error in data phase (data bit time is used)

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::reserved

IR[30:31] Reserved, not writable.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::RF0F

IR[2] RF0F: Rx FIFO 0 full.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::RF0L

IR[3] RF0L: Rx FIFO 0 message lost.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::RF0N

IR[0] RF0N: Rx FIFO 0 new message.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::RF0W

IR[1] RF0W: Rx FIFO 0 watermark reached.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::RF1F

IR[6] RF1F: Rx FIFO 1 full.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::RF1L

IR[7] RF1L: Rx FIFO 1 message lost.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::RF1N

IR[4] RF1N: Rx FIFO 1 new message.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::RF1W

IR[5] RF1W: RX FIFO 1 watermark reached.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::TC

IR[9] TC: Transmission completed.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::TCF

IR[10] TCF: Transmission cancellation finished.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::TEFF

IR[14] TEFF: Tx Event FIFO full.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::TEFL

IR[15] TEFL: Tx Event FIFO element lost.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::TEFN

IR[12] TEFN: Tx Event FIFO new entry.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::TEFW

IR[13] TEFW: Tx Event FIFO water mark reached.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::TFE

IR[11] TFE: Tx FIFO Empty.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::TOO

IR[18] TOO: Time out occurred.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::TSW

IR[16] TSW: Timestamp wrapped around.

#### uint8\_t TCAN4x5x\_MCAN\_Interrupts::WDI

IR[26] WDI: MRAM Watchdog Interrupt.

#### uint32\_t TCAN4x5x\_MCAN\_Interrupts::word

Full register as single 32-bit word.

#### The documentation for this struct was generated from the following file:

C:/Alphi/PCIeMiniSoftware/include/**TCAN4x5x\_Data\_Structs.h**

## TCAN4x5x\_MCAN\_Nominal\_Timing\_Raw Struct Reference

Used to setup the nominal timing parameters of the MCAN module This is the raw MCAN form of the struct which takes in the same values as the actual Bosch MCAN core.

#include <TCAN4x5x\_Data\_Structs.h>

### Public Attributes

uint16\_t **NominalBitRatePrescaler**: 9

*NBRP: The prescaler value from the MCAN system clock. Interpreted by MCAN as the value is this field + 1*

*Valid range is: 0 to 511.*

uint8\_t **NominalTimeSeg1andProp**: 8

*NTSEG1: Data time segment 1 + prop segment value. Interpreted by MCAN as the value is this field + 1*

*Valid values are: 0 to 255.*

uint8\_t **NominalTimeSeg2**: 7

*NTSEG2: Data time segment 2. Interpreted by MCAN as the value is this field + 1*

*Valid values are: 0 to 127.*

uint8\_t **NominalSyncJumpWidth**: 7

*NSJW: Nominal time Resynchronization jump width. Interpreted by MCAN as the value is this field + 1*

*Valid values are: 0 to 127.*

### Detailed Description

Used to setup the nominal timing parameters of the MCAN module This is the raw MCAN form of the struct which takes in the same values as the actual Bosch MCAN core.

### Member Data Documentation

#### uint16\_t TCAN4x5x\_MCAN\_Nominal\_Timing\_Raw::NominalBitRatePrescaler

NBRP: The prescaler value from the MCAN system clock. Interpreted by MCAN as the value is this field + 1

Valid range is: 0 to 511.

#### uint8\_t TCAN4x5x\_MCAN\_Nominal\_Timing\_Raw::NominalSyncJumpWidth

NSJW: Nominal time Resynchronization jump width. Interpreted by MCAN as the value is this field + 1

Valid values are: 0 to 127.

#### uint8\_t TCAN4x5x\_MCAN\_Nominal\_Timing\_Raw::NominalTimeSeg1andProp

NTSEG1: Data time segment 1 + prop segment value. Interpreted by MCAN as the value is this field + 1

Valid values are: 0 to 255.

#### uint8\_t TCAN4x5x\_MCAN\_Nominal\_Timing\_Raw::NominalTimeSeg2

NTSEG2: Data time segment 2. Interpreted by MCAN as the value is this field + 1

Valid values are: 0 to 127.

#### The documentation for this struct was generated from the following file:

C:/Alphi/PCIeMiniSoftware/include/**TCAN4x5x\_Data\_Structs.h**

## TCAN4x5x\_MCAN\_Nominal\_Timing\_Simple Struct Reference

Used to setup the nominal timing parameters of the MCAN module This is a simplified struct, requiring only the prescaler value (1:x), number of time quanta before and after the sample point.

#include <TCAN4x5x\_Data\_Structs.h>

### Public Attributes

uint16\_t **NominalBitRatePrescaler**: 10

*NBRP: The prescaler value from the MCAN system clock. Value interpreted as 1:x*

*Valid range is: 1 to 512.*

uint16\_t **NominalTqBeforeSamplePoint**: 9

*NTQBSP: The total number of time quanta prior to sample point*

*Valid values are: 2 to 257.*

uint8\_t **NominalTqAfterSamplePoint**: 8

*NTQASP: The total number of time quanta after the sample point*

*Valid values are: 2 to 128.*

### Detailed Description

Used to setup the nominal timing parameters of the MCAN module This is a simplified struct, requiring only the prescaler value (1:x), number of time quanta before and after the sample point.

### Member Data Documentation

#### uint16\_t TCAN4x5x\_MCAN\_Nominal\_Timing\_Simple::NominalBitRatePrescaler

NBRP: The prescaler value from the MCAN system clock. Value interpreted as 1:x

Valid range is: 1 to 512.

#### uint8\_t TCAN4x5x\_MCAN\_Nominal\_Timing\_Simple::NominalTqAfterSamplePoint

NTQASP: The total number of time quanta after the sample point

Valid values are: 2 to 128.

#### uint16\_t TCAN4x5x\_MCAN\_Nominal\_Timing\_Simple::NominalTqBeforeSamplePoint

NTQBSP: The total number of time quanta prior to sample point

Valid values are: 2 to 257.

#### The documentation for this struct was generated from the following file:

C:/Alphi/PCIeMiniSoftware/include/**TCAN4x5x\_Data\_Structs.h**

## TCAN4x5x\_MCAN\_RX\_Header Struct Reference

CAN message header.

#include <TCAN4x5x\_Data\_Structs.h>

### Public Attributes

uint32\_t **ID**: 29

*CAN ID received.*

uint8\_t **RTR**: 1

*Remote Transmission Request flag.*

uint8\_t **XTD**: 1

*Extended Identifier flag.*

uint8\_t **ESI**: 1

*Error state indicator flag.*

uint16\_t **RXTS**: 16

*Receive time stamp.*

uint8\_t **DLCode**: 4

*Data length code.*

uint8\_t **BRS**: 1

*Bit rate switch used flag.*

uint8\_t **FDF**: 1

*CAN FD Format flag.*

uint8\_t **reserved**: 2

*Reserved (0)*

uint8\_t **FIDX**: 7

*Filter index that this message matched.*

uint8\_t **ANMF**: 1

*Accepted non matching frame flag.*

### Detailed Description

CAN message header.

### Member Data Documentation

#### uint8\_t TCAN4x5x\_MCAN\_RX\_Header::ANMF

Accepted non matching frame flag.

#### uint8\_t TCAN4x5x\_MCAN\_RX\_Header::BRS

Bit rate switch used flag.

#### uint8\_t TCAN4x5x\_MCAN\_RX\_Header::DLCode

Data length code.

#### uint8\_t TCAN4x5x\_MCAN\_RX\_Header::ESI

Error state indicator flag.

#### uint8\_t TCAN4x5x\_MCAN\_RX\_Header::FDF

CAN FD Format flag.

#### uint8\_t TCAN4x5x\_MCAN\_RX\_Header::FIDX

Filter index that this message matched.

#### uint32\_t TCAN4x5x\_MCAN\_RX\_Header::ID

CAN ID received.

#### uint8\_t TCAN4x5x\_MCAN\_RX\_Header::reserved

Reserved (0)

#### uint8\_t TCAN4x5x\_MCAN\_RX\_Header::RTR

Remote Transmission Request flag.

#### uint16\_t TCAN4x5x\_MCAN\_RX\_Header::RXTS

Receive time stamp.

#### uint8\_t TCAN4x5x\_MCAN\_RX\_Header::XTD

Extended Identifier flag.

#### The documentation for this struct was generated from the following file:

C:/Alphi/PCIeMiniSoftware/include/**TCAN4x5x\_Data\_Structs.h**

## TCAN4x5x\_MCAN\_SID\_Filter Struct Reference

Standard ID filter struct.

#include <TCAN4x5x\_Data\_Structs.h>

### Public Attributes

union {

  uint32\_t **word**

*full register as single 32-bit word*

  struct {

    uint16\_t **SFID2**: 11

*SFID2[10:0].*

    uint8\_t **reserved**: 5

*Reserved.*

    uint16\_t **SFID1**: 11

*SFID1[10:0].*

**TCAN4x5x\_SID\_SFEC\_Values** **SFEC**: 3

*SFEC[2:0] Standard filter element configuration.*

**TCAN4x5x\_SID\_SFT\_Values** **SFT**: 2

*SFT Standard Filter Type.*

  }

};

### Detailed Description

Standard ID filter struct.

### Member Data Documentation

#### union { ... }

#### uint8\_t TCAN4x5x\_MCAN\_SID\_Filter::reserved

Reserved.

#### TCAN4x5x\_SID\_SFEC\_Values TCAN4x5x\_MCAN\_SID\_Filter::SFEC

SFEC[2:0] Standard filter element configuration.

#### uint16\_t TCAN4x5x\_MCAN\_SID\_Filter::SFID1

SFID1[10:0].

#### uint16\_t TCAN4x5x\_MCAN\_SID\_Filter::SFID2

SFID2[10:0].

#### TCAN4x5x\_SID\_SFT\_Values TCAN4x5x\_MCAN\_SID\_Filter::SFT

SFT Standard Filter Type.

#### uint32\_t TCAN4x5x\_MCAN\_SID\_Filter::word

full register as single 32-bit word

#### The documentation for this struct was generated from the following file:

C:/Alphi/PCIeMiniSoftware/include/**TCAN4x5x\_Data\_Structs.h**

## TCAN4x5x\_MCAN\_TX\_Header Struct Reference

CAN message header for transmitted messages.

#include <TCAN4x5x\_Data\_Structs.h>

### Public Attributes

uint32\_t **ID**: 29

*CAN ID to send.*

uint8\_t **RTR**: 1

*Remote Transmission Request flag.*

uint8\_t **XTD**: 1

*Extended Identifier flag.*

uint8\_t **ESI**: 1

*Error state indicator flag.*

uint8\_t **DLCode**: 4

*Data length code.*

uint8\_t **BRS**: 1

*Bit rate switch used flag.*

uint8\_t **FDF**: 1

*CAN FD Format flag.*

uint8\_t **reserved**: 1

*Reserved.*

uint8\_t **EFC**: 1

*Event FIFO Control flag, to store tx events or not.*

uint8\_t **MM**: 8

*Message Marker, used if EFC is set to 1.*

### Detailed Description

CAN message header for transmitted messages.

### Member Data Documentation

#### uint8\_t TCAN4x5x\_MCAN\_TX\_Header::BRS

Bit rate switch used flag.

#### uint8\_t TCAN4x5x\_MCAN\_TX\_Header::DLCode

Data length code.

#### uint8\_t TCAN4x5x\_MCAN\_TX\_Header::EFC

Event FIFO Control flag, to store tx events or not.

#### uint8\_t TCAN4x5x\_MCAN\_TX\_Header::ESI

Error state indicator flag.

#### uint8\_t TCAN4x5x\_MCAN\_TX\_Header::FDF

CAN FD Format flag.

#### uint32\_t TCAN4x5x\_MCAN\_TX\_Header::ID

CAN ID to send.

#### uint8\_t TCAN4x5x\_MCAN\_TX\_Header::MM

Message Marker, used if EFC is set to 1.

#### uint8\_t TCAN4x5x\_MCAN\_TX\_Header::reserved

Reserved.

#### uint8\_t TCAN4x5x\_MCAN\_TX\_Header::RTR

Remote Transmission Request flag.

#### uint8\_t TCAN4x5x\_MCAN\_TX\_Header::XTD

Extended Identifier flag.

#### The documentation for this struct was generated from the following file:

C:/Alphi/PCIeMiniSoftware/include/**TCAN4x5x\_Data\_Structs.h**

## TCAN4x5x\_MCAN\_XID\_Filter Struct Reference

Extended ID filter struct.

#include <TCAN4x5x\_Data\_Structs.h>

### Public Attributes

uint32\_t **EFID2**: 29

*EFID2[28:0].*

uint8\_t **reserved**: 1

*Reserved.*

**TCAN4x5x\_XID\_EFT\_Values** **EFT**: 2

*EFT[1:0].*

uint32\_t **EFID1**: 29

*EFID1[28:0].*

**TCAN4x5x\_XID\_EFEC\_Values** **EFEC**: 3

*SFT Standard Filter Type.*

### Detailed Description

Extended ID filter struct.

### Member Data Documentation

#### TCAN4x5x\_XID\_EFEC\_Values TCAN4x5x\_MCAN\_XID\_Filter::EFEC

SFT Standard Filter Type.

#### uint32\_t TCAN4x5x\_MCAN\_XID\_Filter::EFID1

EFID1[28:0].

#### uint32\_t TCAN4x5x\_MCAN\_XID\_Filter::EFID2

EFID2[28:0].

#### TCAN4x5x\_XID\_EFT\_Values TCAN4x5x\_MCAN\_XID\_Filter::EFT

EFT[1:0].

#### uint8\_t TCAN4x5x\_MCAN\_XID\_Filter::reserved

Reserved.

#### The documentation for this struct was generated from the following file:

C:/Alphi/PCIeMiniSoftware/include/**TCAN4x5x\_Data\_Structs.h**

## TCAN4x5x\_MRAM\_Config Struct Reference

Defines the number of MRAM elements and the size of the elements.

#include <TCAN4x5x\_Data\_Structs.h>

### Public Attributes

uint8\_t **SIDNumElements**: 8

*Standard ID Number of Filter Elements: The number of 11-bit filters the user would like*

*Valid range is: 0 to 128.*

uint8\_t **XIDNumElements**: 7

*Extended ID Number of Filter Elements: The number of 29-bit filters the user would like*

*Valid range is: 0 to 64.*

uint8\_t **Rx0NumElements**: 7

*RX FIFO 0 number of elements: The number of elements for the RX FIFO 0*

*Valid range is: 0 to 64.*

**TCAN4x5x\_MRAM\_Element\_Data\_Size** **Rx0ElementSize**: 3

*RX FIFO 0 element size: The number of bytes for the RX 0 FIFO (data payload)*

uint8\_t **Rx1NumElements**: 7

*RX FIFO 1 number of elements: The number of elements for the RX FIFO 1*

*Valid range is: 0 to 64.*

**TCAN4x5x\_MRAM\_Element\_Data\_Size** **Rx1ElementSize**: 3

*RX FIFO 1 element size: The number of bytes for the RX 1 FIFO (data payload)*

uint8\_t **RxBufNumElements**: 7

*RX Buffers number of elements: The number of elements for the RX Buffers (Not the FIFO)*

*Valid range is: 0 to 64.*

**TCAN4x5x\_MRAM\_Element\_Data\_Size** **RxBufElementSize**: 3

*RX Buffers element size: The number of bytes for the RX Buffers (data payload), not the FIFO.*

uint8\_t **TxEventFIFONumElements**: 6

*TX Event FIFO number of elements: The number of elements for the TX Event FIFO*

*Valid range is: 0 to 32.*

uint8\_t **TxBufferNumElements**: 6

*TX Buffers number of elements: The number of elements for the TX Buffers*

*Valid range is: 0 to 32.*

**TCAN4x5x\_MRAM\_Element\_Data\_Size** **TxBufferElementSize**: 3

*TX Buffers element size: The number of bytes for the TX Buffers (data payload)*

### Detailed Description

Defines the number of MRAM elements and the size of the elements.

### Member Data Documentation

#### TCAN4x5x\_MRAM\_Element\_Data\_Size TCAN4x5x\_MRAM\_Config::Rx0ElementSize

RX FIFO 0 element size: The number of bytes for the RX 0 FIFO (data payload)

#### uint8\_t TCAN4x5x\_MRAM\_Config::Rx0NumElements

RX FIFO 0 number of elements: The number of elements for the RX FIFO 0

Valid range is: 0 to 64.

#### TCAN4x5x\_MRAM\_Element\_Data\_Size TCAN4x5x\_MRAM\_Config::Rx1ElementSize

RX FIFO 1 element size: The number of bytes for the RX 1 FIFO (data payload)

#### uint8\_t TCAN4x5x\_MRAM\_Config::Rx1NumElements

RX FIFO 1 number of elements: The number of elements for the RX FIFO 1

Valid range is: 0 to 64.

#### TCAN4x5x\_MRAM\_Element\_Data\_Size TCAN4x5x\_MRAM\_Config::RxBufElementSize

RX Buffers element size: The number of bytes for the RX Buffers (data payload), not the FIFO.

#### uint8\_t TCAN4x5x\_MRAM\_Config::RxBufNumElements

RX Buffers number of elements: The number of elements for the RX Buffers (Not the FIFO)

Valid range is: 0 to 64.

#### uint8\_t TCAN4x5x\_MRAM\_Config::SIDNumElements

Standard ID Number of Filter Elements: The number of 11-bit filters the user would like

Valid range is: 0 to 128.

#### TCAN4x5x\_MRAM\_Element\_Data\_Size TCAN4x5x\_MRAM\_Config::TxBufferElementSize

TX Buffers element size: The number of bytes for the TX Buffers (data payload)

#### uint8\_t TCAN4x5x\_MRAM\_Config::TxBufferNumElements

TX Buffers number of elements: The number of elements for the TX Buffers

Valid range is: 0 to 32.

#### uint8\_t TCAN4x5x\_MRAM\_Config::TxEventFIFONumElements

TX Event FIFO number of elements: The number of elements for the TX Event FIFO

Valid range is: 0 to 32.

#### uint8\_t TCAN4x5x\_MRAM\_Config::XIDNumElements

Extended ID Number of Filter Elements: The number of 29-bit filters the user would like

Valid range is: 0 to 64.

#### The documentation for this struct was generated from the following file:

C:/Alphi/PCIeMiniSoftware/include/**TCAN4x5x\_Data\_Structs.h**

## TcanInterface Class Reference

This class implements the **TCAN4550** SPI interface.

#include <TCAN4550.h>

Inheritance diagram for TcanInterface:

### Public Member Functions

**TcanInterface** (volatile void \*spiAddr, AlteraPio \*rst, uint8\_t nbr)

*Constructor.*

void **reset** ()

*reset the* ***TCAN4550*** *chip*

uint32\_t **getStatus** ()

*Get the SPI interface status.*

volatile void \* **getAddress** ()

*Get the address of the SPI controller.*

void **AHB\_WRITE\_32** (uint16\_t address, uint32\_t data)

void **AHB\_WRITE\_BURST\_START** (uint16\_t address, uint8\_t words)

void **AHB\_WRITE\_BURST\_WRITE** (uint32\_t data)

void **AHB\_WRITE\_BURST\_END** (void)

uint32\_t **AHB\_READ\_32** (uint16\_t address)

void **AHB\_READ\_BURST\_START** (uint16\_t address, uint8\_t words)

uint32\_t **AHB\_READ\_BURST\_READ** (void)

void **AHB\_READ\_BURST\_END** (void)

### Public Attributes

uint8\_t **slave**

***TCAN4550*** *chip select index.*

AlteraPio \* **rstPio**

### Detailed Description

This class implements the **TCAN4550** SPI interface.

### Constructor & Destructor Documentation

#### TcanInterface::TcanInterface (volatile void \* *spiAddr*, AlteraPio \* *rst*, uint8\_t *nbr*)[inline]

Constructor.

##### Parameters

|  |  |
| --- | --- |
| *spiAddr* | Address of the SPI controller |
| *rst* | Address of the PIO controlling the reset line |
| *nbr* | Chip select line number. |

### Member Function Documentation

#### uint32\_t TcanInterface::AHB\_READ\_32 (uint16\_t *address*)

#### void TcanInterface::AHB\_READ\_BURST\_END (void )

#### uint32\_t TcanInterface::AHB\_READ\_BURST\_READ (void )

#### void TcanInterface::AHB\_READ\_BURST\_START (uint16\_t *address*, uint8\_t *words*)

#### void TcanInterface::AHB\_WRITE\_32 (uint16\_t *address*, uint32\_t *data*)

#### void TcanInterface::AHB\_WRITE\_BURST\_END (void )

#### void TcanInterface::AHB\_WRITE\_BURST\_START (uint16\_t *address*, uint8\_t *words*)

#### void TcanInterface::AHB\_WRITE\_BURST\_WRITE (uint32\_t *data*)

#### volatile void\* TcanInterface::getAddress ()[inline]

Get the address of the SPI controller.

##### Return values

|  |  |
| --- | --- |
| *Address* | in user space. |

#### uint32\_t TcanInterface::getStatus ()[inline]

Get the SPI interface status.

The bits are defined by the **AlteraSpi** class.

##### Return values

|  |  |
| --- | --- |
| *Status* | register content. |

#### void TcanInterface::reset ()[inline]

reset the **TCAN4550** chip

### Member Data Documentation

#### AlteraPio\* TcanInterface::rstPio

#### uint8\_t TcanInterface::slave

**TCAN4550** chip select index.

#### The documentation for this class was generated from the following files:

C:/Alphi/PCIeMiniSoftware/include/**TCAN4550.h**

**TCAN4x5x\_SPI.cpp**

## TransferDesc Class Reference

Structure containing the details of a DMA transaction.

#include <AlteraDma.h>

### Public Attributes

volatile uint32\_t **dest\_offset**

*32-bit FPGA Avalon bus address of destination*

volatile uint32\_t **src\_offset**

*32-bit FPGA Avalon bus address of source*

uint32\_t **tfr\_length**

*length of the transfer*

uint32\_t **flags**

uint32\_t **txs\_offset**

*Offset in the mapping area.*

uint32\_t **bufLength**

*Size of the buffer in bytes.*

uint32\_t \* **userSpaceBuffer**

*PC buffer address as an user-space address.*

bool **fPolling**

*Polling or interrupt for end of transfer (not yet implemented)*

### Detailed Description

Structure containing the details of a DMA transaction.

### Member Data Documentation

#### uint32\_t TransferDesc::bufLength

Size of the buffer in bytes.

#### volatile uint32\_t TransferDesc::dest\_offset

32-bit FPGA Avalon bus address of destination

#### uint32\_t TransferDesc::flags

#### bool TransferDesc::fPolling

Polling or interrupt for end of transfer (not yet implemented)

#### volatile uint32\_t TransferDesc::src\_offset

32-bit FPGA Avalon bus address of source

#### uint32\_t TransferDesc::tfr\_length

length of the transfer

#### uint32\_t TransferDesc::txs\_offset

Offset in the mapping area.

#### uint32\_t\* TransferDesc::userSpaceBuffer

PC buffer address as an user-space address.

#### The documentation for this class was generated from the following file:

C:/Alphi/PCIeMiniSoftware/include/**AlteraDma.h**

# File Documentation

## C:/Alphi/PCIeMiniSoftware/include/AlphiBoard.h File Reference

Base PCIe board class with Jungo driver and Altera PCIe hardware.

#include <stdint.h>

#include <stdio.h>

#include <Windows.h>

#include "AlphiDll.h"

#include <wdc\_defs.h>

#include "PcieCra.h"

#include "AlteraDma.h"

### Classes

struct **MINIPCIE\_DEV\_CTX**

*Minipcie Device Information Structure.*

struct **LinearAddress**

*Memory Segment Descriptor.*

struct **MINIPCIE\_INT\_RESULT**

*Interrupt result information structure.*

class **BoardVersion**

*Board Hardware identification and version.*

class **AlphiBoard**

*Base class implementing a PCIe board and the Jungo driver.*

### Macros

#define **ErrLog**  printf

#define **TraceLog**  printf

#define **CLOCK\_REALTIME**  0

### Typedefs

typedef void(\* **MINIPCIE\_INT\_HANDLER**) (void \*pIntData)

*minipcie\_arinc429 diagnostics interrupt handler function type*

typedef void(\* **MINIPCIE\_EVENT\_HANDLER**) (WDC\_DEVICE\_HANDLE hDev, DWORD dwAction)

*minipcie\_arinc429 diagnostics plug-and-play and power management events handler function type*

typedef struct **MINIPCIE\_DEV\_CTX** \* **PMINIPCIE\_DEV\_CTX**

typedef struct **LinearAddress** **LinearAddress**

*Memory Segment Descriptor.*

### Functions

static int **clock\_gettime** (int, struct timespec \*spec)

*get system time with a nanosecond definition*

static void **usleep** (\_\_int64 usec)

*Sleep for a number of microseconds.*

static void **nanosleep** (struct timespec \*t, void \*na)

### Detailed Description

Base PCIe board class with Jungo driver and Altera PCIe hardware.

### Macro Definition Documentation

#### #define CLOCK\_REALTIME  0

#### #define ErrLog  printf

#### #define TraceLog  printf

### Typedef Documentation

#### typedef struct LinearAddress LinearAddress

Memory Segment Descriptor.

#### typedef void(\* MINIPCIE\_EVENT\_HANDLER) (WDC\_DEVICE\_HANDLE hDev, DWORD dwAction)

minipcie\_arinc429 diagnostics plug-and-play and power management events handler function type

#### typedef void(\* MINIPCIE\_INT\_HANDLER) (void \*pIntData)

minipcie\_arinc429 diagnostics interrupt handler function type

#### typedef struct MINIPCIE\_DEV\_CTX \* PMINIPCIE\_DEV\_CTX

### Function Documentation

#### static int clock\_gettime (int , struct timespec \* *spec*)[inline], [static]

get system time with a nanosecond definition

#### static void nanosleep (struct timespec \* *t*, void \* *na*)[inline], [static]

#### static void usleep (\_\_int64 *usec*)[inline], [static]

Sleep for a number of microseconds.

##### Parameters

|  |  |
| --- | --- |
| *usec* | Number of microseconds to sleep |

## C:/Alphi/PCIeMiniSoftware/include/AlphiDll.h File Reference

Utility DLL definitions.

#include "time.h"

### Macros

#define **LINUX**

#define **UNIX**

#define **FALSE**  0

#define **TRUE**  1

#define **ALPHI\_S\_OK**  0

#define **Dll**  \_\_declspec( dllimport )

### Typedefs

typedef int **HRESULT**

### Detailed Description

Utility DLL definitions.

### Macro Definition Documentation

#### #define ALPHI\_S\_OK  0

#### #define Dll  \_\_declspec( dllimport )

#### #define FALSE  0

#### #define LINUX

#### #define TRUE  1

#### #define UNIX

### Typedef Documentation

#### typedef int HRESULT

#### 

## C:/Alphi/PCIeMiniSoftware/include/AlphiErrorCodes.h File Reference

Description of the Error Codes used by the libraries.

#include "AlphiDll.h"

### Macros

#define **ERRCODE\_NO\_ERROR**  0

#define **ERRCODE\_SUCCESS**  0

#define **ERRCODE\_INTERNAL\_ERROR**  1

#define **ERRCODE\_INVALID\_BOARD\_NUM**  **ERRCODE\_INTERNAL\_ERROR** + 1

#define **ERRCODE\_BOARD\_NOT\_PRESENT**  **ERRCODE\_INTERNAL\_ERROR** + 1

#define **ERRCODE\_INVALID\_HANDLE**  **ERRCODE\_INVALID\_BOARD\_NUM** + 1

#define **ERRCODE\_INVALID\_CHANNEL\_NUM**  **ERRCODE\_INVALID\_HANDLE** + 1

#define **ERRCODE\_INVALID\_SELF\_TEST\_ENABLE\_VAL**  **ERRCODE\_INVALID\_CHANNEL\_NUM** + 1

#define **ERRCODE\_INVALID\_VALUE**  **ERRCODE\_INVALID\_SELF\_TEST\_ENABLE\_VAL** + 1

#define **ERRCODE\_INPUT\_MODE**  **ERRCODE\_INVALID\_VALUE** + 1

#define **ERRCODE\_OUTPUT\_MODE**  **ERRCODE\_INPUT\_MODE** + 1

#define **ERRCODE\_INVALID\_LOGIC\_SEL**  **ERRCODE\_OUTPUT\_MODE** + 1

#define **ERRCODE\_INVALID\_STROBE\_MODE**  **ERRCODE\_INVALID\_LOGIC\_SEL** + 1

#define **ERRCODE\_INVALID\_GROUP**  **ERRCODE\_INVALID\_STROBE\_MODE** + 1

#define **ERRCODE\_INVALID\_FREQUENCY**  **ERRCODE\_INVALID\_GROUP** + 1

#define **ERRCODE\_INVALID\_INPUT\_MODE**  **ERRCODE\_INVALID\_FREQUENCY** + 1

#define **ERRCODE\_INVALID\_MASK\_VALUE**  **ERRCODE\_INVALID\_INPUT\_MODE** + 1

#define **ERRCODE\_INVALID\_MODE**  **ERRCODE\_INVALID\_MASK\_VALUE** + 1

#define **ERRCODE\_INVALID\_SELF\_TEST\_DATA**  **ERRCODE\_INVALID\_MODE** + 1

#define **ERRCODE\_SELF\_TEST\_DISABLE**  **ERRCODE\_INVALID\_SELF\_TEST\_DATA** + 1

#define **ERRCODE\_FAILED\_SELF\_TEST**  **ERRCODE\_SELF\_TEST\_DISABLE** + 1

#define **ERRCODE\_INVALID\_TIME\_BOUNCE\_VAL**  **ERRCODE\_FAILED\_SELF\_TEST** + 1

#define **ERRCODE\_INT\_ALREADY\_ENABLED**  **ERRCODE\_INVALID\_TIME\_BOUNCE\_VAL** + 1

#define **ERRCODE\_INT\_NOT\_ENABLED**  **ERRCODE\_INT\_ALREADY\_ENABLED** + 1

#define **ERRCODE\_INVALID\_TRANRECVSTS**  **ERRCODE\_INT\_NOT\_ENABLED** + 1

#define **ERRCODE\_INVALID\_TRANSNUMBER**  **ERRCODE\_INVALID\_TRANRECVSTS** + 1

#define **ERRCODE\_INVALID\_RECVNUMBER**  **ERRCODE\_INVALID\_TRANSNUMBER** + 1

#define **ERRCODE\_INVALID\_RECVSTATUS**  **ERRCODE\_INVALID\_RECVNUMBER**+1

#define **ERRCODE\_INVALID\_TRPAIR**  **ERRCODE\_INVALID\_RECVSTATUS** + 1

#define **ERRCODE\_INVALID\_TRANSSTATUS**  **ERRCODE\_INVALID\_TRPAIR** + 1

#define **ERRCODE\_INVALID\_LABELNUMBER**  **ERRCODE\_INVALID\_TRANSSTATUS** + 1

#define **ERRCODE\_INVALID\_SDI**  **ERRCODE\_INVALID\_LABELNUMBER** + 1

#define **ERRCODE\_INVALID\_PARITY**  **ERRCODE\_INVALID\_SDI** + 1

#define **ERRCODE\_INVALID\_DATARATE**  **ERRCODE\_INVALID\_PARITY** + 1

#define **ERRCODE\_INVALID\_DATALENGTH**  **ERRCODE\_INVALID\_DATARATE** + 1

#define **ERRCODE\_INVALID\_SSM**  **ERRCODE\_INVALID\_DATALENGTH** + 1

#define **ERRCODE\_INVALID\_EMTINTSTS**  **ERRCODE\_INVALID\_SSM** + 1

#define **ERRCODE\_INVALID\_HFINTSTS**  **ERRCODE\_INVALID\_EMTINTSTS** + 1

#define **ERRCODE\_INVALID\_FINTSTS**  **ERRCODE\_INVALID\_HFINTSTS** + 1

#define **ERRCODE\_INVALID\_COMMAND**  **ERRCODE\_INVALID\_FINTSTS** + 1

#define **ERRCODE\_RX\_UNDERFLOW**  **ERRCODE\_INVALID\_COMMAND** + 1

#define **ERRCODE\_TX\_OVERFLOW**  **ERRCODE\_RX\_UNDERFLOW** + 1

#define **ERRCODE\_BUSY**  **ERRCODE\_TX\_OVERFLOW** + 1

#define **ERRCODE\_TIMEOUT**  **ERRCODE\_BUSY** + 1

#define **ERRCODE\_INVALID\_ALIGNMENT**  **ERRCODE\_TIMEOUT** + 1

### Typedefs

typedef int **PCIeMini\_status**

### Functions

DLL char \* **getAlphiErrorMsg** (**PCIeMini\_status** errCode)

*Gives the string description of an error code.*

### Detailed Description

Description of the Error Codes used by the libraries.

### Macro Definition Documentation

#### #define ERRCODE\_BOARD\_NOT\_PRESENT  ERRCODE\_INTERNAL\_ERROR + 1

#### #define ERRCODE\_BUSY  ERRCODE\_TX\_OVERFLOW + 1

#### #define ERRCODE\_FAILED\_SELF\_TEST  ERRCODE\_SELF\_TEST\_DISABLE + 1

#### #define ERRCODE\_INPUT\_MODE  ERRCODE\_INVALID\_VALUE + 1

#### #define ERRCODE\_INT\_ALREADY\_ENABLED  ERRCODE\_INVALID\_TIME\_BOUNCE\_VAL + 1

#### #define ERRCODE\_INT\_NOT\_ENABLED  ERRCODE\_INT\_ALREADY\_ENABLED + 1

#### #define ERRCODE\_INTERNAL\_ERROR  1

#### #define ERRCODE\_INVALID\_ALIGNMENT  ERRCODE\_TIMEOUT + 1

#### #define ERRCODE\_INVALID\_BOARD\_NUM  ERRCODE\_INTERNAL\_ERROR + 1

#### #define ERRCODE\_INVALID\_CHANNEL\_NUM  ERRCODE\_INVALID\_HANDLE + 1

#### #define ERRCODE\_INVALID\_COMMAND  ERRCODE\_INVALID\_FINTSTS + 1

#### #define ERRCODE\_INVALID\_DATALENGTH  ERRCODE\_INVALID\_DATARATE + 1

#### #define ERRCODE\_INVALID\_DATARATE  ERRCODE\_INVALID\_PARITY + 1

#### #define ERRCODE\_INVALID\_EMTINTSTS  ERRCODE\_INVALID\_SSM + 1

#### #define ERRCODE\_INVALID\_FINTSTS  ERRCODE\_INVALID\_HFINTSTS + 1

#### #define ERRCODE\_INVALID\_FREQUENCY  ERRCODE\_INVALID\_GROUP + 1

#### #define ERRCODE\_INVALID\_GROUP  ERRCODE\_INVALID\_STROBE\_MODE + 1

#### #define ERRCODE\_INVALID\_HANDLE  ERRCODE\_INVALID\_BOARD\_NUM + 1

#### #define ERRCODE\_INVALID\_HFINTSTS  ERRCODE\_INVALID\_EMTINTSTS + 1

#### #define ERRCODE\_INVALID\_INPUT\_MODE  ERRCODE\_INVALID\_FREQUENCY + 1

#### #define ERRCODE\_INVALID\_LABELNUMBER  ERRCODE\_INVALID\_TRANSSTATUS + 1

#### #define ERRCODE\_INVALID\_LOGIC\_SEL  ERRCODE\_OUTPUT\_MODE + 1

#### #define ERRCODE\_INVALID\_MASK\_VALUE  ERRCODE\_INVALID\_INPUT\_MODE + 1

#### #define ERRCODE\_INVALID\_MODE  ERRCODE\_INVALID\_MASK\_VALUE + 1

#### #define ERRCODE\_INVALID\_PARITY  ERRCODE\_INVALID\_SDI + 1

#### #define ERRCODE\_INVALID\_RECVNUMBER  ERRCODE\_INVALID\_TRANSNUMBER + 1

#### #define ERRCODE\_INVALID\_RECVSTATUS  ERRCODE\_INVALID\_RECVNUMBER+1

#### #define ERRCODE\_INVALID\_SDI  ERRCODE\_INVALID\_LABELNUMBER + 1

#### #define ERRCODE\_INVALID\_SELF\_TEST\_DATA  ERRCODE\_INVALID\_MODE + 1

#### #define ERRCODE\_INVALID\_SELF\_TEST\_ENABLE\_VAL  ERRCODE\_INVALID\_CHANNEL\_NUM + 1

#### #define ERRCODE\_INVALID\_SSM  ERRCODE\_INVALID\_DATALENGTH + 1

#### #define ERRCODE\_INVALID\_STROBE\_MODE  ERRCODE\_INVALID\_LOGIC\_SEL + 1

#### #define ERRCODE\_INVALID\_TIME\_BOUNCE\_VAL  ERRCODE\_FAILED\_SELF\_TEST + 1

#### #define ERRCODE\_INVALID\_TRANRECVSTS  ERRCODE\_INT\_NOT\_ENABLED + 1

#### #define ERRCODE\_INVALID\_TRANSNUMBER  ERRCODE\_INVALID\_TRANRECVSTS + 1

#### #define ERRCODE\_INVALID\_TRANSSTATUS  ERRCODE\_INVALID\_TRPAIR + 1

#### #define ERRCODE\_INVALID\_TRPAIR  ERRCODE\_INVALID\_RECVSTATUS + 1

#### #define ERRCODE\_INVALID\_VALUE  ERRCODE\_INVALID\_SELF\_TEST\_ENABLE\_VAL + 1

#### #define ERRCODE\_NO\_ERROR  0

#### #define ERRCODE\_OUTPUT\_MODE  ERRCODE\_INPUT\_MODE + 1

#### #define ERRCODE\_RX\_UNDERFLOW  ERRCODE\_INVALID\_COMMAND + 1

#### #define ERRCODE\_SELF\_TEST\_DISABLE  ERRCODE\_INVALID\_SELF\_TEST\_DATA + 1

#### #define ERRCODE\_SUCCESS  0

#### #define ERRCODE\_TIMEOUT  ERRCODE\_BUSY + 1

#### #define ERRCODE\_TX\_OVERFLOW  ERRCODE\_RX\_UNDERFLOW + 1

### Typedef Documentation

#### typedef int PCIeMini\_status

### Function Documentation

#### DLL char\* getAlphiErrorMsg (PCIeMini\_status *errCode*)

Gives the string description of an error code.

##### Parameters

|  |  |
| --- | --- |
| *errCode* | Error code returned by a function |

##### Return values

|  |  |
| --- | --- |
| *C-string,description* | of the error |

## C:/Alphi/PCIeMiniSoftware/include/AlteraDma.h File Reference

Description of the low-level access routines to the SPI.

#include <stddef.h>

#include <stdio.h>

#include <stdint.h>

#include "AlphiDll.h"

### Classes

class **TransferDesc**

*Structure containing the details of a DMA transaction.*

class **AlteraDma**

*Low level SPI interface to the SPI hardware.*

### Macros

#define **ALT\_AVALON\_DMA\_MODE\_MSK**  (0xf)

#define **ALT\_AVALON\_DMA\_MODE\_8**  (0x0)

#define **ALT\_AVALON\_DMA\_MODE\_16**  (0x1)

#define **ALT\_AVALON\_DMA\_MODE\_32**  (0x3)

#define **ALT\_AVALON\_DMA\_MODE\_64**  (0x7)

#define **ALT\_AVALON\_DMA\_MODE\_128**  (0xf)

#define **ALT\_AVALON\_DMA\_TX\_STREAM**  (0x20)

#define **ALT\_AVALON\_DMA\_RX\_STREAM**  (0x40)

#define **ALT\_DMA\_TX\_STREAM\_ON**  (0x1)

#define **ALT\_DMA\_TX\_STREAM\_OFF**  (0x2)

#define **ALT\_DMA\_RX\_STREAM\_ON**  (0x3)

#define **ALT\_DMA\_RX\_STREAM\_OFF**  (0x4)

#define **ALT\_DMA\_SET\_MODE\_8**  (0x5)

*Transfer data in units of 8 bits.*

#define **ALT\_DMA\_SET\_MODE\_16**  (0x6)

*Transfer data in units of 16 bits.*

#define **ALT\_DMA\_SET\_MODE\_32**  (0x7)

*Transfer data in units of 32 bits.*

#define **ALT\_DMA\_SET\_MODE\_64**  (0x8)

*Transfer data in units of 64 bits.*

#define **ALT\_DMA\_SET\_MODE\_128**  (0x9)

*Transfer data in units of 128 bits.*

#define **ALT\_DMA\_GET\_MODE**  (0xa)

*Get the current transfer mode.*

#define **ALTERA\_AVALON\_DMA\_CONTROL\_BYTE\_MSK**  (0x1)

#define **ALTERA\_AVALON\_DMA\_CONTROL\_HW\_MSK**  (0x2)

#define **ALTERA\_AVALON\_DMA\_CONTROL\_WORD\_MSK**  (0x4)

#define **ALTERA\_AVALON\_DMA\_CONTROL\_GO\_MSK**  (0x8)

#define **ALTERA\_AVALON\_DMA\_CONTROL\_I\_EN\_MSK**  (0x10)

#define **ALTERA\_AVALON\_DMA\_CONTROL\_REEN\_MSK**  (0x20)

#define **ALTERA\_AVALON\_DMA\_CONTROL\_WEEN\_MSK**  (0x40)

#define **ALTERA\_AVALON\_DMA\_CONTROL\_LEEN\_MSK**  (0x80)

#define **ALTERA\_AVALON\_DMA\_CONTROL\_RCON\_MSK**  (0x100)

#define **ALTERA\_AVALON\_DMA\_CONTROL\_WCON\_MSK**  (0x200)

#define **ALTERA\_AVALON\_DMA\_CONTROL\_DWORD\_MSK**  (0x400)

#define **ALTERA\_AVALON\_DMA\_CONTROL\_QWORD\_MSK**  (0x800)

#define **ALTERA\_AVALON\_DMA\_CONTROL\_SOFTWARERESET\_MSK**  (0x1000)

#define **ALTERA\_AVALON\_DMA\_STATUS\_DONE\_MSK**  (0x1)

#define **ALTERA\_AVALON\_DMA\_STATUS\_BUSY\_MSK**  (0x2)

#define **ALTERA\_AVALON\_DMA\_STATUS\_REOP\_MSK**  (0x4)

#define **ALTERA\_AVALON\_DMA\_STATUS\_WEOP\_MSK**  (0x8)

#define **ALTERA\_AVALON\_DMA\_STATUS\_LEN\_MSK**  (0x10)

#define **ALT\_AVALON\_DMA\_NSLOTS**  (4)

#define **ALT\_AVALON\_DMA\_NSLOTS\_MSK**  (**ALT\_AVALON\_DMA\_NSLOTS** - 1)

### Typedefs

typedef void() **alt\_txchan\_done**(void \*handle)

typedef void() **alt\_rxchan\_done**(void \*handle, void \*data)

### Detailed Description

Description of the low-level access routines to the SPI.

### Macro Definition Documentation

#### #define ALT\_AVALON\_DMA\_MODE\_128  (0xf)

#### #define ALT\_AVALON\_DMA\_MODE\_16  (0x1)

#### #define ALT\_AVALON\_DMA\_MODE\_32  (0x3)

#### #define ALT\_AVALON\_DMA\_MODE\_64  (0x7)

#### #define ALT\_AVALON\_DMA\_MODE\_8  (0x0)

#### #define ALT\_AVALON\_DMA\_MODE\_MSK  (0xf)

#### #define ALT\_AVALON\_DMA\_NSLOTS  (4)

#### #define ALT\_AVALON\_DMA\_NSLOTS\_MSK  (ALT\_AVALON\_DMA\_NSLOTS - 1)

#### #define ALT\_AVALON\_DMA\_RX\_STREAM  (0x40)

#### #define ALT\_AVALON\_DMA\_TX\_STREAM  (0x20)

#### #define ALT\_DMA\_GET\_MODE  (0xa)

Get the current transfer mode.

#### #define ALT\_DMA\_RX\_STREAM\_OFF  (0x4)

#### #define ALT\_DMA\_RX\_STREAM\_ON  (0x3)

#### #define ALT\_DMA\_SET\_MODE\_128  (0x9)

Transfer data in units of 128 bits.

#### #define ALT\_DMA\_SET\_MODE\_16  (0x6)

Transfer data in units of 16 bits.

#### #define ALT\_DMA\_SET\_MODE\_32  (0x7)

Transfer data in units of 32 bits.

#### #define ALT\_DMA\_SET\_MODE\_64  (0x8)

Transfer data in units of 64 bits.

#### #define ALT\_DMA\_SET\_MODE\_8  (0x5)

Transfer data in units of 8 bits.

#### #define ALT\_DMA\_TX\_STREAM\_OFF  (0x2)

#### #define ALT\_DMA\_TX\_STREAM\_ON  (0x1)

#### #define ALTERA\_AVALON\_DMA\_CONTROL\_BYTE\_MSK  (0x1)

#### #define ALTERA\_AVALON\_DMA\_CONTROL\_DWORD\_MSK  (0x400)

#### #define ALTERA\_AVALON\_DMA\_CONTROL\_GO\_MSK  (0x8)

#### #define ALTERA\_AVALON\_DMA\_CONTROL\_HW\_MSK  (0x2)

#### #define ALTERA\_AVALON\_DMA\_CONTROL\_I\_EN\_MSK  (0x10)

#### #define ALTERA\_AVALON\_DMA\_CONTROL\_LEEN\_MSK  (0x80)

#### #define ALTERA\_AVALON\_DMA\_CONTROL\_QWORD\_MSK  (0x800)

#### #define ALTERA\_AVALON\_DMA\_CONTROL\_RCON\_MSK  (0x100)

#### #define ALTERA\_AVALON\_DMA\_CONTROL\_REEN\_MSK  (0x20)

#### #define ALTERA\_AVALON\_DMA\_CONTROL\_SOFTWARERESET\_MSK  (0x1000)

#### #define ALTERA\_AVALON\_DMA\_CONTROL\_WCON\_MSK  (0x200)

#### #define ALTERA\_AVALON\_DMA\_CONTROL\_WEEN\_MSK  (0x40)

#### #define ALTERA\_AVALON\_DMA\_CONTROL\_WORD\_MSK  (0x4)

#### #define ALTERA\_AVALON\_DMA\_STATUS\_BUSY\_MSK  (0x2)

#### #define ALTERA\_AVALON\_DMA\_STATUS\_DONE\_MSK  (0x1)

#### #define ALTERA\_AVALON\_DMA\_STATUS\_LEN\_MSK  (0x10)

#### #define ALTERA\_AVALON\_DMA\_STATUS\_REOP\_MSK  (0x4)

#### #define ALTERA\_AVALON\_DMA\_STATUS\_WEOP\_MSK  (0x8)

### Typedef Documentation

#### typedef void() alt\_rxchan\_done(void \*handle, void \*data)

#### typedef void() alt\_txchan\_done(void \*handle)

#### 

## C:/Alphi/PCIeMiniSoftware/include/AlteraSpi.h File Reference

Description of the low-level access routines to the SPI.

#include <stddef.h>

#include <stdint.h>

#include "AlphiDll.h"

### Classes

class **AlteraSpi**

*Low level SPI interface to the SPI hardware.*

### Macros

#define **ALT\_AVALON\_SPI\_COMMAND\_MERGE**  (0x01)

#define **ALT\_AVALON\_SPI\_COMMAND\_TOGGLE\_SS\_N**  (0x02)

*option: toggle the SS line between words*

#define **ALTERA\_AVALON\_SPI\_STATUS\_E\_MSK**  (0x100)

#define **ALTERA\_AVALON\_SPI\_STATUS\_E\_OFST**  (8)

#define **ALTERA\_AVALON\_SPI\_CONTROL\_IROE\_MSK**  (0x8)

#define **ALTERA\_AVALON\_SPI\_CONTROL\_IROE\_OFST**  (3)

#define **ALTERA\_AVALON\_SPI\_CONTROL\_ITOE\_MSK**  (0x10)

#define **ALTERA\_AVALON\_SPI\_CONTROL\_ITOE\_OFST**  (4)

#define **ALTERA\_AVALON\_SPI\_CONTROL\_ITRDY\_MSK**  (0x40)

#define **ALTERA\_AVALON\_SPI\_CONTROL\_ITRDY\_OFS**  (6)

#define **ALTERA\_AVALON\_SPI\_CONTROL\_IRRDY\_MSK**  (0x80)

#define **ALTERA\_AVALON\_SPI\_CONTROL\_IRRDY\_OFS**  (7)

#define **ALTERA\_AVALON\_SPI\_CONTROL\_IE\_MSK**  (0x100)

#define **ALTERA\_AVALON\_SPI\_CONTROL\_IE\_OFST**  (8)

#define **ALTERA\_AVALON\_SPI\_CONTROL\_SSO\_MSK**  (0x400)

#define **ALTERA\_AVALON\_SPI\_CONTROL\_SSO\_OFST**  (10)

### Detailed Description

Description of the low-level access routines to the SPI.

### Macro Definition Documentation

#### #define ALT\_AVALON\_SPI\_COMMAND\_MERGE  (0x01)

#### #define ALT\_AVALON\_SPI\_COMMAND\_TOGGLE\_SS\_N  (0x02)

option: toggle the SS line between words

If you need the slave select line to be toggled between words then you should set the toggle bit in the flag.

#### #define ALTERA\_AVALON\_SPI\_CONTROL\_IE\_MSK  (0x100)

#### #define ALTERA\_AVALON\_SPI\_CONTROL\_IE\_OFST  (8)

#### #define ALTERA\_AVALON\_SPI\_CONTROL\_IROE\_MSK  (0x8)

#### #define ALTERA\_AVALON\_SPI\_CONTROL\_IROE\_OFST  (3)

#### #define ALTERA\_AVALON\_SPI\_CONTROL\_IRRDY\_MSK  (0x80)

#### #define ALTERA\_AVALON\_SPI\_CONTROL\_IRRDY\_OFS  (7)

#### #define ALTERA\_AVALON\_SPI\_CONTROL\_ITOE\_MSK  (0x10)

#### #define ALTERA\_AVALON\_SPI\_CONTROL\_ITOE\_OFST  (4)

#### #define ALTERA\_AVALON\_SPI\_CONTROL\_ITRDY\_MSK  (0x40)

#### #define ALTERA\_AVALON\_SPI\_CONTROL\_ITRDY\_OFS  (6)

#### #define ALTERA\_AVALON\_SPI\_CONTROL\_SSO\_MSK  (0x400)

#### #define ALTERA\_AVALON\_SPI\_CONTROL\_SSO\_OFST  (10)

#### #define ALTERA\_AVALON\_SPI\_STATUS\_E\_MSK  (0x100)

#### #define ALTERA\_AVALON\_SPI\_STATUS\_E\_OFST  (8)

#### 

## C:/Alphi/PCIeMiniSoftware/include/IrigDecoder.h File Reference

Irig Decoder class to get time.

#include "AlphiDll.h"

#include <stdint.h>

#include <ctime>

### Classes

class **IrigDecoder**

struct **IrigDecoder::IrigDate**

### Detailed Description

Irig Decoder class to get time.

## C:/Alphi/PCIeMiniSoftware/include/ParallelInput.h File Reference

Description of the Alphi Parallel Input class.

#include "AlphiDll.h"

#include <stdint.h>

#include "AlphiErrorCodes.h"

### Classes

class **ParallelInput**

*Alphi Avalon Pio controller class.*

### Detailed Description

Description of the Alphi Parallel Input class.

## C:/Alphi/PCIeMiniSoftware/include/PcieCra.h File Reference

PCIe interface CRA class.

#include <stdint.h>

#include <stdio.h>

#include <Windows.h>

#include "AlphiDll.h"

#include "AlphiErrorCodes.h"

### Classes

class **PcieCra**

*PCIe CRA module controller class.*

### Detailed Description

PCIe interface CRA class.

## C:/Alphi/PCIeMiniSoftware/include/PCIeMini\_CAN\_FD.h File Reference

Definitition of the **PCIeMini\_CAN\_FD** board class.

#include <stdint.h>

#include <iostream>

#include "AlphiDll.h"

#include "AlphiBoard.h"

#include "AlphiErrorCodes.h"

#include "TCAN4550.h"

#include "IrigDecoder.h"

#include "AlteraPio.h"

#include "ParallelInput.h"

#include "CanFdNiosComm.h"

#include "AlteraDma.h"

#include "PcieCra.h"

### Classes

class **PCIeMini\_CAN\_FD**

***PCIeMini\_CAN\_FD*** *controller board object.*

### Macros

#define **PI\_nINT\_0**  0x0001

#define **PI\_nWAKE\_0**  0x0002

#define **PI\_GPIO1\_0**  0x0004

#define **PI\_GPO2\_0**  0x0008

#define **PI\_nINT\_1**  0x0010

#define **PI\_nWAKE\_1**  0x0020

#define **PI\_GPIO1\_1**  0x0040

#define **PI\_GPO2\_1**  0x0080

#define **PI\_nINT\_2**  0x0100

#define **PI\_nWAKE\_2**  0x0200

#define **PI\_GPIO1\_2**  0x0400

#define **PI\_GPO2\_2**  0x0800

#define **PI\_nINT\_3**  0x1000

#define **PI\_nWAKE\_3**  0x2000

#define **PI\_GPIO1\_3**  0x4000

#define **PI\_GPO2\_3**  0x8000

### Detailed Description

Definitition of the **PCIeMini\_CAN\_FD** board class.

### Macro Definition Documentation

#### #define PI\_GPIO1\_0  0x0004

#### #define PI\_GPIO1\_1  0x0040

#### #define PI\_GPIO1\_2  0x0400

#### #define PI\_GPIO1\_3  0x4000

#### #define PI\_GPO2\_0  0x0008

#### #define PI\_GPO2\_1  0x0080

#### #define PI\_GPO2\_2  0x0800

#### #define PI\_GPO2\_3  0x8000

#### #define PI\_nINT\_0  0x0001

#### #define PI\_nINT\_1  0x0010

#### #define PI\_nINT\_2  0x0100

#### #define PI\_nINT\_3  0x1000

#### #define PI\_nWAKE\_0  0x0002

#### #define PI\_nWAKE\_1  0x0020

#### #define PI\_nWAKE\_2  0x0200

#### #define PI\_nWAKE\_3  0x2000

#### 

## C:/Alphi/PCIeMiniSoftware/include/TCAN4550.h File Reference

This file contains **TCAN4550** functions.

#include "eusci\_b\_spi.h"

#include "TCAN4x5x\_SPI.h"

#include "TCAN4x5x\_Reg.h"

#include "TCAN4x5x\_Data\_Structs.h"

#include "AlteraPio.h"

#include "ParallelInput.h"

### Classes

class **TcanInterface**

*This class implements the* ***TCAN4550*** *SPI interface.*

class **TCAN4550**

### Macros

#define **TCAN4x5x\_MCAN\_VERIFY\_CONFIGURATION\_WRITES**

*If TCAN4x5x\_MCAN\_VERIFY\_CONFIGURATION\_WRITES is defined, then each MCAN configuration write will be read and verified for correctness.*

#define **TCAN4x5x\_DEVICE\_VERIFY\_CONFIGURATION\_WRITES**

*If TCAN4x5x\_DEVICE\_VERIFY\_CONFIGURATION\_WRITES is defined, then each device configuration write will be read and verified for correctness.*

### Enumerations

enum **TCAN4x5x\_MCAN\_FIFO\_Enum** { **RXFIFO0**, **RXFIFO1** }

enum **TCAN4x5x\_WDT\_Timer\_Enum** { **TCAN4x5x\_WDT\_60MS**, **TCAN4x5x\_WDT\_600MS**, **TCAN4x5x\_WDT\_3S**, **TCAN4x5x\_WDT\_6S** }

enum **TCAN4x5x\_Device\_Test\_Mode\_Enum** { **TCAN4x5x\_DEVICE\_TEST\_MODE\_NORMAL**, **TCAN4x5x\_DEVICE\_TEST\_MODE\_PHY**, **TCAN4x5x\_DEVICE\_TEST\_MODE\_CONTROLLER** }

enum **TCAN4x5x\_Device\_Mode\_Enum** { **TCAN4x5x\_DEVICE\_MODE\_NORMAL**, **TCAN4x5x\_DEVICE\_MODE\_STANDBY**, **TCAN4x5x\_DEVICE\_MODE\_SLEEP** }

### Detailed Description

This file contains **TCAN4550** functions.

It relies on the TCAN4x5x\_SPI abstraction functions Additional Feature Sets of **TCAN4550** vs TCAN4x5x:

Watchdog Timer Functions

### Macro Definition Documentation

#### #define TCAN4x5x\_DEVICE\_VERIFY\_CONFIGURATION\_WRITES

If TCAN4x5x\_DEVICE\_VERIFY\_CONFIGURATION\_WRITES is defined, then each device configuration write will be read and verified for correctness.

#### #define TCAN4x5x\_MCAN\_VERIFY\_CONFIGURATION\_WRITES

If TCAN4x5x\_MCAN\_VERIFY\_CONFIGURATION\_WRITES is defined, then each MCAN configuration write will be read and verified for correctness.

### Enumeration Type Documentation

#### enum TCAN4x5x\_Device\_Mode\_Enum

##### Enumerator:

|  |  |
| --- | --- |
| TCAN4x5x\_DEVICE\_MODE\_NORMAL |  |
| TCAN4x5x\_DEVICE\_MODE\_STANDBY |  |
| TCAN4x5x\_DEVICE\_MODE\_SLEEP |  |

#### enum TCAN4x5x\_Device\_Test\_Mode\_Enum

##### Enumerator:

|  |  |
| --- | --- |
| TCAN4x5x\_DEVICE\_TEST\_MODE\_NORMAL |  |
| TCAN4x5x\_DEVICE\_TEST\_MODE\_PHY |  |
| TCAN4x5x\_DEVICE\_TEST\_MODE\_CONTROLLER |  |

#### enum TCAN4x5x\_MCAN\_FIFO\_Enum

##### Enumerator:

|  |  |
| --- | --- |
| RXFIFO0 |  |
| RXFIFO1 |  |

#### enum TCAN4x5x\_WDT\_Timer\_Enum

##### Enumerator:

|  |  |
| --- | --- |
| TCAN4x5x\_WDT\_60MS |  |
| TCAN4x5x\_WDT\_600MS |  |
| TCAN4x5x\_WDT\_3S |  |
| TCAN4x5x\_WDT\_6S |  |

## C:/Alphi/PCIeMiniSoftware/include/TCAN4x5x\_Data\_Structs.h File Reference

This file contains the TCAN4x5x data structures.

#include <stdint.h>

### Classes

struct **TCAN4x5x\_MCAN\_Data\_Timing\_Simple**

*Used to setup the data timing parameters of the MCAN module This is a simplified struct, requiring only the prescaler value (1:x), number of time quanta before and after the sample point.*

struct **TCAN4x5x\_MCAN\_Data\_Timing\_Raw**

*Used to setup the timing parameters of the MCAN module This is the raw MCAN form of the struct which takes in the same values as the actual Bosch MCAN core.*

struct **TCAN4x5x\_MCAN\_Nominal\_Timing\_Simple**

*Used to setup the nominal timing parameters of the MCAN module This is a simplified struct, requiring only the prescaler value (1:x), number of time quanta before and after the sample point.*

struct **TCAN4x5x\_MCAN\_Nominal\_Timing\_Raw**

*Used to setup the nominal timing parameters of the MCAN module This is the raw MCAN form of the struct which takes in the same values as the actual Bosch MCAN core.*

struct **TCAN4x5x\_MRAM\_Config**

*Defines the number of MRAM elements and the size of the elements.*

struct **TCAN4x5x\_MCAN\_CCCR\_Config**

*struct containing the bit fields of the MCAN CCCR register*

struct **TCAN4x5x\_MCAN\_Interrupts**

*Struct containing the MCAN interrupt bit field.*

struct **TCAN4x5x\_MCAN\_Interrupt\_Enable**

*Struct containing the MCAN interrupt enable bit field.*

struct **TCAN4x5x\_MCAN\_RX\_Header**

*CAN message header.*

struct **TCAN4x5x\_MCAN\_TX\_Header**

*CAN message header for transmitted messages.*

struct **TCAN4x5x\_MCAN\_SID\_Filter**

*Standard ID filter struct.*

struct **TCAN4x5x\_MCAN\_XID\_Filter**

*Extended ID filter struct.*

struct **TCAN4x5x\_Device\_Interrupts**

*Struct containing the device interrupt bit field.*

struct **TCAN4x5x\_Device\_Interrupt\_Enable**

*Struct containing the device interrupt enable bit field.*

### Enumerations

enum **TCAN4x5x\_MRAM\_Element\_Data\_Size** { **MRAM\_8\_Byte\_Data** = 0, **MRAM\_12\_Byte\_Data** = 0x1, **MRAM\_16\_Byte\_Data** = 0x2, **MRAM\_20\_Byte\_Data** = 0x3, **MRAM\_24\_Byte\_Data** = 0x4, **MRAM\_32\_Byte\_Data** = 0x5, **MRAM\_48\_Byte\_Data** = 0x6, **MRAM\_64\_Byte\_Data** = 0x7 }

*Data payload defines for the different MRAM sections, used by the* ***TCAN4x5x\_MRAM\_Config*** *struct.*

enum **TCAN4x5x\_SID\_SFEC\_Values** { **TCAN4x5x\_SID\_SFEC\_DISABLED** = 0x0, **TCAN4x5x\_SID\_SFEC\_STORERX0** = 0x1, **TCAN4x5x\_SID\_SFEC\_STORERX1** = 0x2, **TCAN4x5x\_SID\_SFEC\_REJECTMATCH** = 0x3, **TCAN4x5x\_SID\_SFEC\_PRIORITY** = 0x4, **TCAN4x5x\_SID\_SFEC\_PRIORITYSTORERX0** = 0x5, **TCAN4x5x\_SID\_SFEC\_PRIORITYSTORERX1** = 0x6, **TCAN4x5x\_SID\_SFEC\_STORERXBUFORDEBUG** = 0x7 }

enum **TCAN4x5x\_SID\_SFT\_Values** { **TCAN4x5x\_SID\_SFT\_DISABLED** = 0x3, **TCAN4x5x\_SID\_SFT\_CLASSIC** = 0x2, **TCAN4x5x\_SID\_SFT\_DUALID** = 0x1, **TCAN4x5x\_SID\_SFT\_RANGE** = 0x0 }

enum **TCAN4x5x\_XID\_EFEC\_Values** { **TCAN4x5x\_XID\_EFEC\_DISABLED** = 0x0, **TCAN4x5x\_XID\_EFEC\_STORERX0** = 0x1, **TCAN4x5x\_XID\_EFEC\_STORERX1** = 0x2, **TCAN4x5x\_XID\_EFEC\_REJECTMATCH** = 0x3, **TCAN4x5x\_XID\_EFEC\_PRIORITY** = 0x4, **TCAN4x5x\_XID\_EFEC\_PRIORITYSTORERX0** = 0x5, **TCAN4x5x\_XID\_EFEC\_PRIORITYSTORERX1** = 0x6, **TCAN4x5x\_XID\_EFEC\_STORERXBUFORDEBUG** = 0x7 }

enum **TCAN4x5x\_XID\_EFT\_Values** { **TCAN4x5x\_XID\_EFT\_RANGENOMASK** = 0x3, **TCAN4x5x\_XID\_EFT\_CLASSIC** = 0x2, **TCAN4x5x\_XID\_EFT\_DUALID** = 0x1, **TCAN4x5x\_XID\_EFT\_RANGE** = 0x0 }

### Detailed Description

This file contains the TCAN4x5x data structures.

It relies on the TCAN4x5x\_SPI abstraction functions Additional Feature Sets of **TCAN4550** vs TCAN4x5x:

Watchdog Timer Functions

Created on: Oct 1, 2017 Author: Texas Instruments

Copyright (c) 2017 Texas Instruments Incorporated. All rights reserved. Software License Agreement

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Texas Instruments Incorporated nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

### Enumeration Type Documentation

#### enum TCAN4x5x\_MRAM\_Element\_Data\_Size

Data payload defines for the different MRAM sections, used by the **TCAN4x5x\_MRAM\_Config** struct.

##### Enumerator:

|  |  |
| --- | --- |
| MRAM\_8\_Byte\_Data | 8 bytes of data payload |
| MRAM\_12\_Byte\_Data | 12 bytes of data payload |
| MRAM\_16\_Byte\_Data | 16 bytes of data payload |
| MRAM\_20\_Byte\_Data | 20 bytes of data payload |
| MRAM\_24\_Byte\_Data | 24 bytes of data payload |
| MRAM\_32\_Byte\_Data | 32 bytes of data payload |
| MRAM\_48\_Byte\_Data | 48 bytes of data payload |
| MRAM\_64\_Byte\_Data | 64 bytes of data payload |

#### enum TCAN4x5x\_SID\_SFEC\_Values

##### Enumerator:

|  |  |
| --- | --- |
| TCAN4x5x\_SID\_SFEC\_DISABLED | Disabled filter. This filter will do nothing if it matches a packet. |
| TCAN4x5x\_SID\_SFEC\_STORERX0 | Store in RX FIFO 0 if the filter matches the incoming message. |
| TCAN4x5x\_SID\_SFEC\_STORERX1 | Store in RX FIFO 1 if the filter matches the incoming message. |
| TCAN4x5x\_SID\_SFEC\_REJECTMATCH | Reject the packet (do not store, do not notify MCU) if the filter matches the incoming message. |
| TCAN4x5x\_SID\_SFEC\_PRIORITY | Store in default location but set a high priority message interrupt if the filter matches the incoming message. |
| TCAN4x5x\_SID\_SFEC\_PRIORITYSTORERX0 | Store in RX FIFO 0 and set a high priority message interrupt if the filter matches the incoming message. |
| TCAN4x5x\_SID\_SFEC\_PRIORITYSTORERX1 | Store in RX FIFO 1 and set a high priority message interrupt if the filter matches the incoming message. |
| TCAN4x5x\_SID\_SFEC\_STORERXBUFORDEBUG | Store in RX Buffer for debug if the filter matches the incoming message. SFT is ignored if this is selected. |

#### enum TCAN4x5x\_SID\_SFT\_Values

##### Enumerator:

|  |  |
| --- | --- |
| TCAN4x5x\_SID\_SFT\_DISABLED | Disabled filter. This filter will match nothing. |
| TCAN4x5x\_SID\_SFT\_CLASSIC | Classic filter with SFID1 as the ID to match, and SFID2 as the bit mask that applies to SFID1. |
| TCAN4x5x\_SID\_SFT\_DUALID | Dual ID filter, where both SFID1 and SFID2 hold IDs that can match (must match exactly) |
| TCAN4x5x\_SID\_SFT\_RANGE | Range Filter. SFID1 holds the start address, and SFID2 holds the end address. Any address in between will match. |

#### enum TCAN4x5x\_XID\_EFEC\_Values

##### Enumerator:

|  |  |
| --- | --- |
| TCAN4x5x\_XID\_EFEC\_DISABLED | Disabled filter. This filter will do nothing if it matches a packet. |
| TCAN4x5x\_XID\_EFEC\_STORERX0 | Store in RX FIFO 0 if the filter matches the incoming message. |
| TCAN4x5x\_XID\_EFEC\_STORERX1 | Store in RX FIFO 1 if the filter matches the incoming message. |
| TCAN4x5x\_XID\_EFEC\_REJECTMATCH | Reject the packet (do not store, do not notify MCU) if the filter matches the incoming message. |
| TCAN4x5x\_XID\_EFEC\_PRIORITY | Store in default location but set a high priority message interrupt if the filter matches the incoming message. |
| TCAN4x5x\_XID\_EFEC\_PRIORITYSTORERX0 | Store in RX FIFO 0 and set a high priority message interrupt if the filter matches the incoming message. |
| TCAN4x5x\_XID\_EFEC\_PRIORITYSTORERX1 | Store in RX FIFO 1 and set a high priority message interrupt if the filter matches the incoming message. |
| TCAN4x5x\_XID\_EFEC\_STORERXBUFORDEBUG | Store in RX Buffer for debug if the filter matches the incoming message. |

#### enum TCAN4x5x\_XID\_EFT\_Values

##### Enumerator:

|  |  |
| --- | --- |
| TCAN4x5x\_XID\_EFT\_RANGENOMASK | Range filter from EFID1 to EFID2, The XIDAM mask is not applied. |
| TCAN4x5x\_XID\_EFT\_CLASSIC | Classic Filter, EFID1 is the ID/filter, and EFID2 is the mask. |
| TCAN4x5x\_XID\_EFT\_DUALID | Dual ID filter matches if the incoming ID matches EFID1 or EFID2. |
| TCAN4x5x\_XID\_EFT\_RANGE | Range filter from EFID1 to EFID2. |

## C:/Alphi/PCIeMiniSoftware/include/TCAN4x5x\_Reg.h File Reference

This file contains the register definitions for the TCAN4x5x Family.

### Macros

#define **MRAM\_SIZE**  2048

#define **REG\_SPI\_CONFIG**  0x0000

#define **REG\_DEV\_CONFIG**  0x0800

#define **REG\_MCAN**  0x1000

#define **REG\_MRAM**  0x8000

#define **REG\_SPI\_DEVICE\_ID0**  0x0000

#define **REG\_SPI\_DEVICE\_ID1**  0x0004

#define **REG\_SPI\_REVISION**  0x0008

#define **REG\_SPI\_STATUS**  0x000C

#define **REG\_SPI\_ERROR\_STATUS\_MASK**  0x0010

#define **REG\_DEV\_MODES\_AND\_PINS**  0x0800

#define **REG\_DEV\_TIMESTAMP\_PRESCALER**  0x0804

#define **REG\_DEV\_TEST\_REGISTERS**  0x0808

#define **REG\_DEV\_IR**  0x0820

#define **REG\_DEV\_IE**  0x0830

#define **REG\_MCAN\_CREL**  0x1000

#define **REG\_MCAN\_ENDN**  0x1004

#define **REG\_MCAN\_CUST**  0x1008

#define **REG\_MCAN\_DBTP**  0x100C

#define **REG\_MCAN\_TEST**  0x1010

#define **REG\_MCAN\_RWD**  0x1014

#define **REG\_MCAN\_CCCR**  0x1018

#define **REG\_MCAN\_NBTP**  0x101C

#define **REG\_MCAN\_TSCC**  0x1020

#define **REG\_MCAN\_TSCV**  0x1024

#define **REG\_MCAN\_TOCC**  0x1028

#define **REG\_MCAN\_TOCV**  0x102C

#define **REG\_MCAN\_ECR**  0x1040

#define **REG\_MCAN\_PSR**  0x1044

#define **REG\_MCAN\_TDCR**  0x1048

#define **REG\_MCAN\_IR**  0x1050

#define **REG\_MCAN\_IE**  0x1054

#define **REG\_MCAN\_ILS**  0x1058

#define **REG\_MCAN\_ILE**  0x105C

#define **REG\_MCAN\_GFC**  0x1080

#define **REG\_MCAN\_SIDFC**  0x1084

#define **REG\_MCAN\_XIDFC**  0x1088

#define **REG\_MCAN\_XIDAM**  0x1090

#define **REG\_MCAN\_HPMS**  0x1094

#define **REG\_MCAN\_NDAT1**  0x1098

#define **REG\_MCAN\_NDAT2**  0x109C

#define **REG\_MCAN\_RXF0C**  0x10A0

#define **REG\_MCAN\_RXF0S**  0x10A4

#define **REG\_MCAN\_RXF0A**  0x10A8

#define **REG\_MCAN\_RXBC**  0x10AC

#define **REG\_MCAN\_RXF1C**  0x10B0

#define **REG\_MCAN\_RXF1S**  0x10B4

#define **REG\_MCAN\_RXF1A**  0x10B8

#define **REG\_MCAN\_RXESC**  0x10BC

#define **REG\_MCAN\_TXBC**  0x10C0

#define **REG\_MCAN\_TXFQS**  0x10C4

#define **REG\_MCAN\_TXESC**  0x10C8

#define **REG\_MCAN\_TXBRP**  0x10CC

#define **REG\_MCAN\_TXBAR**  0x10D0

#define **REG\_MCAN\_TXBCR**  0x10D4

#define **REG\_MCAN\_TXBTO**  0x10D8

#define **REG\_MCAN\_TXBCF**  0x10DC

#define **REG\_MCAN\_TXBTIE**  0x10E0

#define **REG\_MCAN\_TXBCIE**  0x10E4

#define **REG\_MCAN\_TXEFC**  0x10F0

#define **REG\_MCAN\_TXEFS**  0x10F4

#define **REG\_MCAN\_TXEFA**  0x10F8

#define **MCAN\_DLC\_0B**  0x00000000

#define **MCAN\_DLC\_1B**  0x00000001

#define **MCAN\_DLC\_2B**  0x00000002

#define **MCAN\_DLC\_3B**  0x00000003

#define **MCAN\_DLC\_4B**  0x00000004

#define **MCAN\_DLC\_5B**  0x00000005

#define **MCAN\_DLC\_6B**  0x00000006

#define **MCAN\_DLC\_7B**  0x00000007

#define **MCAN\_DLC\_8B**  0x00000008

#define **MCAN\_DLC\_12B**  0x00000009

#define **MCAN\_DLC\_16B**  0x0000000A

#define **MCAN\_DLC\_20B**  0x0000000B

#define **MCAN\_DLC\_24B**  0x0000000C

#define **MCAN\_DLC\_32B**  0x0000000D

#define **MCAN\_DLC\_48B**  0x0000000E

#define **MCAN\_DLC\_64B**  0x0000000F

#define **REG\_BITS\_MCAN\_DBTP\_TDC\_EN**  0x00800000

#define **REG\_BITS\_MCAN\_TEST\_RX\_DOM**  0x00000000

#define **REG\_BITS\_MCAN\_TEST\_RX\_REC**  0x00000080

#define **REG\_BITS\_MCAN\_TEST\_TX\_SP**  0x00000020

#define **REG\_BITS\_MCAN\_TEST\_TX\_DOM**  0x00000040

#define **REG\_BITS\_MCAN\_TEST\_TX\_REC**  0x00000060

#define **REG\_BITS\_MCAN\_TEST\_LOOP\_BACK**  0x00000010

#define **REG\_BITS\_MCAN\_CCCR\_RESERVED\_MASK**  0xFFFF0C00

#define **REG\_BITS\_MCAN\_CCCR\_NISO\_ISO**  0x00000000

#define **REG\_BITS\_MCAN\_CCCR\_NISO\_BOSCH**  0x00008000

#define **REG\_BITS\_MCAN\_CCCR\_TXP**  0x00004000

#define **REG\_BITS\_MCAN\_CCCR\_EFBI**  0x00002000

#define **REG\_BITS\_MCAN\_CCCR\_PXHD\_DIS**  0x00001000

#define **REG\_BITS\_MCAN\_CCCR\_BRSE**  0x00000200

#define **REG\_BITS\_MCAN\_CCCR\_FDOE**  0x00000100

#define **REG\_BITS\_MCAN\_CCCR\_TEST**  0x00000080

#define **REG\_BITS\_MCAN\_CCCR\_DAR\_DIS**  0x00000040

#define **REG\_BITS\_MCAN\_CCCR\_MON**  0x00000020

#define **REG\_BITS\_MCAN\_CCCR\_CSR**  0x00000010

#define **REG\_BITS\_MCAN\_CCCR\_CSA**  0x00000008

#define **REG\_BITS\_MCAN\_CCCR\_ASM**  0x00000004

#define **REG\_BITS\_MCAN\_CCCR\_CCE**  0x00000002

#define **REG\_BITS\_MCAN\_CCCR\_INIT**  0x00000001

#define **REG\_BITS\_MCAN\_IE\_ARAE**  0x20000000

#define **REG\_BITS\_MCAN\_IE\_PEDE**  0x10000000

#define **REG\_BITS\_MCAN\_IE\_PEAE**  0x08000000

#define **REG\_BITS\_MCAN\_IE\_WDIE**  0x04000000

#define **REG\_BITS\_MCAN\_IE\_BOE**  0x02000000

#define **REG\_BITS\_MCAN\_IE\_EWE**  0x01000000

#define **REG\_BITS\_MCAN\_IE\_EPE**  0x00800000

#define **REG\_BITS\_MCAN\_IE\_ELOE**  0x00400000

#define **REG\_BITS\_MCAN\_IE\_BEUE**  0x00200000

#define **REG\_BITS\_MCAN\_IE\_BECE**  0x00100000

#define **REG\_BITS\_MCAN\_IE\_DRXE**  0x00080000

#define **REG\_BITS\_MCAN\_IE\_TOOE**  0x00040000

#define **REG\_BITS\_MCAN\_IE\_MRAFE**  0x00020000

#define **REG\_BITS\_MCAN\_IE\_TSWE**  0x00010000

#define **REG\_BITS\_MCAN\_IE\_TEFLE**  0x00008000

#define **REG\_BITS\_MCAN\_IE\_TEFFE**  0x00004000

#define **REG\_BITS\_MCAN\_IE\_TEFWE**  0x00002000

#define **REG\_BITS\_MCAN\_IE\_TEFNE**  0x00001000

#define **REG\_BITS\_MCAN\_IE\_TFEE**  0x00000800

#define **REG\_BITS\_MCAN\_IE\_TCFE**  0x00000400

#define **REG\_BITS\_MCAN\_IE\_TCE**  0x00000200

#define **REG\_BITS\_MCAN\_IE\_HPME**  0x00000100

#define **REG\_BITS\_MCAN\_IE\_RF1LE**  0x00000080

#define **REG\_BITS\_MCAN\_IE\_RF1FE**  0x00000040

#define **REG\_BITS\_MCAN\_IE\_RF1WE**  0x00000020

#define **REG\_BITS\_MCAN\_IE\_RF1NE**  0x00000010

#define **REG\_BITS\_MCAN\_IE\_RF0LE**  0x00000008

#define **REG\_BITS\_MCAN\_IE\_RF0FE**  0x00000004

#define **REG\_BITS\_MCAN\_IE\_RF0WE**  0x00000002

#define **REG\_BITS\_MCAN\_IE\_RF0NE**  0x00000001

#define **REG\_BITS\_MCAN\_IR\_ARA**  0x20000000

#define **REG\_BITS\_MCAN\_IR\_PED**  0x10000000

#define **REG\_BITS\_MCAN\_IR\_PEA**  0x08000000

#define **REG\_BITS\_MCAN\_IR\_WDI**  0x04000000

#define **REG\_BITS\_MCAN\_IR\_BO**  0x02000000

#define **REG\_BITS\_MCAN\_IR\_EW**  0x01000000

#define **REG\_BITS\_MCAN\_IR\_EP**  0x00800000

#define **REG\_BITS\_MCAN\_IR\_ELO**  0x00400000

#define **REG\_BITS\_MCAN\_IR\_BEU**  0x00200000

#define **REG\_BITS\_MCAN\_IR\_BEC**  0x00100000

#define **REG\_BITS\_MCAN\_IR\_DRX**  0x00080000

#define **REG\_BITS\_MCAN\_IR\_TOO**  0x00040000

#define **REG\_BITS\_MCAN\_IR\_MRAF**  0x00020000

#define **REG\_BITS\_MCAN\_IR\_TSW**  0x00010000

#define **REG\_BITS\_MCAN\_IR\_TEFL**  0x00008000

#define **REG\_BITS\_MCAN\_IR\_TEFF**  0x00004000

#define **REG\_BITS\_MCAN\_IR\_TEFW**  0x00002000

#define **REG\_BITS\_MCAN\_IR\_TEFN**  0x00001000

#define **REG\_BITS\_MCAN\_IR\_TFE**  0x00000800

#define **REG\_BITS\_MCAN\_IR\_TCF**  0x00000400

#define **REG\_BITS\_MCAN\_IR\_TC**  0x00000200

#define **REG\_BITS\_MCAN\_IR\_HPM**  0x00000100

#define **REG\_BITS\_MCAN\_IR\_RF1L**  0x00000080

#define **REG\_BITS\_MCAN\_IR\_RF1F**  0x00000040

#define **REG\_BITS\_MCAN\_IR\_RF1W**  0x00000020

#define **REG\_BITS\_MCAN\_IR\_RF1N**  0x00000010

#define **REG\_BITS\_MCAN\_IR\_RF0L**  0x00000008

#define **REG\_BITS\_MCAN\_IR\_RF0F**  0x00000004

#define **REG\_BITS\_MCAN\_IR\_RF0W**  0x00000002

#define **REG\_BITS\_MCAN\_IR\_RF0N**  0x00000001

#define **REG\_BITS\_MCAN\_IE\_ARAL**  0x20000000

#define **REG\_BITS\_MCAN\_IE\_PEDL**  0x10000000

#define **REG\_BITS\_MCAN\_IE\_PEAL**  0x08000000

#define **REG\_BITS\_MCAN\_IE\_WDIL**  0x04000000

#define **REG\_BITS\_MCAN\_IE\_BOL**  0x02000000

#define **REG\_BITS\_MCAN\_IE\_EWL**  0x01000000

#define **REG\_BITS\_MCAN\_IE\_EPL**  0x00800000

#define **REG\_BITS\_MCAN\_IE\_ELOL**  0x00400000

#define **REG\_BITS\_MCAN\_IE\_BEUL**  0x00200000

#define **REG\_BITS\_MCAN\_IE\_BECL**  0x00100000

#define **REG\_BITS\_MCAN\_IE\_DRXL**  0x00080000

#define **REG\_BITS\_MCAN\_IE\_TOOL**  0x00040000

#define **REG\_BITS\_MCAN\_IE\_MRAFL**  0x00020000

#define **REG\_BITS\_MCAN\_IE\_TSWL**  0x00010000

#define **REG\_BITS\_MCAN\_IE\_TEFLL**  0x00008000

#define **REG\_BITS\_MCAN\_IE\_TEFFL**  0x00004000

#define **REG\_BITS\_MCAN\_IE\_TEFWL**  0x00002000

#define **REG\_BITS\_MCAN\_IE\_TEFNL**  0x00001000

#define **REG\_BITS\_MCAN\_IE\_TFEL**  0x00000800

#define **REG\_BITS\_MCAN\_IE\_TCFL**  0x00000400

#define **REG\_BITS\_MCAN\_IE\_TCL**  0x00000200

#define **REG\_BITS\_MCAN\_IE\_HPML**  0x00000100

#define **REG\_BITS\_MCAN\_IE\_RF1LL**  0x00000080

#define **REG\_BITS\_MCAN\_IE\_RF1FL**  0x00000040

#define **REG\_BITS\_MCAN\_IE\_RF1WL**  0x00000020

#define **REG\_BITS\_MCAN\_IE\_RF1NL**  0x00000010

#define **REG\_BITS\_MCAN\_IE\_RF0LL**  0x00000008

#define **REG\_BITS\_MCAN\_IE\_RF0FL**  0x00000004

#define **REG\_BITS\_MCAN\_IE\_RF0WL**  0x00000002

#define **REG\_BITS\_MCAN\_IE\_RF0NL**  0x00000001

#define **REG\_BITS\_MCAN\_ILE\_EINT1**  0x00000002

#define **REG\_BITS\_MCAN\_ILE\_EINT0**  0x00000001

#define **REG\_BITS\_MCAN\_GFC\_ANFS\_FIFO0**  0x00000000

#define **REG\_BITS\_MCAN\_GFC\_ANFS\_FIFO1**  0x00000010

#define **REG\_BITS\_MCAN\_GFC\_ANFE\_FIFO0**  0x00000000

#define **REG\_BITS\_MCAN\_GFC\_ANFE\_FIFO1**  0x00000004

#define **REG\_BITS\_MCAN\_GFC\_RRFS**  0x00000002

#define **REG\_BITS\_MCAN\_GFC\_RRFE**  0x00000001

#define **REG\_BITS\_MCAN\_RXF0C\_F0OM\_OVERWRITE**  0x80000000

#define **REG\_BITS\_MCAN\_RXESC\_RBDS\_8B**  0x00000000

#define **REG\_BITS\_MCAN\_RXESC\_RBDS\_12B**  0x00000100

#define **REG\_BITS\_MCAN\_RXESC\_RBDS\_16B**  0x00000200

#define **REG\_BITS\_MCAN\_RXESC\_RBDS\_20B**  0x00000300

#define **REG\_BITS\_MCAN\_RXESC\_RBDS\_24B**  0x00000400

#define **REG\_BITS\_MCAN\_RXESC\_RBDS\_32B**  0x00000500

#define **REG\_BITS\_MCAN\_RXESC\_RBDS\_48B**  0x00000600

#define **REG\_BITS\_MCAN\_RXESC\_RBDS\_64B**  0x00000700

#define **REG\_BITS\_MCAN\_RXESC\_F1DS\_8B**  0x00000000

#define **REG\_BITS\_MCAN\_RXESC\_F1DS\_12B**  0x00000010

#define **REG\_BITS\_MCAN\_RXESC\_F1DS\_16B**  0x00000020

#define **REG\_BITS\_MCAN\_RXESC\_F1DS\_20B**  0x00000030

#define **REG\_BITS\_MCAN\_RXESC\_F1DS\_24B**  0x00000040

#define **REG\_BITS\_MCAN\_RXESC\_F1DS\_32B**  0x00000050

#define **REG\_BITS\_MCAN\_RXESC\_F1DS\_48B**  0x00000060

#define **REG\_BITS\_MCAN\_RXESC\_F1DS\_64B**  0x00000070

#define **REG\_BITS\_MCAN\_RXESC\_F0DS\_8B**  0x00000000

#define **REG\_BITS\_MCAN\_RXESC\_F0DS\_12B**  0x00000001

#define **REG\_BITS\_MCAN\_RXESC\_F0DS\_16B**  0x00000002

#define **REG\_BITS\_MCAN\_RXESC\_F0DS\_20B**  0x00000003

#define **REG\_BITS\_MCAN\_RXESC\_F0DS\_24B**  0x00000004

#define **REG\_BITS\_MCAN\_RXESC\_F0DS\_32B**  0x00000005

#define **REG\_BITS\_MCAN\_RXESC\_F0DS\_48B**  0x00000006

#define **REG\_BITS\_MCAN\_RXESC\_F0DS\_64B**  0x00000007

#define **REG\_BITS\_MCAN\_TXBC\_TFQM**  0x40000000

#define **REG\_BITS\_MCAN\_TXESC\_TBDS\_8**  0x00000000

#define **REG\_BITS\_MCAN\_TXESC\_TBDS\_12**  0x00000001

#define **REG\_BITS\_MCAN\_TXESC\_TBDS\_16**  0x00000002

#define **REG\_BITS\_MCAN\_TXESC\_TBDS\_20**  0x00000003

#define **REG\_BITS\_MCAN\_TXESC\_TBDS\_24**  0x00000004

#define **REG\_BITS\_MCAN\_TXESC\_TBDS\_32**  0x00000005

#define **REG\_BITS\_MCAN\_TXESC\_TBDS\_48**  0x00000006

#define **REG\_BITS\_MCAN\_TXESC\_TBDS\_64**  0x00000007

#define **REG\_BITS\_MCAN\_TSCC\_PRESCALER\_MASK**  0x000F0000

#define **REG\_BITS\_MCAN\_TSCC\_COUNTER\_ALWAYS\_0**  0x00000000

#define **REG\_BITS\_MCAN\_TSCC\_COUNTER\_USE\_TCP**  0x00000001

#define **REG\_BITS\_MCAN\_TSCC\_COUNTER\_EXTERNAL**  0x00000002

#define **REG\_BITS\_MCAN\_TXBAR\_AR31**  0x80000000

#define **REG\_BITS\_MCAN\_TXBAR\_AR30**  0x40000000

#define **REG\_BITS\_MCAN\_TXBAR\_AR29**  0x20000000

#define **REG\_BITS\_MCAN\_TXBAR\_AR28**  0x10000000

#define **REG\_BITS\_MCAN\_TXBAR\_AR27**  0x08000000

#define **REG\_BITS\_MCAN\_TXBAR\_AR26**  0x04000000

#define **REG\_BITS\_MCAN\_TXBAR\_AR25**  0x02000000

#define **REG\_BITS\_MCAN\_TXBAR\_AR24**  0x01000000

#define **REG\_BITS\_MCAN\_TXBAR\_AR23**  0x00800000

#define **REG\_BITS\_MCAN\_TXBAR\_AR22**  0x00400000

#define **REG\_BITS\_MCAN\_TXBAR\_AR21**  0x00200000

#define **REG\_BITS\_MCAN\_TXBAR\_AR20**  0x00100000

#define **REG\_BITS\_MCAN\_TXBAR\_AR19**  0x00080000

#define **REG\_BITS\_MCAN\_TXBAR\_AR18**  0x00040000

#define **REG\_BITS\_MCAN\_TXBAR\_AR17**  0x00020000

#define **REG\_BITS\_MCAN\_TXBAR\_AR16**  0x00010000

#define **REG\_BITS\_MCAN\_TXBAR\_AR15**  0x00008000

#define **REG\_BITS\_MCAN\_TXBAR\_AR14**  0x00004000

#define **REG\_BITS\_MCAN\_TXBAR\_AR13**  0x00002000

#define **REG\_BITS\_MCAN\_TXBAR\_AR12**  0x00001000

#define **REG\_BITS\_MCAN\_TXBAR\_AR11**  0x00000800

#define **REG\_BITS\_MCAN\_TXBAR\_AR10**  0x00000400

#define **REG\_BITS\_MCAN\_TXBAR\_AR9**  0x00000200

#define **REG\_BITS\_MCAN\_TXBAR\_AR8**  0x00000100

#define **REG\_BITS\_MCAN\_TXBAR\_AR7**  0x00000080

#define **REG\_BITS\_MCAN\_TXBAR\_AR6**  0x00000040

#define **REG\_BITS\_MCAN\_TXBAR\_AR5**  0x00000020

#define **REG\_BITS\_MCAN\_TXBAR\_AR4**  0x00000010

#define **REG\_BITS\_MCAN\_TXBAR\_AR3**  0x00000008

#define **REG\_BITS\_MCAN\_TXBAR\_AR2**  0x00000004

#define **REG\_BITS\_MCAN\_TXBAR\_AR1**  0x00000002

#define **REG\_BITS\_MCAN\_TXBAR\_AR0**  0x00000001

#define **REG\_BITS\_MCAN\_TXBCR\_CR31**  0x80000000

#define **REG\_BITS\_MCAN\_TXBCR\_CR30**  0x40000000

#define **REG\_BITS\_MCAN\_TXBCR\_CR29**  0x20000000

#define **REG\_BITS\_MCAN\_TXBCR\_CR28**  0x10000000

#define **REG\_BITS\_MCAN\_TXBCR\_CR27**  0x08000000

#define **REG\_BITS\_MCAN\_TXBCR\_CR26**  0x04000000

#define **REG\_BITS\_MCAN\_TXBCR\_CR25**  0x02000000

#define **REG\_BITS\_MCAN\_TXBCR\_CR24**  0x01000000

#define **REG\_BITS\_MCAN\_TXBCR\_CR23**  0x00800000

#define **REG\_BITS\_MCAN\_TXBCR\_CR22**  0x00400000

#define **REG\_BITS\_MCAN\_TXBCR\_CR21**  0x00200000

#define **REG\_BITS\_MCAN\_TXBCR\_CR20**  0x00100000

#define **REG\_BITS\_MCAN\_TXBCR\_CR19**  0x00080000

#define **REG\_BITS\_MCAN\_TXBCR\_CR18**  0x00040000

#define **REG\_BITS\_MCAN\_TXBCR\_CR17**  0x00020000

#define **REG\_BITS\_MCAN\_TXBCR\_CR16**  0x00010000

#define **REG\_BITS\_MCAN\_TXBCR\_CR15**  0x00008000

#define **REG\_BITS\_MCAN\_TXBCR\_CR14**  0x00004000

#define **REG\_BITS\_MCAN\_TXBCR\_CR13**  0x00002000

#define **REG\_BITS\_MCAN\_TXBCR\_CR12**  0x00001000

#define **REG\_BITS\_MCAN\_TXBCR\_CR11**  0x00000800

#define **REG\_BITS\_MCAN\_TXBCR\_CR10**  0x00000400

#define **REG\_BITS\_MCAN\_TXBCR\_CR9**  0x00000200

#define **REG\_BITS\_MCAN\_TXBCR\_CR8**  0x00000100

#define **REG\_BITS\_MCAN\_TXBCR\_CR7**  0x00000080

#define **REG\_BITS\_MCAN\_TXBCR\_CR6**  0x00000040

#define **REG\_BITS\_MCAN\_TXBCR\_CR5**  0x00000020

#define **REG\_BITS\_MCAN\_TXBCR\_CR4**  0x00000010

#define **REG\_BITS\_MCAN\_TXBCR\_CR3**  0x00000008

#define **REG\_BITS\_MCAN\_TXBCR\_CR2**  0x00000004

#define **REG\_BITS\_MCAN\_TXBCR\_CR1**  0x00000002

#define **REG\_BITS\_MCAN\_TXBCR\_CR0**  0x00000001

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE31**  0x80000000

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE30**  0x40000000

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE29**  0x20000000

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE28**  0x10000000

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE27**  0x08000000

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE26**  0x04000000

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE25**  0x02000000

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE24**  0x01000000

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE23**  0x00800000

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE22**  0x00400000

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE21**  0x00200000

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE20**  0x00100000

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE19**  0x00080000

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE18**  0x00040000

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE17**  0x00020000

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE16**  0x00010000

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE15**  0x00008000

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE14**  0x00004000

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE13**  0x00002000

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE12**  0x00001000

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE11**  0x00000800

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE10**  0x00000400

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE9**  0x00000200

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE8**  0x00000100

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE7**  0x00000080

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE6**  0x00000040

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE5**  0x00000020

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE4**  0x00000010

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE3**  0x00000008

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE2**  0x00000004

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE1**  0x00000002

#define **REG\_BITS\_MCAN\_TXBTIE\_TIE0**  0x00000001

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE31**  0x80000000

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE30**  0x40000000

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE29**  0x20000000

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE28**  0x10000000

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE27**  0x08000000

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE26**  0x04000000

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE25**  0x02000000

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE24**  0x01000000

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE23**  0x00800000

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE22**  0x00400000

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE21**  0x00200000

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE20**  0x00100000

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE19**  0x00080000

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE18**  0x00040000

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE17**  0x00020000

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE16**  0x00010000

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE15**  0x00008000

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE14**  0x00004000

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE13**  0x00002000

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE12**  0x00001000

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE11**  0x00000800

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE10**  0x00000400

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE9**  0x00000200

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE8**  0x00000100

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE7**  0x00000080

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE6**  0x00000040

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE5**  0x00000020

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE4**  0x00000010

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE3**  0x00000008

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE2**  0x00000004

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE1**  0x00000002

#define **REG\_BITS\_MCAN\_TXBCIE\_CFIE0**  0x00000001

#define **REG\_BITS\_DEVICE\_MODE\_WAKE\_PIN\_MASK**  0xC0000000

#define **REG\_BITS\_DEVICE\_MODE\_WAKE\_PIN\_DIS**  0x00000000

#define **REG\_BITS\_DEVICE\_MODE\_WAKE\_PIN\_RISING**  0x40000000

#define **REG\_BITS\_DEVICE\_MODE\_WAKE\_PIN\_FALLING**  0x80000000

#define **REG\_BITS\_DEVICE\_MODE\_WAKE\_PIN\_BOTHEDGES**  0xC0000000

#define **REG\_BITS\_DEVICE\_MODE\_WD\_TIMER\_MASK**  0x30000000

#define **REG\_BITS\_DEVICE\_MODE\_WD\_TIMER\_60MS**  0x00000000

#define **REG\_BITS\_DEVICE\_MODE\_WD\_TIMER\_600MS**  0x10000000

#define **REG\_BITS\_DEVICE\_MODE\_WD\_TIMER\_3S**  0x20000000

#define **REG\_BITS\_DEVICE\_MODE\_WD\_TIMER\_6S**  0x30000000

#define **REG\_BITS\_DEVICE\_MODE\_WD\_CLK\_MASK**  0x08000000

#define **REG\_BITS\_DEVICE\_MODE\_WD\_CLK\_20MHZ**  0x00000000

#define **REG\_BITS\_DEVICE\_MODE\_WD\_CLK\_40MHZ**  0x08000000

#define **REG\_BITS\_DEVICE\_MODE\_GPO2\_MASK**  0x00C00000

#define **REG\_BITS\_DEVICE\_MODE\_GPO2\_CAN\_FAULT**  0x00000000

#define **REG\_BITS\_DEVICE\_MODE\_GPO2\_MCAN\_INT0**  0x00400000

#define **REG\_BITS\_DEVICE\_MODE\_GPO2\_WDT**  0x00800000

#define **REG\_BITS\_DEVICE\_MODE\_GPO2\_NINT**  0x00C00000

#define **REG\_BITS\_DEVICE\_MODE\_TESTMODE\_ENMASK**  0x00200000

#define **REG\_BITS\_DEVICE\_MODE\_TESTMODE\_EN**  0x00200000

#define **REG\_BITS\_DEVICE\_MODE\_TESTMODE\_DIS**  0x00000000

#define **REG\_BITS\_DEVICE\_MODE\_NWKRQ\_VOLT\_MASK**  0x00080000

#define **REG\_BITS\_DEVICE\_MODE\_NWKRQ\_VOLT\_INTERNAL**  0x00000000

#define **REG\_BITS\_DEVICE\_MODE\_NWKRQ\_VOLT\_VIO**  0x00080000

#define **REG\_BITS\_DEVICE\_MODE\_WDT\_RESET\_BIT**  0x00040000

#define **REG\_BITS\_DEVICE\_MODE\_WDT\_ACTION\_MASK**  0x00020000

#define **REG\_BITS\_DEVICE\_MODE\_WDT\_ACTION\_INT**  0x00000000

#define **REG\_BITS\_DEVICE\_MODE\_WDT\_ACTION\_INH\_PULSE**  0x00010000

#define **REG\_BITS\_DEVICE\_MODE\_WDT\_ACTION\_WDT\_PULSE**  0x00020000

#define **REG\_BITS\_DEVICE\_MODE\_GPO1\_MODE\_MASK**  0x0000C000

#define **REG\_BITS\_DEVICE\_MODE\_GPO1\_MODE\_GPO**  0x00000000

#define **REG\_BITS\_DEVICE\_MODE\_GPO1\_MODE\_CLKOUT**  0x00004000

#define **REG\_BITS\_DEVICE\_MODE\_GPO1\_MODE\_GPI**  0x00008000

#define **REG\_BITS\_DEVICE\_MODE\_FAIL\_SAFE\_MASK**  0x00002000

#define **REG\_BITS\_DEVICE\_MODE\_FAIL\_SAFE\_EN**  0x00002000

#define **REG\_BITS\_DEVICE\_MODE\_FAIL\_SAFE\_DIS**  0x00000000

#define **REG\_BITS\_DEVICE\_MODE\_CLKOUT\_MASK**  0x00001000

#define **REG\_BITS\_DEVICE\_MODE\_CLKOUT\_DIV1**  0x00000000

#define **REG\_BITS\_DEVICE\_MODE\_CLKOUT\_DIV2**  0x00001000

#define **REG\_BITS\_DEVICE\_MODE\_GPO1\_FUNC\_MASK**  0x00000C00

#define **REG\_BITS\_DEVICE\_MODE\_GPO1\_FUNC\_SPI\_INT**  0x00000000

#define **REG\_BITS\_DEVICE\_MODE\_GPO1\_FUNC\_MCAN\_INT1**  0x00000400

#define **REG\_BITS\_DEVICE\_MODE\_GPO1\_FUNC\_UVLO\_THERM**  0x00000800

#define **REG\_BITS\_DEVICE\_MODE\_INH\_MASK**  0x00000200

#define **REG\_BITS\_DEVICE\_MODE\_INH\_DIS**  0x00000200

#define **REG\_BITS\_DEVICE\_MODE\_INH\_EN**  0x00000000

#define **REG\_BITS\_DEVICE\_MODE\_NWKRQ\_CONFIG\_MASK**  0x00000100

#define **REG\_BITS\_DEVICE\_MODE\_NWKRQ\_CONFIG\_INH**  0x00000000

#define **REG\_BITS\_DEVICE\_MODE\_NWKRQ\_CONFIG\_WKRQ**  0x00000100

#define **REG\_BITS\_DEVICE\_MODE\_DEVICEMODE\_MASK**  0x000000C0

#define **REG\_BITS\_DEVICE\_MODE\_DEVICEMODE\_SLEEP**  0x00000000

#define **REG\_BITS\_DEVICE\_MODE\_DEVICEMODE\_STANDBY**  0x00000040

#define **REG\_BITS\_DEVICE\_MODE\_DEVICEMODE\_NORMAL**  0x00000080

#define **REG\_BITS\_DEVICE\_MODE\_WDT\_MASK**  0x00000008

#define **REG\_BITS\_DEVICE\_MODE\_WDT\_EN**  0x00000008

#define **REG\_BITS\_DEVICE\_MODE\_WDT\_DIS**  0x00000000

#define **REG\_BITS\_DEVICE\_MODE\_DEVICE\_RESET**  0x00000004

#define **REG\_BITS\_DEVICE\_MODE\_SWE\_MASK**  0x00000002

#define **REG\_BITS\_DEVICE\_MODE\_SWE\_DIS**  0x00000002

#define **REG\_BITS\_DEVICE\_MODE\_SWE\_EN**  0x00000000

#define **REG\_BITS\_DEVICE\_MODE\_TESTMODE\_MASK**  0x00000001

#define **REG\_BITS\_DEVICE\_MODE\_TESTMODE\_PHY**  0x00000000

#define **REG\_BITS\_DEVICE\_MODE\_TESTMODE\_CONTROLLER**  0x00000001

#define **REG\_BITS\_DEVICE\_IR\_CANBUSNOM**  0x80000000

#define **REG\_BITS\_DEVICE\_IR\_CANBUSTERMOPEN**  0x40000000

#define **REG\_BITS\_DEVICE\_IR\_CANHCANL**  0x20000000

#define **REG\_BITS\_DEVICE\_IR\_CANHBAT**  0x10000000

#define **REG\_BITS\_DEVICE\_IR\_CANLGND**  0x08000000

#define **REG\_BITS\_DEVICE\_IR\_CANBUSOPEN**  0x04000000

#define **REG\_BITS\_DEVICE\_IR\_CANBUSGND**  0x02000000

#define **REG\_BITS\_DEVICE\_IR\_CANBUSBAT**  0x01000000

#define **REG\_BITS\_DEVICE\_IR\_UVSUP**  0x00400000

#define **REG\_BITS\_DEVICE\_IR\_UVIO**  0x00200000

#define **REG\_BITS\_DEVICE\_IR\_PWRON**  0x00100000

#define **REG\_BITS\_DEVICE\_IR\_TSD**  0x00080000

#define **REG\_BITS\_DEVICE\_IR\_WDTO**  0x00040000

#define **REG\_BITS\_DEVICE\_IR\_ECCERR**  0x00010000

#define **REG\_BITS\_DEVICE\_IR\_CANINT**  0x00008000

#define **REG\_BITS\_DEVICE\_IR\_LWU**  0x00004000

#define **REG\_BITS\_DEVICE\_IR\_WKERR**  0x00002000

#define **REG\_BITS\_DEVICE\_IR\_FRAME\_OVF**  0x00001000

#define **REG\_BITS\_DEVICE\_IR\_CANSLNT**  0x00000400

#define **REG\_BITS\_DEVICE\_IR\_CANDOM**  0x00000100

#define **REG\_BITS\_DEVICE\_IR\_GLOBALERR**  0x00000080

#define **REG\_BITS\_DEVICE\_IR\_nWKRQ**  0x00000040

#define **REG\_BITS\_DEVICE\_IR\_CANERR**  0x00000020

#define **REG\_BITS\_DEVICE\_IR\_CANBUSFAULT**  0x00000010

#define **REG\_BITS\_DEVICE\_IR\_SPIERR**  0x00000008

#define **REG\_BITS\_DEVICE\_IR\_SWERR**  0x00000004

#define **REG\_BITS\_DEVICE\_IR\_M\_CAN\_INT**  0x00000002

#define **REG\_BITS\_DEVICE\_IR\_VTWD**  0x00000001

#define **REG\_BITS\_DEVICE\_IE\_CANBUSNOM**  0x80000000

#define **REG\_BITS\_DEVICE\_IE\_CANBUSTERMOPEN**  0x40000000

#define **REG\_BITS\_DEVICE\_IE\_CANHCANL**  0x20000000

#define **REG\_BITS\_DEVICE\_IE\_CANHBAT**  0x10000000

#define **REG\_BITS\_DEVICE\_IE\_CANLGND**  0x08000000

#define **REG\_BITS\_DEVICE\_IE\_CANBUSOPEN**  0x04000000

#define **REG\_BITS\_DEVICE\_IE\_CANBUSGND**  0x02000000

#define **REG\_BITS\_DEVICE\_IE\_CANBUSBAT**  0x01000000

#define **REG\_BITS\_DEVICE\_IE\_UVCCOUT**  0x00800000

#define **REG\_BITS\_DEVICE\_IE\_UVSUP**  0x00400000

#define **REG\_BITS\_DEVICE\_IE\_UVIO**  0x00200000

#define **REG\_BITS\_DEVICE\_IE\_PWRON**  0x00100000

#define **REG\_BITS\_DEVICE\_IE\_TSD**  0x00080000

#define **REG\_BITS\_DEVICE\_IE\_WDTO**  0x00040000

#define **REG\_BITS\_DEVICE\_IE\_ECCERR**  0x00010000

#define **REG\_BITS\_DEVICE\_IE\_CANINT**  0x00008000

#define **REG\_BITS\_DEVICE\_IE\_LWU**  0x00004000

#define **REG\_BITS\_DEVICE\_IE\_WKERR**  0x00002000

#define **REG\_BITS\_DEVICE\_IE\_FRAME\_OVF**  0x00001000

#define **REG\_BITS\_DEVICE\_IE\_CANSLNT**  0x00000400

#define **REG\_BITS\_DEVICE\_IE\_CANDOM**  0x00000100

#define **REG\_BITS\_DEVICE\_IE\_MASK**  0xFF69D700

### Detailed Description

This file contains the register definitions for the TCAN4x5x Family.

There are a few different define domains:

REG\_MCAN\_x: MCAN register address defines

MCAN\_DLC\_x: DLC values for the RX and TX FIFO element defines

REG\_SPI\_x : SPI Controller register address defines

REG\_DEV\_x : TCAN4x5x Device-specific register address defines

REG\_BITS\_x: Register bit defines in a similar manner as above. EX: REG\_BITS\_MCAN\_CCCR\_INIT is the hex value corresponding to the REG\_MCAN\_CCCR register's INIT bit

Created on: Oct 1, 2017

##### Author

Texas Instruments

### Macro Definition Documentation

#### #define MCAN\_DLC\_0B  0x00000000

#### #define MCAN\_DLC\_12B  0x00000009

#### #define MCAN\_DLC\_16B  0x0000000A

#### #define MCAN\_DLC\_1B  0x00000001

#### #define MCAN\_DLC\_20B  0x0000000B

#### #define MCAN\_DLC\_24B  0x0000000C

#### #define MCAN\_DLC\_2B  0x00000002

#### #define MCAN\_DLC\_32B  0x0000000D

#### #define MCAN\_DLC\_3B  0x00000003

#### #define MCAN\_DLC\_48B  0x0000000E

#### #define MCAN\_DLC\_4B  0x00000004

#### #define MCAN\_DLC\_5B  0x00000005

#### #define MCAN\_DLC\_64B  0x0000000F

#### #define MCAN\_DLC\_6B  0x00000006

#### #define MCAN\_DLC\_7B  0x00000007

#### #define MCAN\_DLC\_8B  0x00000008

#### #define MRAM\_SIZE  2048

#### #define REG\_BITS\_DEVICE\_IE\_CANBUSBAT  0x01000000

#### #define REG\_BITS\_DEVICE\_IE\_CANBUSGND  0x02000000

#### #define REG\_BITS\_DEVICE\_IE\_CANBUSNOM  0x80000000

#### #define REG\_BITS\_DEVICE\_IE\_CANBUSOPEN  0x04000000

#### #define REG\_BITS\_DEVICE\_IE\_CANBUSTERMOPEN  0x40000000

#### #define REG\_BITS\_DEVICE\_IE\_CANDOM  0x00000100

#### #define REG\_BITS\_DEVICE\_IE\_CANHBAT  0x10000000

#### #define REG\_BITS\_DEVICE\_IE\_CANHCANL  0x20000000

#### #define REG\_BITS\_DEVICE\_IE\_CANINT  0x00008000

#### #define REG\_BITS\_DEVICE\_IE\_CANLGND  0x08000000

#### #define REG\_BITS\_DEVICE\_IE\_CANSLNT  0x00000400

#### #define REG\_BITS\_DEVICE\_IE\_ECCERR  0x00010000

#### #define REG\_BITS\_DEVICE\_IE\_FRAME\_OVF  0x00001000

#### #define REG\_BITS\_DEVICE\_IE\_LWU  0x00004000

#### #define REG\_BITS\_DEVICE\_IE\_MASK  0xFF69D700

#### #define REG\_BITS\_DEVICE\_IE\_PWRON  0x00100000

#### #define REG\_BITS\_DEVICE\_IE\_TSD  0x00080000

#### #define REG\_BITS\_DEVICE\_IE\_UVCCOUT  0x00800000

#### #define REG\_BITS\_DEVICE\_IE\_UVIO  0x00200000

#### #define REG\_BITS\_DEVICE\_IE\_UVSUP  0x00400000

#### #define REG\_BITS\_DEVICE\_IE\_WDTO  0x00040000

#### #define REG\_BITS\_DEVICE\_IE\_WKERR  0x00002000

#### #define REG\_BITS\_DEVICE\_IR\_CANBUSBAT  0x01000000

#### #define REG\_BITS\_DEVICE\_IR\_CANBUSFAULT  0x00000010

#### #define REG\_BITS\_DEVICE\_IR\_CANBUSGND  0x02000000

#### #define REG\_BITS\_DEVICE\_IR\_CANBUSNOM  0x80000000

#### #define REG\_BITS\_DEVICE\_IR\_CANBUSOPEN  0x04000000

#### #define REG\_BITS\_DEVICE\_IR\_CANBUSTERMOPEN  0x40000000

#### #define REG\_BITS\_DEVICE\_IR\_CANDOM  0x00000100

#### #define REG\_BITS\_DEVICE\_IR\_CANERR  0x00000020

#### #define REG\_BITS\_DEVICE\_IR\_CANHBAT  0x10000000

#### #define REG\_BITS\_DEVICE\_IR\_CANHCANL  0x20000000

#### #define REG\_BITS\_DEVICE\_IR\_CANINT  0x00008000

#### #define REG\_BITS\_DEVICE\_IR\_CANLGND  0x08000000

#### #define REG\_BITS\_DEVICE\_IR\_CANSLNT  0x00000400

#### #define REG\_BITS\_DEVICE\_IR\_ECCERR  0x00010000

#### #define REG\_BITS\_DEVICE\_IR\_FRAME\_OVF  0x00001000

#### #define REG\_BITS\_DEVICE\_IR\_GLOBALERR  0x00000080

#### #define REG\_BITS\_DEVICE\_IR\_LWU  0x00004000

#### #define REG\_BITS\_DEVICE\_IR\_M\_CAN\_INT  0x00000002

#### #define REG\_BITS\_DEVICE\_IR\_nWKRQ  0x00000040

#### #define REG\_BITS\_DEVICE\_IR\_PWRON  0x00100000

#### #define REG\_BITS\_DEVICE\_IR\_SPIERR  0x00000008

#### #define REG\_BITS\_DEVICE\_IR\_SWERR  0x00000004

#### #define REG\_BITS\_DEVICE\_IR\_TSD  0x00080000

#### #define REG\_BITS\_DEVICE\_IR\_UVIO  0x00200000

#### #define REG\_BITS\_DEVICE\_IR\_UVSUP  0x00400000

#### #define REG\_BITS\_DEVICE\_IR\_VTWD  0x00000001

#### #define REG\_BITS\_DEVICE\_IR\_WDTO  0x00040000

#### #define REG\_BITS\_DEVICE\_IR\_WKERR  0x00002000

#### #define REG\_BITS\_DEVICE\_MODE\_CLKOUT\_DIV1  0x00000000

#### #define REG\_BITS\_DEVICE\_MODE\_CLKOUT\_DIV2  0x00001000

#### #define REG\_BITS\_DEVICE\_MODE\_CLKOUT\_MASK  0x00001000

#### #define REG\_BITS\_DEVICE\_MODE\_DEVICE\_RESET  0x00000004

#### #define REG\_BITS\_DEVICE\_MODE\_DEVICEMODE\_MASK  0x000000C0

#### #define REG\_BITS\_DEVICE\_MODE\_DEVICEMODE\_NORMAL  0x00000080

#### #define REG\_BITS\_DEVICE\_MODE\_DEVICEMODE\_SLEEP  0x00000000

#### #define REG\_BITS\_DEVICE\_MODE\_DEVICEMODE\_STANDBY  0x00000040

#### #define REG\_BITS\_DEVICE\_MODE\_FAIL\_SAFE\_DIS  0x00000000

#### #define REG\_BITS\_DEVICE\_MODE\_FAIL\_SAFE\_EN  0x00002000

#### #define REG\_BITS\_DEVICE\_MODE\_FAIL\_SAFE\_MASK  0x00002000

#### #define REG\_BITS\_DEVICE\_MODE\_GPO1\_FUNC\_MASK  0x00000C00

#### #define REG\_BITS\_DEVICE\_MODE\_GPO1\_FUNC\_MCAN\_INT1  0x00000400

#### #define REG\_BITS\_DEVICE\_MODE\_GPO1\_FUNC\_SPI\_INT  0x00000000

#### #define REG\_BITS\_DEVICE\_MODE\_GPO1\_FUNC\_UVLO\_THERM  0x00000800

#### #define REG\_BITS\_DEVICE\_MODE\_GPO1\_MODE\_CLKOUT  0x00004000

#### #define REG\_BITS\_DEVICE\_MODE\_GPO1\_MODE\_GPI  0x00008000

#### #define REG\_BITS\_DEVICE\_MODE\_GPO1\_MODE\_GPO  0x00000000

#### #define REG\_BITS\_DEVICE\_MODE\_GPO1\_MODE\_MASK  0x0000C000

#### #define REG\_BITS\_DEVICE\_MODE\_GPO2\_CAN\_FAULT  0x00000000

#### #define REG\_BITS\_DEVICE\_MODE\_GPO2\_MASK  0x00C00000

#### #define REG\_BITS\_DEVICE\_MODE\_GPO2\_MCAN\_INT0  0x00400000

#### #define REG\_BITS\_DEVICE\_MODE\_GPO2\_NINT  0x00C00000

#### #define REG\_BITS\_DEVICE\_MODE\_GPO2\_WDT  0x00800000

#### #define REG\_BITS\_DEVICE\_MODE\_INH\_DIS  0x00000200

#### #define REG\_BITS\_DEVICE\_MODE\_INH\_EN  0x00000000

#### #define REG\_BITS\_DEVICE\_MODE\_INH\_MASK  0x00000200

#### #define REG\_BITS\_DEVICE\_MODE\_NWKRQ\_CONFIG\_INH  0x00000000

#### #define REG\_BITS\_DEVICE\_MODE\_NWKRQ\_CONFIG\_MASK  0x00000100

#### #define REG\_BITS\_DEVICE\_MODE\_NWKRQ\_CONFIG\_WKRQ  0x00000100

#### #define REG\_BITS\_DEVICE\_MODE\_NWKRQ\_VOLT\_INTERNAL  0x00000000

#### #define REG\_BITS\_DEVICE\_MODE\_NWKRQ\_VOLT\_MASK  0x00080000

#### #define REG\_BITS\_DEVICE\_MODE\_NWKRQ\_VOLT\_VIO  0x00080000

#### #define REG\_BITS\_DEVICE\_MODE\_SWE\_DIS  0x00000002

#### #define REG\_BITS\_DEVICE\_MODE\_SWE\_EN  0x00000000

#### #define REG\_BITS\_DEVICE\_MODE\_SWE\_MASK  0x00000002

#### #define REG\_BITS\_DEVICE\_MODE\_TESTMODE\_CONTROLLER  0x00000001

#### #define REG\_BITS\_DEVICE\_MODE\_TESTMODE\_DIS  0x00000000

#### #define REG\_BITS\_DEVICE\_MODE\_TESTMODE\_EN  0x00200000

#### #define REG\_BITS\_DEVICE\_MODE\_TESTMODE\_ENMASK  0x00200000

#### #define REG\_BITS\_DEVICE\_MODE\_TESTMODE\_MASK  0x00000001

#### #define REG\_BITS\_DEVICE\_MODE\_TESTMODE\_PHY  0x00000000

#### #define REG\_BITS\_DEVICE\_MODE\_WAKE\_PIN\_BOTHEDGES  0xC0000000

#### #define REG\_BITS\_DEVICE\_MODE\_WAKE\_PIN\_DIS  0x00000000

#### #define REG\_BITS\_DEVICE\_MODE\_WAKE\_PIN\_FALLING  0x80000000

#### #define REG\_BITS\_DEVICE\_MODE\_WAKE\_PIN\_MASK  0xC0000000

#### #define REG\_BITS\_DEVICE\_MODE\_WAKE\_PIN\_RISING  0x40000000

#### #define REG\_BITS\_DEVICE\_MODE\_WD\_CLK\_20MHZ  0x00000000

#### #define REG\_BITS\_DEVICE\_MODE\_WD\_CLK\_40MHZ  0x08000000

#### #define REG\_BITS\_DEVICE\_MODE\_WD\_CLK\_MASK  0x08000000

#### #define REG\_BITS\_DEVICE\_MODE\_WD\_TIMER\_3S  0x20000000

#### #define REG\_BITS\_DEVICE\_MODE\_WD\_TIMER\_600MS  0x10000000

#### #define REG\_BITS\_DEVICE\_MODE\_WD\_TIMER\_60MS  0x00000000

#### #define REG\_BITS\_DEVICE\_MODE\_WD\_TIMER\_6S  0x30000000

#### #define REG\_BITS\_DEVICE\_MODE\_WD\_TIMER\_MASK  0x30000000

#### #define REG\_BITS\_DEVICE\_MODE\_WDT\_ACTION\_INH\_PULSE  0x00010000

#### #define REG\_BITS\_DEVICE\_MODE\_WDT\_ACTION\_INT  0x00000000

#### #define REG\_BITS\_DEVICE\_MODE\_WDT\_ACTION\_MASK  0x00020000

#### #define REG\_BITS\_DEVICE\_MODE\_WDT\_ACTION\_WDT\_PULSE  0x00020000

#### #define REG\_BITS\_DEVICE\_MODE\_WDT\_DIS  0x00000000

#### #define REG\_BITS\_DEVICE\_MODE\_WDT\_EN  0x00000008

#### #define REG\_BITS\_DEVICE\_MODE\_WDT\_MASK  0x00000008

#### #define REG\_BITS\_DEVICE\_MODE\_WDT\_RESET\_BIT  0x00040000

#### #define REG\_BITS\_MCAN\_CCCR\_ASM  0x00000004

#### #define REG\_BITS\_MCAN\_CCCR\_BRSE  0x00000200

#### #define REG\_BITS\_MCAN\_CCCR\_CCE  0x00000002

#### #define REG\_BITS\_MCAN\_CCCR\_CSA  0x00000008

#### #define REG\_BITS\_MCAN\_CCCR\_CSR  0x00000010

#### #define REG\_BITS\_MCAN\_CCCR\_DAR\_DIS  0x00000040

#### #define REG\_BITS\_MCAN\_CCCR\_EFBI  0x00002000

#### #define REG\_BITS\_MCAN\_CCCR\_FDOE  0x00000100

#### #define REG\_BITS\_MCAN\_CCCR\_INIT  0x00000001

#### #define REG\_BITS\_MCAN\_CCCR\_MON  0x00000020

#### #define REG\_BITS\_MCAN\_CCCR\_NISO\_BOSCH  0x00008000

#### #define REG\_BITS\_MCAN\_CCCR\_NISO\_ISO  0x00000000

#### #define REG\_BITS\_MCAN\_CCCR\_PXHD\_DIS  0x00001000

#### #define REG\_BITS\_MCAN\_CCCR\_RESERVED\_MASK  0xFFFF0C00

#### #define REG\_BITS\_MCAN\_CCCR\_TEST  0x00000080

#### #define REG\_BITS\_MCAN\_CCCR\_TXP  0x00004000

#### #define REG\_BITS\_MCAN\_DBTP\_TDC\_EN  0x00800000

#### #define REG\_BITS\_MCAN\_GFC\_ANFE\_FIFO0  0x00000000

#### #define REG\_BITS\_MCAN\_GFC\_ANFE\_FIFO1  0x00000004

#### #define REG\_BITS\_MCAN\_GFC\_ANFS\_FIFO0  0x00000000

#### #define REG\_BITS\_MCAN\_GFC\_ANFS\_FIFO1  0x00000010

#### #define REG\_BITS\_MCAN\_GFC\_RRFE  0x00000001

#### #define REG\_BITS\_MCAN\_GFC\_RRFS  0x00000002

#### #define REG\_BITS\_MCAN\_IE\_ARAE  0x20000000

#### #define REG\_BITS\_MCAN\_IE\_ARAL  0x20000000

#### #define REG\_BITS\_MCAN\_IE\_BECE  0x00100000

#### #define REG\_BITS\_MCAN\_IE\_BECL  0x00100000

#### #define REG\_BITS\_MCAN\_IE\_BEUE  0x00200000

#### #define REG\_BITS\_MCAN\_IE\_BEUL  0x00200000

#### #define REG\_BITS\_MCAN\_IE\_BOE  0x02000000

#### #define REG\_BITS\_MCAN\_IE\_BOL  0x02000000

#### #define REG\_BITS\_MCAN\_IE\_DRXE  0x00080000

#### #define REG\_BITS\_MCAN\_IE\_DRXL  0x00080000

#### #define REG\_BITS\_MCAN\_IE\_ELOE  0x00400000

#### #define REG\_BITS\_MCAN\_IE\_ELOL  0x00400000

#### #define REG\_BITS\_MCAN\_IE\_EPE  0x00800000

#### #define REG\_BITS\_MCAN\_IE\_EPL  0x00800000

#### #define REG\_BITS\_MCAN\_IE\_EWE  0x01000000

#### #define REG\_BITS\_MCAN\_IE\_EWL  0x01000000

#### #define REG\_BITS\_MCAN\_IE\_HPME  0x00000100

#### #define REG\_BITS\_MCAN\_IE\_HPML  0x00000100

#### #define REG\_BITS\_MCAN\_IE\_MRAFE  0x00020000

#### #define REG\_BITS\_MCAN\_IE\_MRAFL  0x00020000

#### #define REG\_BITS\_MCAN\_IE\_PEAE  0x08000000

#### #define REG\_BITS\_MCAN\_IE\_PEAL  0x08000000

#### #define REG\_BITS\_MCAN\_IE\_PEDE  0x10000000

#### #define REG\_BITS\_MCAN\_IE\_PEDL  0x10000000

#### #define REG\_BITS\_MCAN\_IE\_RF0FE  0x00000004

#### #define REG\_BITS\_MCAN\_IE\_RF0FL  0x00000004

#### #define REG\_BITS\_MCAN\_IE\_RF0LE  0x00000008

#### #define REG\_BITS\_MCAN\_IE\_RF0LL  0x00000008

#### #define REG\_BITS\_MCAN\_IE\_RF0NE  0x00000001

#### #define REG\_BITS\_MCAN\_IE\_RF0NL  0x00000001

#### #define REG\_BITS\_MCAN\_IE\_RF0WE  0x00000002

#### #define REG\_BITS\_MCAN\_IE\_RF0WL  0x00000002

#### #define REG\_BITS\_MCAN\_IE\_RF1FE  0x00000040

#### #define REG\_BITS\_MCAN\_IE\_RF1FL  0x00000040

#### #define REG\_BITS\_MCAN\_IE\_RF1LE  0x00000080

#### #define REG\_BITS\_MCAN\_IE\_RF1LL  0x00000080

#### #define REG\_BITS\_MCAN\_IE\_RF1NE  0x00000010

#### #define REG\_BITS\_MCAN\_IE\_RF1NL  0x00000010

#### #define REG\_BITS\_MCAN\_IE\_RF1WE  0x00000020

#### #define REG\_BITS\_MCAN\_IE\_RF1WL  0x00000020

#### #define REG\_BITS\_MCAN\_IE\_TCE  0x00000200

#### #define REG\_BITS\_MCAN\_IE\_TCFE  0x00000400

#### #define REG\_BITS\_MCAN\_IE\_TCFL  0x00000400

#### #define REG\_BITS\_MCAN\_IE\_TCL  0x00000200

#### #define REG\_BITS\_MCAN\_IE\_TEFFE  0x00004000

#### #define REG\_BITS\_MCAN\_IE\_TEFFL  0x00004000

#### #define REG\_BITS\_MCAN\_IE\_TEFLE  0x00008000

#### #define REG\_BITS\_MCAN\_IE\_TEFLL  0x00008000

#### #define REG\_BITS\_MCAN\_IE\_TEFNE  0x00001000

#### #define REG\_BITS\_MCAN\_IE\_TEFNL  0x00001000

#### #define REG\_BITS\_MCAN\_IE\_TEFWE  0x00002000

#### #define REG\_BITS\_MCAN\_IE\_TEFWL  0x00002000

#### #define REG\_BITS\_MCAN\_IE\_TFEE  0x00000800

#### #define REG\_BITS\_MCAN\_IE\_TFEL  0x00000800

#### #define REG\_BITS\_MCAN\_IE\_TOOE  0x00040000

#### #define REG\_BITS\_MCAN\_IE\_TOOL  0x00040000

#### #define REG\_BITS\_MCAN\_IE\_TSWE  0x00010000

#### #define REG\_BITS\_MCAN\_IE\_TSWL  0x00010000

#### #define REG\_BITS\_MCAN\_IE\_WDIE  0x04000000

#### #define REG\_BITS\_MCAN\_IE\_WDIL  0x04000000

#### #define REG\_BITS\_MCAN\_ILE\_EINT0  0x00000001

#### #define REG\_BITS\_MCAN\_ILE\_EINT1  0x00000002

#### #define REG\_BITS\_MCAN\_IR\_ARA  0x20000000

#### #define REG\_BITS\_MCAN\_IR\_BEC  0x00100000

#### #define REG\_BITS\_MCAN\_IR\_BEU  0x00200000

#### #define REG\_BITS\_MCAN\_IR\_BO  0x02000000

#### #define REG\_BITS\_MCAN\_IR\_DRX  0x00080000

#### #define REG\_BITS\_MCAN\_IR\_ELO  0x00400000

#### #define REG\_BITS\_MCAN\_IR\_EP  0x00800000

#### #define REG\_BITS\_MCAN\_IR\_EW  0x01000000

#### #define REG\_BITS\_MCAN\_IR\_HPM  0x00000100

#### #define REG\_BITS\_MCAN\_IR\_MRAF  0x00020000

#### #define REG\_BITS\_MCAN\_IR\_PEA  0x08000000

#### #define REG\_BITS\_MCAN\_IR\_PED  0x10000000

#### #define REG\_BITS\_MCAN\_IR\_RF0F  0x00000004

#### #define REG\_BITS\_MCAN\_IR\_RF0L  0x00000008

#### #define REG\_BITS\_MCAN\_IR\_RF0N  0x00000001

#### #define REG\_BITS\_MCAN\_IR\_RF0W  0x00000002

#### #define REG\_BITS\_MCAN\_IR\_RF1F  0x00000040

#### #define REG\_BITS\_MCAN\_IR\_RF1L  0x00000080

#### #define REG\_BITS\_MCAN\_IR\_RF1N  0x00000010

#### #define REG\_BITS\_MCAN\_IR\_RF1W  0x00000020

#### #define REG\_BITS\_MCAN\_IR\_TC  0x00000200

#### #define REG\_BITS\_MCAN\_IR\_TCF  0x00000400

#### #define REG\_BITS\_MCAN\_IR\_TEFF  0x00004000

#### #define REG\_BITS\_MCAN\_IR\_TEFL  0x00008000

#### #define REG\_BITS\_MCAN\_IR\_TEFN  0x00001000

#### #define REG\_BITS\_MCAN\_IR\_TEFW  0x00002000

#### #define REG\_BITS\_MCAN\_IR\_TFE  0x00000800

#### #define REG\_BITS\_MCAN\_IR\_TOO  0x00040000

#### #define REG\_BITS\_MCAN\_IR\_TSW  0x00010000

#### #define REG\_BITS\_MCAN\_IR\_WDI  0x04000000

#### #define REG\_BITS\_MCAN\_RXESC\_F0DS\_12B  0x00000001

#### #define REG\_BITS\_MCAN\_RXESC\_F0DS\_16B  0x00000002

#### #define REG\_BITS\_MCAN\_RXESC\_F0DS\_20B  0x00000003

#### #define REG\_BITS\_MCAN\_RXESC\_F0DS\_24B  0x00000004

#### #define REG\_BITS\_MCAN\_RXESC\_F0DS\_32B  0x00000005

#### #define REG\_BITS\_MCAN\_RXESC\_F0DS\_48B  0x00000006

#### #define REG\_BITS\_MCAN\_RXESC\_F0DS\_64B  0x00000007

#### #define REG\_BITS\_MCAN\_RXESC\_F0DS\_8B  0x00000000

#### #define REG\_BITS\_MCAN\_RXESC\_F1DS\_12B  0x00000010

#### #define REG\_BITS\_MCAN\_RXESC\_F1DS\_16B  0x00000020

#### #define REG\_BITS\_MCAN\_RXESC\_F1DS\_20B  0x00000030

#### #define REG\_BITS\_MCAN\_RXESC\_F1DS\_24B  0x00000040

#### #define REG\_BITS\_MCAN\_RXESC\_F1DS\_32B  0x00000050

#### #define REG\_BITS\_MCAN\_RXESC\_F1DS\_48B  0x00000060

#### #define REG\_BITS\_MCAN\_RXESC\_F1DS\_64B  0x00000070

#### #define REG\_BITS\_MCAN\_RXESC\_F1DS\_8B  0x00000000

#### #define REG\_BITS\_MCAN\_RXESC\_RBDS\_12B  0x00000100

#### #define REG\_BITS\_MCAN\_RXESC\_RBDS\_16B  0x00000200

#### #define REG\_BITS\_MCAN\_RXESC\_RBDS\_20B  0x00000300

#### #define REG\_BITS\_MCAN\_RXESC\_RBDS\_24B  0x00000400

#### #define REG\_BITS\_MCAN\_RXESC\_RBDS\_32B  0x00000500

#### #define REG\_BITS\_MCAN\_RXESC\_RBDS\_48B  0x00000600

#### #define REG\_BITS\_MCAN\_RXESC\_RBDS\_64B  0x00000700

#### #define REG\_BITS\_MCAN\_RXESC\_RBDS\_8B  0x00000000

#### #define REG\_BITS\_MCAN\_RXF0C\_F0OM\_OVERWRITE  0x80000000

#### #define REG\_BITS\_MCAN\_TEST\_LOOP\_BACK  0x00000010

#### #define REG\_BITS\_MCAN\_TEST\_RX\_DOM  0x00000000

#### #define REG\_BITS\_MCAN\_TEST\_RX\_REC  0x00000080

#### #define REG\_BITS\_MCAN\_TEST\_TX\_DOM  0x00000040

#### #define REG\_BITS\_MCAN\_TEST\_TX\_REC  0x00000060

#### #define REG\_BITS\_MCAN\_TEST\_TX\_SP  0x00000020

#### #define REG\_BITS\_MCAN\_TSCC\_COUNTER\_ALWAYS\_0  0x00000000

#### #define REG\_BITS\_MCAN\_TSCC\_COUNTER\_EXTERNAL  0x00000002

#### #define REG\_BITS\_MCAN\_TSCC\_COUNTER\_USE\_TCP  0x00000001

#### #define REG\_BITS\_MCAN\_TSCC\_PRESCALER\_MASK  0x000F0000

#### #define REG\_BITS\_MCAN\_TXBAR\_AR0  0x00000001

#### #define REG\_BITS\_MCAN\_TXBAR\_AR1  0x00000002

#### #define REG\_BITS\_MCAN\_TXBAR\_AR10  0x00000400

#### #define REG\_BITS\_MCAN\_TXBAR\_AR11  0x00000800

#### #define REG\_BITS\_MCAN\_TXBAR\_AR12  0x00001000

#### #define REG\_BITS\_MCAN\_TXBAR\_AR13  0x00002000

#### #define REG\_BITS\_MCAN\_TXBAR\_AR14  0x00004000

#### #define REG\_BITS\_MCAN\_TXBAR\_AR15  0x00008000

#### #define REG\_BITS\_MCAN\_TXBAR\_AR16  0x00010000

#### #define REG\_BITS\_MCAN\_TXBAR\_AR17  0x00020000

#### #define REG\_BITS\_MCAN\_TXBAR\_AR18  0x00040000

#### #define REG\_BITS\_MCAN\_TXBAR\_AR19  0x00080000

#### #define REG\_BITS\_MCAN\_TXBAR\_AR2  0x00000004

#### #define REG\_BITS\_MCAN\_TXBAR\_AR20  0x00100000

#### #define REG\_BITS\_MCAN\_TXBAR\_AR21  0x00200000

#### #define REG\_BITS\_MCAN\_TXBAR\_AR22  0x00400000

#### #define REG\_BITS\_MCAN\_TXBAR\_AR23  0x00800000

#### #define REG\_BITS\_MCAN\_TXBAR\_AR24  0x01000000

#### #define REG\_BITS\_MCAN\_TXBAR\_AR25  0x02000000

#### #define REG\_BITS\_MCAN\_TXBAR\_AR26  0x04000000

#### #define REG\_BITS\_MCAN\_TXBAR\_AR27  0x08000000

#### #define REG\_BITS\_MCAN\_TXBAR\_AR28  0x10000000

#### #define REG\_BITS\_MCAN\_TXBAR\_AR29  0x20000000

#### #define REG\_BITS\_MCAN\_TXBAR\_AR3  0x00000008

#### #define REG\_BITS\_MCAN\_TXBAR\_AR30  0x40000000

#### #define REG\_BITS\_MCAN\_TXBAR\_AR31  0x80000000

#### #define REG\_BITS\_MCAN\_TXBAR\_AR4  0x00000010

#### #define REG\_BITS\_MCAN\_TXBAR\_AR5  0x00000020

#### #define REG\_BITS\_MCAN\_TXBAR\_AR6  0x00000040

#### #define REG\_BITS\_MCAN\_TXBAR\_AR7  0x00000080

#### #define REG\_BITS\_MCAN\_TXBAR\_AR8  0x00000100

#### #define REG\_BITS\_MCAN\_TXBAR\_AR9  0x00000200

#### #define REG\_BITS\_MCAN\_TXBC\_TFQM  0x40000000

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE0  0x00000001

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE1  0x00000002

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE10  0x00000400

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE11  0x00000800

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE12  0x00001000

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE13  0x00002000

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE14  0x00004000

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE15  0x00008000

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE16  0x00010000

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE17  0x00020000

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE18  0x00040000

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE19  0x00080000

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE2  0x00000004

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE20  0x00100000

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE21  0x00200000

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE22  0x00400000

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE23  0x00800000

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE24  0x01000000

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE25  0x02000000

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE26  0x04000000

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE27  0x08000000

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE28  0x10000000

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE29  0x20000000

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE3  0x00000008

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE30  0x40000000

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE31  0x80000000

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE4  0x00000010

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE5  0x00000020

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE6  0x00000040

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE7  0x00000080

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE8  0x00000100

#### #define REG\_BITS\_MCAN\_TXBCIE\_CFIE9  0x00000200

#### #define REG\_BITS\_MCAN\_TXBCR\_CR0  0x00000001

#### #define REG\_BITS\_MCAN\_TXBCR\_CR1  0x00000002

#### #define REG\_BITS\_MCAN\_TXBCR\_CR10  0x00000400

#### #define REG\_BITS\_MCAN\_TXBCR\_CR11  0x00000800

#### #define REG\_BITS\_MCAN\_TXBCR\_CR12  0x00001000

#### #define REG\_BITS\_MCAN\_TXBCR\_CR13  0x00002000

#### #define REG\_BITS\_MCAN\_TXBCR\_CR14  0x00004000

#### #define REG\_BITS\_MCAN\_TXBCR\_CR15  0x00008000

#### #define REG\_BITS\_MCAN\_TXBCR\_CR16  0x00010000

#### #define REG\_BITS\_MCAN\_TXBCR\_CR17  0x00020000

#### #define REG\_BITS\_MCAN\_TXBCR\_CR18  0x00040000

#### #define REG\_BITS\_MCAN\_TXBCR\_CR19  0x00080000

#### #define REG\_BITS\_MCAN\_TXBCR\_CR2  0x00000004

#### #define REG\_BITS\_MCAN\_TXBCR\_CR20  0x00100000

#### #define REG\_BITS\_MCAN\_TXBCR\_CR21  0x00200000

#### #define REG\_BITS\_MCAN\_TXBCR\_CR22  0x00400000

#### #define REG\_BITS\_MCAN\_TXBCR\_CR23  0x00800000

#### #define REG\_BITS\_MCAN\_TXBCR\_CR24  0x01000000

#### #define REG\_BITS\_MCAN\_TXBCR\_CR25  0x02000000

#### #define REG\_BITS\_MCAN\_TXBCR\_CR26  0x04000000

#### #define REG\_BITS\_MCAN\_TXBCR\_CR27  0x08000000

#### #define REG\_BITS\_MCAN\_TXBCR\_CR28  0x10000000

#### #define REG\_BITS\_MCAN\_TXBCR\_CR29  0x20000000

#### #define REG\_BITS\_MCAN\_TXBCR\_CR3  0x00000008

#### #define REG\_BITS\_MCAN\_TXBCR\_CR30  0x40000000

#### #define REG\_BITS\_MCAN\_TXBCR\_CR31  0x80000000

#### #define REG\_BITS\_MCAN\_TXBCR\_CR4  0x00000010

#### #define REG\_BITS\_MCAN\_TXBCR\_CR5  0x00000020

#### #define REG\_BITS\_MCAN\_TXBCR\_CR6  0x00000040

#### #define REG\_BITS\_MCAN\_TXBCR\_CR7  0x00000080

#### #define REG\_BITS\_MCAN\_TXBCR\_CR8  0x00000100

#### #define REG\_BITS\_MCAN\_TXBCR\_CR9  0x00000200

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE0  0x00000001

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE1  0x00000002

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE10  0x00000400

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE11  0x00000800

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE12  0x00001000

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE13  0x00002000

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE14  0x00004000

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE15  0x00008000

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE16  0x00010000

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE17  0x00020000

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE18  0x00040000

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE19  0x00080000

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE2  0x00000004

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE20  0x00100000

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE21  0x00200000

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE22  0x00400000

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE23  0x00800000

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE24  0x01000000

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE25  0x02000000

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE26  0x04000000

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE27  0x08000000

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE28  0x10000000

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE29  0x20000000

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE3  0x00000008

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE30  0x40000000

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE31  0x80000000

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE4  0x00000010

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE5  0x00000020

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE6  0x00000040

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE7  0x00000080

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE8  0x00000100

#### #define REG\_BITS\_MCAN\_TXBTIE\_TIE9  0x00000200

#### #define REG\_BITS\_MCAN\_TXESC\_TBDS\_12  0x00000001

#### #define REG\_BITS\_MCAN\_TXESC\_TBDS\_16  0x00000002

#### #define REG\_BITS\_MCAN\_TXESC\_TBDS\_20  0x00000003

#### #define REG\_BITS\_MCAN\_TXESC\_TBDS\_24  0x00000004

#### #define REG\_BITS\_MCAN\_TXESC\_TBDS\_32  0x00000005

#### #define REG\_BITS\_MCAN\_TXESC\_TBDS\_48  0x00000006

#### #define REG\_BITS\_MCAN\_TXESC\_TBDS\_64  0x00000007

#### #define REG\_BITS\_MCAN\_TXESC\_TBDS\_8  0x00000000

#### #define REG\_DEV\_CONFIG  0x0800

#### #define REG\_DEV\_IE  0x0830

#### #define REG\_DEV\_IR  0x0820

#### #define REG\_DEV\_MODES\_AND\_PINS  0x0800

#### #define REG\_DEV\_TEST\_REGISTERS  0x0808

#### #define REG\_DEV\_TIMESTAMP\_PRESCALER  0x0804

#### #define REG\_MCAN  0x1000

#### #define REG\_MCAN\_CCCR  0x1018

#### #define REG\_MCAN\_CREL  0x1000

#### #define REG\_MCAN\_CUST  0x1008

#### #define REG\_MCAN\_DBTP  0x100C

#### #define REG\_MCAN\_ECR  0x1040

#### #define REG\_MCAN\_ENDN  0x1004

#### #define REG\_MCAN\_GFC  0x1080

#### #define REG\_MCAN\_HPMS  0x1094

#### #define REG\_MCAN\_IE  0x1054

#### #define REG\_MCAN\_ILE  0x105C

#### #define REG\_MCAN\_ILS  0x1058

#### #define REG\_MCAN\_IR  0x1050

#### #define REG\_MCAN\_NBTP  0x101C

#### #define REG\_MCAN\_NDAT1  0x1098

#### #define REG\_MCAN\_NDAT2  0x109C

#### #define REG\_MCAN\_PSR  0x1044

#### #define REG\_MCAN\_RWD  0x1014

#### #define REG\_MCAN\_RXBC  0x10AC

#### #define REG\_MCAN\_RXESC  0x10BC

#### #define REG\_MCAN\_RXF0A  0x10A8

#### #define REG\_MCAN\_RXF0C  0x10A0

#### #define REG\_MCAN\_RXF0S  0x10A4

#### #define REG\_MCAN\_RXF1A  0x10B8

#### #define REG\_MCAN\_RXF1C  0x10B0

#### #define REG\_MCAN\_RXF1S  0x10B4

#### #define REG\_MCAN\_SIDFC  0x1084

#### #define REG\_MCAN\_TDCR  0x1048

#### #define REG\_MCAN\_TEST  0x1010

#### #define REG\_MCAN\_TOCC  0x1028

#### #define REG\_MCAN\_TOCV  0x102C

#### #define REG\_MCAN\_TSCC  0x1020

#### #define REG\_MCAN\_TSCV  0x1024

#### #define REG\_MCAN\_TXBAR  0x10D0

#### #define REG\_MCAN\_TXBC  0x10C0

#### #define REG\_MCAN\_TXBCF  0x10DC

#### #define REG\_MCAN\_TXBCIE  0x10E4

#### #define REG\_MCAN\_TXBCR  0x10D4

#### #define REG\_MCAN\_TXBRP  0x10CC

#### #define REG\_MCAN\_TXBTIE  0x10E0

#### #define REG\_MCAN\_TXBTO  0x10D8

#### #define REG\_MCAN\_TXEFA  0x10F8

#### #define REG\_MCAN\_TXEFC  0x10F0

#### #define REG\_MCAN\_TXEFS  0x10F4

#### #define REG\_MCAN\_TXESC  0x10C8

#### #define REG\_MCAN\_TXFQS  0x10C4

#### #define REG\_MCAN\_XIDAM  0x1090

#### #define REG\_MCAN\_XIDFC  0x1088

#### #define REG\_MRAM  0x8000

#### #define REG\_SPI\_CONFIG  0x0000

#### #define REG\_SPI\_DEVICE\_ID0  0x0000

#### #define REG\_SPI\_DEVICE\_ID1  0x0004

#### #define REG\_SPI\_ERROR\_STATUS\_MASK  0x0010

#### #define REG\_SPI\_REVISION  0x0008

#### #define REG\_SPI\_STATUS  0x000C

#### 

## C:/Alphi/PCIeMiniSoftware/include/TCAN4x5x\_SPI.h File Reference

This file is responsible for abstracting the lower-level microcontroller SPI read and write functions.

#include "AlteraSpi.h"

### Macros

#define **AHB\_WRITE\_OPCODE**  0x61

#define **AHB\_READ\_OPCODE**  0x41

### Detailed Description

This file is responsible for abstracting the lower-level microcontroller SPI read and write functions.

### Macro Definition Documentation

#### #define AHB\_READ\_OPCODE  0x41

#### #define AHB\_WRITE\_OPCODE  0x61

#### 

## dllmain.cpp File Reference

#include "Windows.h"

### Functions

BOOL APIENTRY **DllMain** (HMODULE hModule, DWORD ul\_reason\_for\_call, LPVOID lpReserved)

### Function Documentation

#### BOOL APIENTRY DllMain (HMODULE *hModule*, DWORD *ul\_reason\_for\_call*, LPVOID *lpReserved*)

#### 

## PCIe\_Mini\_CAN\_FD.cpp File Reference

Implementation of the **PCIeMini\_CAN\_FD** board class.

#include <stdio.h>

#include "PCIeMini\_CAN\_FD.h"

### Detailed Description

Implementation of the **PCIeMini\_CAN\_FD** board class.

## TCAN4550.cpp File Reference

This file contains **TCAN4550** functions, and relies on the TCAN4x5x\_SPI abstraction functions Additional Feature Sets of **TCAN4550** vs TCAN4x5x:

#include <stdint.h>

#include <stdio.h>

#include "TCAN4550.h"

### Detailed Description

This file contains **TCAN4550** functions, and relies on the TCAN4x5x\_SPI abstraction functions Additional Feature Sets of **TCAN4550** vs TCAN4x5x:

Watchdog Timer Functions

## TCAN4x5x\_SPI.cpp File Reference

This file is responsible for abstracting the lower-level microcontroller SPI read and write functions.

#include <stdint.h>

#include "TCAN4550.h"

### Macros

#define **USE\_AHB\_CODE**  1

#define **WAIT\_FOR\_IDLE**()

### Detailed Description

This file is responsible for abstracting the lower-level microcontroller SPI read and write functions.

### Macro Definition Documentation

#### #define USE\_AHB\_CODE  1

#### #define WAIT\_FOR\_IDLE()

**Value:**do \

{ \

status = getStatus(); \

if (status & ALTERA\_AVALON\_SPI\_CONTROL\_IE\_MSK) resetStatus(); \

} while ((status & status\_TRDY\_mask) == 0);

## x64/Debug/PCIeMini\_CAN\_FD\_lib.vcxproj.FileListAbsolute.txt File Reference

## C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/AlphiBoard.cpp File Reference

Implementation of the base PCIe board class with Jungo driver and Altera PCIe hardware.

#include "wdc\_lib.h"

#include "wdc\_defs.h"

#include "utils.h"

#include "AlphiErrorCodes.h"

#include "AlphiBoard.h"

#include "status\_strings.h"

#include "wdc\_diag\_lib.h"

#include <iostream>

### Macros

#define **QT\_CORE\_LIB**

#define **MINIPCIE\_ARINC429\_DEFAULT\_LICENSE\_STRING**  ((CHAR \*)"872759db47d9ae7988a60332b89b6ea9c386010a7656b25cfdb4076053056c6f590d.WD1440\_64\_NL\_Alphi\_Technology\_Corporation-DIS")

#define **MINIPCIE\_ARINC429\_DEFAULT\_DRIVER\_NAME**  WD\_DEFAULT\_DRIVER\_NAME\_BASE

### Detailed Description

Implementation of the base PCIe board class with Jungo driver and Altera PCIe hardware.

### Macro Definition Documentation

#### #define MINIPCIE\_ARINC429\_DEFAULT\_DRIVER\_NAME  WD\_DEFAULT\_DRIVER\_NAME\_BASE

#### #define MINIPCIE\_ARINC429\_DEFAULT\_LICENSE\_STRING  ((CHAR \*)"872759db47d9ae7988a60332b89b6ea9c386010a7656b25cfdb4076053056c6f590d.WD1440\_64\_NL\_Alphi\_Technology\_Corporation-DIS")

#### #define QT\_CORE\_LIB

#### 

## C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/AlphiBoard\_dma.cpp File Reference

Implementation of the DMAs for the base PCIe board class with Jungo driver and Altera PCIe hardware.

#include "AlphiBoard.h"

#include "wdc\_defs.h"

#include "wdc\_lib.h"

#include "status\_strings.h"

#include "stdio.h"

#include "string.h"

#include "utils.h"

### Variables

**MINIPCIE\_INT\_HANDLER** **MyDmaIntHandler**

### Detailed Description

Implementation of the DMAs for the base PCIe board class with Jungo driver and Altera PCIe hardware.

### Variable Documentation

#### MINIPCIE\_INT\_HANDLER MyDmaIntHandler

#### 

## C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/AlphiBoard\_irq.cpp File Reference

Implementation of the interrupts for the base PCIe board class with Jungo driver and Altera PCIe hardware.

#include "AlphiBoard.h"

#include "wdc\_defs.h"

#include "wdc\_lib.h"

#include "stdio.h"

#include "string.h"

#include "utils.h"

#include "status\_strings.h"

#include "wdc\_diag\_lib.h"

### Macros

#define **NUM\_TRANS\_CMDS**  0

### Functions

static void **MINIPCIE\_ARINC429\_IntHandler** (PVOID pData)

static BOOL **doesItemExists** (PWDC\_DEVICE pDev, ITEM\_TYPE item)

### Detailed Description

Implementation of the interrupts for the base PCIe board class with Jungo driver and Altera PCIe hardware.

### Macro Definition Documentation

#### #define NUM\_TRANS\_CMDS  0

### Function Documentation

#### static BOOL doesItemExists (PWDC\_DEVICE *pDev*, ITEM\_TYPE *item*)[static]

Check whether a given device contains an item of the specified type

#### static void MINIPCIE\_ARINC429\_IntHandler (PVOID *pData*)[static]

Interrupt handler routine

## C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/AlteraDma.cpp File Reference

Implementation of the DMA block controller.

#include "AlteraDma.h"

### Detailed Description

Implementation of the DMA block controller.

## C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/AlteraSpi.cpp File Reference

Implementation of the low-level access routines to the SPI.

#include "stdint.h"

#include "AlteraSpi.h"

### Detailed Description

Implementation of the low-level access routines to the SPI.

## C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/PcieCra.cpp File Reference

PCIe interface CRA class.

#include "PcieCra.h"

### Detailed Description

PCIe interface CRA class.

## C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/PCIeMini\_error.cpp File Reference

Error code to string conversion.

#include "stdio.h"

#include <string.h>

#include "AlphiErrorCodes.h"

#include "windrvr.h"

### Functions

char \* **wdErrorToString** (**PCIeMini\_status** errCode)

DLL char \* **getAlphiErrorMsg** (**PCIeMini\_status** errCode)

*Gives the string description of an error code.*

### Detailed Description

Error code to string conversion.

### Function Documentation

#### DLL char\* getAlphiErrorMsg (PCIeMini\_status *errCode*)

Gives the string description of an error code.

##### Parameters

|  |  |
| --- | --- |
| *errCode* | Error code returned by a function |

##### Return values

|  |  |
| --- | --- |
| *C-string,description* | of the error |

#### char \* wdErrorToString (PCIeMini\_status *errCode*)

#### 

## C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/TestProgram.cpp File Reference

Utility program class for the test programs.

#include "TestProgram.h"

### Detailed Description

Utility program class for the test programs.

## C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/x64/Debug/CodeAnalysisResultManifest.txt File Reference

## C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/x64/Debug/PCIeMini\_lib.vcxproj.FileListAbsolute.txt File Reference

## C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/x64/Release/PCIeMini\_lib.vcxproj.FileListAbsolute.txt File Reference

# Index

~AlphiBoard

AlphiBoard, 10

Address

LinearAddress, 26

AHB\_READ\_32

TcanInterface, 104

AHB\_READ\_BURST\_END

TcanInterface, 104

AHB\_READ\_BURST\_READ

TcanInterface, 104

AHB\_READ\_BURST\_START

TcanInterface, 104

AHB\_READ\_OPCODE

TCAN4x5x\_SPI.h, 164

AHB\_WRITE\_32

TcanInterface, 104

AHB\_WRITE\_BURST\_END

TcanInterface, 104

AHB\_WRITE\_BURST\_START

TcanInterface, 104

AHB\_WRITE\_BURST\_WRITE

TcanInterface, 104

AHB\_WRITE\_OPCODE

TCAN4x5x\_SPI.h, 164

ALPHI\_S\_OK

AlphiDll.h, 111

AlphiBoard, 8

~AlphiBoard, 10

AlphiBoard, 10

bar0, 14

bar2, 14

bar3, 14

Close, 10

cra, 15

disableInterrupts, 10

DMAClose, 11

DMAOpen, 11

DMARoutine, 11

DMATransfer, 11

enableInterrupts, 11

getBar0Address, 11

getBar2Address, 11

getBar3Address, 12

getFpgaID, 12

getFpgaTimeStamp, 12

getIntResults, 12

hookInterruptServiceRoutine, 12

hwDMAInterruptDisable, 13

hwDMAInterruptEnable, 13

hwDMAProgram, 13

hwDMAStart, 13

hwDMAWaitForCompletion, 13

IsValidDevice, 13

libStatus, 15

MsSleep, 13

Open, 13

pDma, 15

reset, 14

setVerbose, 14

sysid, 15

unhookInterruptServiceRoutine, 14

verbose, 15

AlphiBoard.cpp

MINIPCIE\_ARINC429\_DEFAULT\_DRIVER\_NAME, 170

MINIPCIE\_ARINC429\_DEFAULT\_LICENSE\_STRING, 170

QT\_CORE\_LIB, 170

AlphiBoard.h

clock\_gettime, 109

CLOCK\_REALTIME, 109

ErrLog, 109

LinearAddress, 109

MINIPCIE\_EVENT\_HANDLER, 109

MINIPCIE\_INT\_HANDLER, 109

nanosleep, 109

PMINIPCIE\_DEV\_CTX, 109

TraceLog, 109

usleep, 109

AlphiBoard\_dma.cpp

MyDmaIntHandler, 171

AlphiBoard\_irq.cpp

doesItemExists, 172

MINIPCIE\_ARINC429\_IntHandler, 172

NUM\_TRANS\_CMDS, 172

AlphiDll.h

ALPHI\_S\_OK, 111

Dll, 111

FALSE, 111

HRESULT, 111

LINUX, 111

TRUE, 111

UNIX, 111

AlphiErrorCodes.h

ERRCODE\_BOARD\_NOT\_PRESENT, 114

ERRCODE\_BUSY, 114

ERRCODE\_FAILED\_SELF\_TEST, 114

ERRCODE\_INPUT\_MODE, 114

ERRCODE\_INT\_ALREADY\_ENABLED, 114

ERRCODE\_INT\_NOT\_ENABLED, 114

ERRCODE\_INTERNAL\_ERROR, 114

ERRCODE\_INVALID\_ALIGNMENT, 114

ERRCODE\_INVALID\_BOARD\_NUM, 114

ERRCODE\_INVALID\_CHANNEL\_NUM, 114

ERRCODE\_INVALID\_COMMAND, 114

ERRCODE\_INVALID\_DATALENGTH, 114

ERRCODE\_INVALID\_DATARATE, 114

ERRCODE\_INVALID\_EMTINTSTS, 114

ERRCODE\_INVALID\_FINTSTS, 114

ERRCODE\_INVALID\_FREQUENCY, 114

ERRCODE\_INVALID\_GROUP, 114

ERRCODE\_INVALID\_HANDLE, 114

ERRCODE\_INVALID\_HFINTSTS, 114

ERRCODE\_INVALID\_INPUT\_MODE, 114

ERRCODE\_INVALID\_LABELNUMBER, 114

ERRCODE\_INVALID\_LOGIC\_SEL, 114

ERRCODE\_INVALID\_MASK\_VALUE, 114

ERRCODE\_INVALID\_MODE, 114

ERRCODE\_INVALID\_PARITY, 115

ERRCODE\_INVALID\_RECVNUMBER, 115

ERRCODE\_INVALID\_RECVSTATUS, 115

ERRCODE\_INVALID\_SDI, 115

ERRCODE\_INVALID\_SELF\_TEST\_DATA, 115

ERRCODE\_INVALID\_SELF\_TEST\_ENABLE\_VAL, 115

ERRCODE\_INVALID\_SSM, 115

ERRCODE\_INVALID\_STROBE\_MODE, 115

ERRCODE\_INVALID\_TIME\_BOUNCE\_VAL, 115

ERRCODE\_INVALID\_TRANRECVSTS, 115

ERRCODE\_INVALID\_TRANSNUMBER, 115

ERRCODE\_INVALID\_TRANSSTATUS, 115

ERRCODE\_INVALID\_TRPAIR, 115

ERRCODE\_INVALID\_VALUE, 115

ERRCODE\_NO\_ERROR, 115

ERRCODE\_OUTPUT\_MODE, 115

ERRCODE\_RX\_UNDERFLOW, 115

ERRCODE\_SELF\_TEST\_DISABLE, 115

ERRCODE\_SUCCESS, 115

ERRCODE\_TIMEOUT, 115

ERRCODE\_TX\_OVERFLOW, 115

getAlphiErrorMsg, 116

PCIeMini\_status, 115

ALT\_AVALON\_DMA\_MODE\_128

AlteraDma.h, 118

ALT\_AVALON\_DMA\_MODE\_16

AlteraDma.h, 118

ALT\_AVALON\_DMA\_MODE\_32

AlteraDma.h, 118

ALT\_AVALON\_DMA\_MODE\_64

AlteraDma.h, 118

ALT\_AVALON\_DMA\_MODE\_8

AlteraDma.h, 118

ALT\_AVALON\_DMA\_MODE\_MSK

AlteraDma.h, 118

ALT\_AVALON\_DMA\_NSLOTS

AlteraDma.h, 118

ALT\_AVALON\_DMA\_NSLOTS\_MSK

AlteraDma.h, 118

ALT\_AVALON\_DMA\_RX\_STREAM

AlteraDma.h, 118

ALT\_AVALON\_DMA\_TX\_STREAM

AlteraDma.h, 118

ALT\_AVALON\_SPI\_COMMAND\_MERGE

AlteraSpi.h, 121

ALT\_AVALON\_SPI\_COMMAND\_TOGGLE\_SS\_N

AlteraSpi.h, 121

ALT\_DMA\_GET\_MODE

AlteraDma.h, 118

ALT\_DMA\_RX\_STREAM\_OFF

AlteraDma.h, 119

ALT\_DMA\_RX\_STREAM\_ON

AlteraDma.h, 119

ALT\_DMA\_SET\_MODE\_128

AlteraDma.h, 119

ALT\_DMA\_SET\_MODE\_16

AlteraDma.h, 119

ALT\_DMA\_SET\_MODE\_32

AlteraDma.h, 119

ALT\_DMA\_SET\_MODE\_64

AlteraDma.h, 119

ALT\_DMA\_SET\_MODE\_8

AlteraDma.h, 119

ALT\_DMA\_TX\_STREAM\_OFF

AlteraDma.h, 120

ALT\_DMA\_TX\_STREAM\_ON

AlteraDma.h, 120

alt\_rxchan\_done

AlteraDma.h, 120

alt\_txchan\_done

AlteraDma.h, 120

ALTERA\_AVALON\_DMA\_CONTROL\_BYTE\_MSK

AlteraDma.h, 120

ALTERA\_AVALON\_DMA\_CONTROL\_DWORD\_MSK

AlteraDma.h, 120

ALTERA\_AVALON\_DMA\_CONTROL\_GO\_MSK

AlteraDma.h, 120

ALTERA\_AVALON\_DMA\_CONTROL\_HW\_MSK

AlteraDma.h, 120

ALTERA\_AVALON\_DMA\_CONTROL\_I\_EN\_MSK

AlteraDma.h, 120

ALTERA\_AVALON\_DMA\_CONTROL\_LEEN\_MSK

AlteraDma.h, 120

ALTERA\_AVALON\_DMA\_CONTROL\_QWORD\_MSK

AlteraDma.h, 120

ALTERA\_AVALON\_DMA\_CONTROL\_RCON\_MSK

AlteraDma.h, 120

ALTERA\_AVALON\_DMA\_CONTROL\_REEN\_MSK

AlteraDma.h, 120

ALTERA\_AVALON\_DMA\_CONTROL\_SOFTWARERESET\_MSK

AlteraDma.h, 120

ALTERA\_AVALON\_DMA\_CONTROL\_WCON\_MSK

AlteraDma.h, 120

ALTERA\_AVALON\_DMA\_CONTROL\_WEEN\_MSK

AlteraDma.h, 120

ALTERA\_AVALON\_DMA\_CONTROL\_WORD\_MSK

AlteraDma.h, 120

ALTERA\_AVALON\_DMA\_STATUS\_BUSY\_MSK

AlteraDma.h, 120

ALTERA\_AVALON\_DMA\_STATUS\_DONE\_MSK

AlteraDma.h, 120

ALTERA\_AVALON\_DMA\_STATUS\_LEN\_MSK

AlteraDma.h, 120

ALTERA\_AVALON\_DMA\_STATUS\_REOP\_MSK

AlteraDma.h, 120

ALTERA\_AVALON\_DMA\_STATUS\_WEOP\_MSK

AlteraDma.h, 120

ALTERA\_AVALON\_SPI\_CONTROL\_IE\_MSK

AlteraSpi.h, 122

ALTERA\_AVALON\_SPI\_CONTROL\_IE\_OFST

AlteraSpi.h, 122

ALTERA\_AVALON\_SPI\_CONTROL\_IROE\_MSK

AlteraSpi.h, 122

ALTERA\_AVALON\_SPI\_CONTROL\_IROE\_OFST

AlteraSpi.h, 122

ALTERA\_AVALON\_SPI\_CONTROL\_IRRDY\_MSK

AlteraSpi.h, 122

ALTERA\_AVALON\_SPI\_CONTROL\_IRRDY\_OFS

AlteraSpi.h, 122

ALTERA\_AVALON\_SPI\_CONTROL\_ITOE\_MSK

AlteraSpi.h, 122

ALTERA\_AVALON\_SPI\_CONTROL\_ITOE\_OFST

AlteraSpi.h, 122

ALTERA\_AVALON\_SPI\_CONTROL\_ITRDY\_MSK

AlteraSpi.h, 122

ALTERA\_AVALON\_SPI\_CONTROL\_ITRDY\_OFS

AlteraSpi.h, 122

ALTERA\_AVALON\_SPI\_CONTROL\_SSO\_MSK

AlteraSpi.h, 122

ALTERA\_AVALON\_SPI\_CONTROL\_SSO\_OFST

AlteraSpi.h, 122

ALTERA\_AVALON\_SPI\_STATUS\_E\_MSK

AlteraSpi.h, 122

ALTERA\_AVALON\_SPI\_STATUS\_E\_OFST

AlteraSpi.h, 122

AlteraDma, 16

AlteraDma, 16

controlToString, 17

getLength, 17

getStatus, 17

ioctl, 17

irq, 17

launch\_bidir, 17

launch\_rxonly, 17

launch\_txonly, 17

prepare, 17

print, 17

reset, 17

rx\_ioctl, 17

send, 17

space, 17

statusToString, 17

tx\_ioctl, 17

AlteraDma.h

ALT\_AVALON\_DMA\_MODE\_128, 118

ALT\_AVALON\_DMA\_MODE\_16, 118

ALT\_AVALON\_DMA\_MODE\_32, 118

ALT\_AVALON\_DMA\_MODE\_64, 118

ALT\_AVALON\_DMA\_MODE\_8, 118

ALT\_AVALON\_DMA\_MODE\_MSK, 118

ALT\_AVALON\_DMA\_NSLOTS, 118

ALT\_AVALON\_DMA\_NSLOTS\_MSK, 118

ALT\_AVALON\_DMA\_RX\_STREAM, 118

ALT\_AVALON\_DMA\_TX\_STREAM, 118

ALT\_DMA\_GET\_MODE, 118

ALT\_DMA\_RX\_STREAM\_OFF, 119

ALT\_DMA\_RX\_STREAM\_ON, 119

ALT\_DMA\_SET\_MODE\_128, 119

ALT\_DMA\_SET\_MODE\_16, 119

ALT\_DMA\_SET\_MODE\_32, 119

ALT\_DMA\_SET\_MODE\_64, 119

ALT\_DMA\_SET\_MODE\_8, 119

ALT\_DMA\_TX\_STREAM\_OFF, 120

ALT\_DMA\_TX\_STREAM\_ON, 120

alt\_rxchan\_done, 120

alt\_txchan\_done, 120

ALTERA\_AVALON\_DMA\_CONTROL\_BYTE\_MSK, 120

ALTERA\_AVALON\_DMA\_CONTROL\_DWORD\_MSK, 120

ALTERA\_AVALON\_DMA\_CONTROL\_GO\_MSK, 120

ALTERA\_AVALON\_DMA\_CONTROL\_HW\_MSK, 120

ALTERA\_AVALON\_DMA\_CONTROL\_I\_EN\_MSK, 120

ALTERA\_AVALON\_DMA\_CONTROL\_LEEN\_MSK, 120

ALTERA\_AVALON\_DMA\_CONTROL\_QWORD\_MSK, 120

ALTERA\_AVALON\_DMA\_CONTROL\_RCON\_MSK, 120

ALTERA\_AVALON\_DMA\_CONTROL\_REEN\_MSK, 120

ALTERA\_AVALON\_DMA\_CONTROL\_SOFTWARERESET\_MSK, 120

ALTERA\_AVALON\_DMA\_CONTROL\_WCON\_MSK, 120

ALTERA\_AVALON\_DMA\_CONTROL\_WEEN\_MSK, 120

ALTERA\_AVALON\_DMA\_CONTROL\_WORD\_MSK, 120

ALTERA\_AVALON\_DMA\_STATUS\_BUSY\_MSK, 120

ALTERA\_AVALON\_DMA\_STATUS\_DONE\_MSK, 120

ALTERA\_AVALON\_DMA\_STATUS\_LEN\_MSK, 120

ALTERA\_AVALON\_DMA\_STATUS\_REOP\_MSK, 120

ALTERA\_AVALON\_DMA\_STATUS\_WEOP\_MSK, 120

AlteraSpi, 18

AlteraSpi, 19

base, 20

control\_index, 20

getRxData, 19

getStatus, 19

resetStatus, 19

rxData\_index, 20

selectSlave, 19

sendSpiCommand, 19

setControl, 20

setTxData, 20

slaveSelect\_index, 20

status\_Index, 20

status\_ROE\_mask, 20

status\_RRDY\_mask, 20

status\_TMT\_mask, 20

status\_TOE\_mask, 20

status\_TRDY\_mask, 20

txData\_index, 21

wordSize, 21

AlteraSpi.h

ALT\_AVALON\_SPI\_COMMAND\_MERGE, 121

ALT\_AVALON\_SPI\_COMMAND\_TOGGLE\_SS\_N, 121

ALTERA\_AVALON\_SPI\_CONTROL\_IE\_MSK, 122

ALTERA\_AVALON\_SPI\_CONTROL\_IE\_OFST, 122

ALTERA\_AVALON\_SPI\_CONTROL\_IROE\_MSK, 122

ALTERA\_AVALON\_SPI\_CONTROL\_IROE\_OFST, 122

ALTERA\_AVALON\_SPI\_CONTROL\_IRRDY\_MSK, 122

ALTERA\_AVALON\_SPI\_CONTROL\_IRRDY\_OFS, 122

ALTERA\_AVALON\_SPI\_CONTROL\_ITOE\_MSK, 122

ALTERA\_AVALON\_SPI\_CONTROL\_ITOE\_OFST, 122

ALTERA\_AVALON\_SPI\_CONTROL\_ITRDY\_MSK, 122

ALTERA\_AVALON\_SPI\_CONTROL\_ITRDY\_OFS, 122

ALTERA\_AVALON\_SPI\_CONTROL\_SSO\_MSK, 122

ALTERA\_AVALON\_SPI\_CONTROL\_SSO\_OFST, 122

ALTERA\_AVALON\_SPI\_STATUS\_E\_MSK, 122

ALTERA\_AVALON\_SPI\_STATUS\_E\_OFST, 122

ANMF

TCAN4x5x\_MCAN\_RX\_Header, 92

ARA

TCAN4x5x\_MCAN\_Interrupts, 83

ARAE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 77

ASM

TCAN4x5x\_MCAN\_CCCR\_Config, 68

bar0

AlphiBoard, 14

bar2

AlphiBoard, 14

bar3

AlphiBoard, 14

base

AlteraSpi, 20

ParallelInput, 32

BEC

TCAN4x5x\_MCAN\_Interrupts, 83

BECE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 77

BEU

TCAN4x5x\_MCAN\_Interrupts, 83

BEUE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 77

BO

TCAN4x5x\_MCAN\_Interrupts, 83

BoardVersion, 22

BoardVersion, 22

getTimeStamp, 22

getVersion, 22

BOE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 77

BRS

TCAN4x5x\_MCAN\_RX\_Header, 92

TCAN4x5x\_MCAN\_TX\_Header, 96

BRSE

TCAN4x5x\_MCAN\_CCCR\_Config, 68

BUFF\_LEN

TCAN4550, 55

bufLength

TransferDesc, 106

C:/Alphi/PCIeMiniSoftware/include/AlphiBoard.h, 108

C:/Alphi/PCIeMiniSoftware/include/AlphiDll.h, 111

C:/Alphi/PCIeMiniSoftware/include/AlphiErrorCodes.h, 112

C:/Alphi/PCIeMiniSoftware/include/AlteraDma.h, 117

C:/Alphi/PCIeMiniSoftware/include/AlteraSpi.h, 121

C:/Alphi/PCIeMiniSoftware/include/IrigDecoder.h, 123

C:/Alphi/PCIeMiniSoftware/include/ParallelInput.h, 124

C:/Alphi/PCIeMiniSoftware/include/PcieCra.h, 125

C:/Alphi/PCIeMiniSoftware/include/PCIeMini\_CAN\_FD.h, 126

C:/Alphi/PCIeMiniSoftware/include/TCAN4550.h, 128

C:/Alphi/PCIeMiniSoftware/include/TCAN4x5x\_Data\_Structs.h, 131

C:/Alphi/PCIeMiniSoftware/include/TCAN4x5x\_Reg.h, 136

C:/Alphi/PCIeMiniSoftware/include/TCAN4x5x\_SPI.h, 164

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/AlphiBoard.cpp, 170

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/AlphiBoard\_dma.cpp, 171

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/AlphiBoard\_irq.cpp, 172

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/AlteraDma.cpp, 173

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/AlteraSpi.cpp, 174

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/PcieCra.cpp, 175

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/PCIeMini\_error.cpp, 176

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/TestProgram.cpp, 177

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/x64/Debug/CodeAnalysisResultManifest.txt, 178

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/x64/Debug/PCIeMini\_lib.vcxproj.FileListAbsolute.txt, 179

C:/Alphi/PCIeMiniSoftware/PCIeMini\_lib/x64/Release/PCIeMini\_lib.vcxproj.FileListAbsolute.txt, 180

can

PCIeMini\_CAN\_FD, 39

TCAN4550, 55

CANBUSBAT

TCAN4x5x\_Device\_Interrupts, 63

CANBUSBATEN

TCAN4x5x\_Device\_Interrupt\_Enable, 58

CANBUSGND

TCAN4x5x\_Device\_Interrupts, 63

CANBUSGNDEN

TCAN4x5x\_Device\_Interrupt\_Enable, 58

CANBUSNORM

TCAN4x5x\_Device\_Interrupts, 63

CANBUSNORMEN

TCAN4x5x\_Device\_Interrupt\_Enable, 58

CANBUSOPEN

TCAN4x5x\_Device\_Interrupts, 63

CANBUSOPENEN

TCAN4x5x\_Device\_Interrupt\_Enable, 58

CANBUSTERMOPEN

TCAN4x5x\_Device\_Interrupts, 63

CANBUSTERMOPENEN

TCAN4x5x\_Device\_Interrupt\_Enable, 58

CANDOM

TCAN4x5x\_Device\_Interrupts, 63

CANDOMEN

TCAN4x5x\_Device\_Interrupt\_Enable, 58

CANERR

TCAN4x5x\_Device\_Interrupts, 64

CANHBAT

TCAN4x5x\_Device\_Interrupts, 64

CANHBATEN

TCAN4x5x\_Device\_Interrupt\_Enable, 58

CANHCANL

TCAN4x5x\_Device\_Interrupts, 64

CANHCANLEN

TCAN4x5x\_Device\_Interrupt\_Enable, 58

CANINT

TCAN4x5x\_Device\_Interrupts, 64

CANINTEN

TCAN4x5x\_Device\_Interrupt\_Enable, 58

CANLGND

TCAN4x5x\_Device\_Interrupts, 64

CANLGNDEN

TCAN4x5x\_Device\_Interrupt\_Enable, 58

canNios

PCIeMini\_CAN\_FD, 39

CANTO

TCAN4x5x\_Device\_Interrupts, 64

CANTOEN

TCAN4x5x\_Device\_Interrupt\_Enable, 58

CAP\_INPUT

ParallelInput, 32

CAP\_INPUT\_OUTPUT

ParallelInput, 32

CAP\_OUTPUT

ParallelInput, 32

CBF

TCAN4x5x\_Device\_Interrupts, 64

clearIrqStatus

ParallelInput, 31

clock\_gettime

AlphiBoard.h, 109

CLOCK\_REALTIME

AlphiBoard.h, 109

close

PCIeMini\_CAN\_FD, 38

Close

AlphiBoard, 10

control\_index

AlteraSpi, 20

controlRegister

PCIeMini\_CAN\_FD, 39

controlToString

AlteraDma, 17

cra

AlphiBoard, 15

CSR

TCAN4x5x\_MCAN\_CCCR\_Config, 68

DAR

TCAN4x5x\_MCAN\_CCCR\_Config, 68

data\_index

ParallelInput, 32

DataBitRatePrescaler

TCAN4x5x\_MCAN\_Data\_Timing\_Raw, 71

TCAN4x5x\_MCAN\_Data\_Timing\_Simple, 73

dataOut\_index

ParallelInput, 32

DataSyncJumpWidth

TCAN4x5x\_MCAN\_Data\_Timing\_Raw, 72

DataTimeSeg1andProp

TCAN4x5x\_MCAN\_Data\_Timing\_Raw, 72

DataTimeSeg2

TCAN4x5x\_MCAN\_Data\_Timing\_Raw, 72

DataTqAfterSamplePoint

TCAN4x5x\_MCAN\_Data\_Timing\_Simple, 73

DataTqBeforeSamplePoint

TCAN4x5x\_MCAN\_Data\_Timing\_Simple, 73

dest\_offset

TransferDesc, 106

Device\_ClearInterrupts

TCAN4550, 44

Device\_ClearInterruptsAll

TCAN4550, 44

Device\_ConfigureInterruptEnable

TCAN4550, 44

Device\_DisableTestMode

TCAN4550, 45

Device\_EnableTestMode

TCAN4550, 45

Device\_ReadDeviceIdent

TCAN4550, 45

Device\_ReadDeviceVersion

TCAN4550, 45

Device\_ReadInterruptEnable

TCAN4550, 45

Device\_ReadInterrupts

TCAN4550, 46

Device\_ReadMode

TCAN4550, 46

Device\_ReadTestMode

TCAN4550, 46

Device\_SetMode

TCAN4550, 46

direction\_index

ParallelInput, 32

disableInterrupts

AlphiBoard, 10

disableIrq

TCAN4550, 46

DLCode

TCAN4x5x\_MCAN\_RX\_Header, 92

TCAN4x5x\_MCAN\_TX\_Header, 97

Dll

AlphiDll.h, 111

DllMain

dllmain.cpp, 165

dllmain.cpp, 165

DllMain, 165

dma

PCIeMini\_CAN\_FD, 40

DMAClose

AlphiBoard, 11

DMAOpen

AlphiBoard, 11

DMARoutine

AlphiBoard, 11

DMATransfer

AlphiBoard, 11

doesItemExists

AlphiBoard\_irq.cpp, 172

dpr

PCIeMini\_CAN\_FD, 40

dpr\_length

PCIeMini\_CAN\_FD, 40

dpr\_offset

PCIeMini\_CAN\_FD, 40

DRX

TCAN4x5x\_MCAN\_Interrupts, 83

DRXE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 77

dwCounter

MINIPCIE\_INT\_RESULT, 28

dwEnabledIntType

MINIPCIE\_INT\_RESULT, 28

dwLastMessage

MINIPCIE\_INT\_RESULT, 28

dwLost

MINIPCIE\_INT\_RESULT, 28

ECCERR

TCAN4x5x\_Device\_Interrupts, 64

ECCERREN

TCAN4x5x\_Device\_Interrupt\_Enable, 59

edgeReg\_Index

ParallelInput, 32

EFBI

TCAN4x5x\_MCAN\_CCCR\_Config, 68

EFC

TCAN4x5x\_MCAN\_TX\_Header, 97

EFEC

TCAN4x5x\_MCAN\_XID\_Filter, 98

EFID1

TCAN4x5x\_MCAN\_XID\_Filter, 98

EFID2

TCAN4x5x\_MCAN\_XID\_Filter, 98

EFT

TCAN4x5x\_MCAN\_XID\_Filter, 98

ELO

TCAN4x5x\_MCAN\_Interrupts, 83

ELOE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 77

enableInterrupts

AlphiBoard, 11

enableIrq

TCAN4550, 46

EP

TCAN4x5x\_MCAN\_Interrupts, 83

EPE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 77

ERRCODE\_BOARD\_NOT\_PRESENT

AlphiErrorCodes.h, 114

ERRCODE\_BUSY

AlphiErrorCodes.h, 114

ERRCODE\_FAILED\_SELF\_TEST

AlphiErrorCodes.h, 114

ERRCODE\_INPUT\_MODE

AlphiErrorCodes.h, 114

ERRCODE\_INT\_ALREADY\_ENABLED

AlphiErrorCodes.h, 114

ERRCODE\_INT\_NOT\_ENABLED

AlphiErrorCodes.h, 114

ERRCODE\_INTERNAL\_ERROR

AlphiErrorCodes.h, 114

ERRCODE\_INVALID\_ALIGNMENT

AlphiErrorCodes.h, 114

ERRCODE\_INVALID\_BOARD\_NUM

AlphiErrorCodes.h, 114

ERRCODE\_INVALID\_CHANNEL\_NUM

AlphiErrorCodes.h, 114

ERRCODE\_INVALID\_COMMAND

AlphiErrorCodes.h, 114

ERRCODE\_INVALID\_DATALENGTH

AlphiErrorCodes.h, 114

ERRCODE\_INVALID\_DATARATE

AlphiErrorCodes.h, 114

ERRCODE\_INVALID\_EMTINTSTS

AlphiErrorCodes.h, 114

ERRCODE\_INVALID\_FINTSTS

AlphiErrorCodes.h, 114

ERRCODE\_INVALID\_FREQUENCY

AlphiErrorCodes.h, 114

ERRCODE\_INVALID\_GROUP

AlphiErrorCodes.h, 114

ERRCODE\_INVALID\_HANDLE

AlphiErrorCodes.h, 114

ERRCODE\_INVALID\_HFINTSTS

AlphiErrorCodes.h, 114

ERRCODE\_INVALID\_INPUT\_MODE

AlphiErrorCodes.h, 114

ERRCODE\_INVALID\_LABELNUMBER

AlphiErrorCodes.h, 114

ERRCODE\_INVALID\_LOGIC\_SEL

AlphiErrorCodes.h, 114

ERRCODE\_INVALID\_MASK\_VALUE

AlphiErrorCodes.h, 114

ERRCODE\_INVALID\_MODE

AlphiErrorCodes.h, 114

ERRCODE\_INVALID\_PARITY

AlphiErrorCodes.h, 115

ERRCODE\_INVALID\_RECVNUMBER

AlphiErrorCodes.h, 115

ERRCODE\_INVALID\_RECVSTATUS

AlphiErrorCodes.h, 115

ERRCODE\_INVALID\_SDI

AlphiErrorCodes.h, 115

ERRCODE\_INVALID\_SELF\_TEST\_DATA

AlphiErrorCodes.h, 115

ERRCODE\_INVALID\_SELF\_TEST\_ENABLE\_VAL

AlphiErrorCodes.h, 115

ERRCODE\_INVALID\_SSM

AlphiErrorCodes.h, 115

ERRCODE\_INVALID\_STROBE\_MODE

AlphiErrorCodes.h, 115

ERRCODE\_INVALID\_TIME\_BOUNCE\_VAL

AlphiErrorCodes.h, 115

ERRCODE\_INVALID\_TRANRECVSTS

AlphiErrorCodes.h, 115

ERRCODE\_INVALID\_TRANSNUMBER

AlphiErrorCodes.h, 115

ERRCODE\_INVALID\_TRANSSTATUS

AlphiErrorCodes.h, 115

ERRCODE\_INVALID\_TRPAIR

AlphiErrorCodes.h, 115

ERRCODE\_INVALID\_VALUE

AlphiErrorCodes.h, 115

ERRCODE\_NO\_ERROR

AlphiErrorCodes.h, 115

ERRCODE\_OUTPUT\_MODE

AlphiErrorCodes.h, 115

ERRCODE\_RX\_UNDERFLOW

AlphiErrorCodes.h, 115

ERRCODE\_SELF\_TEST\_DISABLE

AlphiErrorCodes.h, 115

ERRCODE\_SUCCESS

AlphiErrorCodes.h, 115

ERRCODE\_TIMEOUT

AlphiErrorCodes.h, 115

ERRCODE\_TX\_OVERFLOW

AlphiErrorCodes.h, 115

ErrLog

AlphiBoard.h, 109

ESI

TCAN4x5x\_MCAN\_RX\_Header, 92

TCAN4x5x\_MCAN\_TX\_Header, 97

EW

TCAN4x5x\_MCAN\_Interrupts, 84

EWE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 78

FALSE

AlphiDll.h, 111

FDF

TCAN4x5x\_MCAN\_RX\_Header, 92

TCAN4x5x\_MCAN\_TX\_Header, 97

FDOE

TCAN4x5x\_MCAN\_CCCR\_Config, 69

FIDX

TCAN4x5x\_MCAN\_RX\_Header, 92

flags

TransferDesc, 107

fPolling

TransferDesc, 107

FRAME\_OVF

TCAN4x5x\_Device\_Interrupts, 64

FRAME\_OVFEN

TCAN4x5x\_Device\_Interrupt\_Enable, 59

funcDiagEventHandler

MINIPCIE\_DEV\_CTX, 27

funcDiagIntHandler

MINIPCIE\_DEV\_CTX, 27

getAddress

TcanInterface, 104

getAlphiErrorMsg

AlphiErrorCodes.h, 116

PCIeMini\_error.cpp, 176

getBar0Address

AlphiBoard, 11

getBar2Address

AlphiBoard, 11

getBar3Address

AlphiBoard, 12

getData

ParallelInput, 31

getDay

IrigDecoder, 25

getDayRaw

IrigDecoder, 25

getFpgaID

AlphiBoard, 12

getFpgaTimeStamp

AlphiBoard, 12

getIntResults

AlphiBoard, 12

getIrigDate

IrigDecoder, 25

getIrqEnable

ParallelInput, 31

getIrqEnableMask

PcieCra, 34

getIrqStatus

ParallelInput, 31

PcieCra, 34

getLength

AlteraDma, 17

getMappedAddress

PcieCra, 34

getRxData

AlteraSpi, 19

getSecond

IrigDecoder, 25

getStatus

AlteraDma, 17

AlteraSpi, 19

TcanInterface, 104

getTime

IrigDecoder, 25

getTimeRaw

IrigDecoder, 25

getTimeStamp

BoardVersion, 22

getVersion

BoardVersion, 22

GLOBALERR

TCAN4x5x\_Device\_Interrupts, 64

hookInterruptServiceRoutine

AlphiBoard, 12

HPM

TCAN4x5x\_MCAN\_Interrupts, 84

HPME

TCAN4x5x\_MCAN\_Interrupt\_Enable, 78

HRESULT

AlphiDll.h, 111

hwDMAInterruptDisable

AlphiBoard, 13

PCIeMini\_CAN\_FD, 38

hwDMAInterruptEnable

AlphiBoard, 13

PCIeMini\_CAN\_FD, 38

hwDMAProgram

AlphiBoard, 13

PCIeMini\_CAN\_FD, 38

hwDMAStart

AlphiBoard, 13

PCIeMini\_CAN\_FD, 38

hwDMAWaitForCompletion

AlphiBoard, 13

PCIeMini\_CAN\_FD, 39

ID

TCAN4x5x\_MCAN\_RX\_Header, 92

TCAN4x5x\_MCAN\_TX\_Header, 97

input0

PCIeMini\_CAN\_FD, 40

input1

PCIeMini\_CAN\_FD, 40

ioctl

AlteraDma, 17

irig

PCIeMini\_CAN\_FD, 40

IrigDecoder, 25

getDay, 25

getDayRaw, 25

getIrigDate, 25

getSecond, 25

getTime, 25

getTimeRaw, 25

IrigDecoder, 25

IrigDecoder::IrigDate, 24

tm\_hour, 24

tm\_min, 24

tm\_sec, 24

tm\_yday, 24

tm\_year, 24

irq

AlteraDma, 17

irqDelay\_index

ParallelInput, 32

irqEnable\_index

ParallelInput, 32

irqStatus\_index

ParallelInput, 32

IsValidDevice

AlphiBoard, 13

launch\_bidir

AlteraDma, 17

launch\_rxonly

AlteraDma, 17

launch\_txonly

AlteraDma, 17

ledPio

PCIeMini\_CAN\_FD, 40

Length

LinearAddress, 26

libStatus

AlphiBoard, 15

LinearAddress, 26

Address, 26

AlphiBoard.h, 109

Length, 26

LINUX

AlphiDll.h, 111

LWU

TCAN4x5x\_Device\_Interrupts, 64

LWUEN

TCAN4x5x\_Device\_Interrupt\_Enable, 59

M\_CAN\_INT

TCAN4x5x\_Device\_Interrupts, 64

MCAN\_ClearInterrupts

TCAN4550, 46

MCAN\_ClearInterruptsAll

TCAN4550, 47

MCAN\_ConfigureCCCRRegister

TCAN4550, 47

MCAN\_ConfigureDataTiming\_Raw

TCAN4550, 47

MCAN\_ConfigureDataTiming\_Simple

TCAN4550, 48

MCAN\_ConfigureInterruptEnable

TCAN4550, 48

MCAN\_ConfigureNominalTiming\_Raw

TCAN4550, 48

MCAN\_ConfigureNominalTiming\_Simple

TCAN4550, 49

MCAN\_DisableProtectedRegisters

TCAN4550, 49

MCAN\_DLC\_0B

TCAN4x5x\_Reg.h, 145

MCAN\_DLC\_12B

TCAN4x5x\_Reg.h, 145

MCAN\_DLC\_16B

TCAN4x5x\_Reg.h, 145

MCAN\_DLC\_1B

TCAN4x5x\_Reg.h, 145

MCAN\_DLC\_20B

TCAN4x5x\_Reg.h, 145

MCAN\_DLC\_24B

TCAN4x5x\_Reg.h, 145

MCAN\_DLC\_2B

TCAN4x5x\_Reg.h, 145

MCAN\_DLC\_32B

TCAN4x5x\_Reg.h, 145

MCAN\_DLC\_3B

TCAN4x5x\_Reg.h, 145

MCAN\_DLC\_48B

TCAN4x5x\_Reg.h, 145

MCAN\_DLC\_4B

TCAN4x5x\_Reg.h, 145

MCAN\_DLC\_5B

TCAN4x5x\_Reg.h, 145

MCAN\_DLC\_64B

TCAN4x5x\_Reg.h, 145

MCAN\_DLC\_6B

TCAN4x5x\_Reg.h, 145

MCAN\_DLC\_7B

TCAN4x5x\_Reg.h, 145

MCAN\_DLC\_8B

TCAN4x5x\_Reg.h, 145

MCAN\_DLCtoBytes

TCAN4550, 49

MCAN\_EnableProtectedRegisters

TCAN4550, 49

MCAN\_ReadCCCRRegister

TCAN4550, 50

MCAN\_ReadDataTimingFD\_Raw

TCAN4550, 50

MCAN\_ReadDataTimingFD\_Simple

TCAN4550, 50

MCAN\_ReadInterruptEnable

TCAN4550, 50

MCAN\_ReadInterrupts

TCAN4550, 50

MCAN\_ReadNextFIFO

TCAN4550, 51

MCAN\_ReadNominalTiming\_Raw

TCAN4550, 51

MCAN\_ReadNominalTiming\_Simple

TCAN4550, 51

MCAN\_ReadRXBuffer

TCAN4550, 51

MCAN\_TransmitBufferContents

TCAN4550, 52

MCAN\_TXRXESC\_DataByteValue

TCAN4550, 52

MCAN\_WriteSIDFilter

TCAN4550, 52

MCAN\_WriteTXBuffer

TCAN4550, 53

MCAN\_WriteXIDFilter

TCAN4550, 53

mddr

PCIeMini\_CAN\_FD, 40

MINIPCIE\_ARINC429\_DEFAULT\_DRIVER\_NAME

AlphiBoard.cpp, 170

MINIPCIE\_ARINC429\_DEFAULT\_LICENSE\_STRING

AlphiBoard.cpp, 170

MINIPCIE\_ARINC429\_IntHandler

AlphiBoard\_irq.cpp, 172

MINIPCIE\_DEV\_CTX, 27

funcDiagEventHandler, 27

funcDiagIntHandler, 27

userData, 27

MINIPCIE\_EVENT\_HANDLER

AlphiBoard.h, 109

MINIPCIE\_INT\_HANDLER

AlphiBoard.h, 109

MINIPCIE\_INT\_RESULT, 28

dwCounter, 28

dwEnabledIntType, 28

dwLastMessage, 28

dwLost, 28

waitResult, 28

MM

TCAN4x5x\_MCAN\_TX\_Header, 97

MON

TCAN4x5x\_MCAN\_CCCR\_Config, 69

MRAF

TCAN4x5x\_MCAN\_Interrupts, 84

MRAFE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 78

MRAM\_12\_Byte\_Data

TCAN4x5x\_Data\_Structs.h, 133

MRAM\_16\_Byte\_Data

TCAN4x5x\_Data\_Structs.h, 133

MRAM\_20\_Byte\_Data

TCAN4x5x\_Data\_Structs.h, 133

MRAM\_24\_Byte\_Data

TCAN4x5x\_Data\_Structs.h, 133

MRAM\_32\_Byte\_Data

TCAN4x5x\_Data\_Structs.h, 133

MRAM\_48\_Byte\_Data

TCAN4x5x\_Data\_Structs.h, 133

MRAM\_64\_Byte\_Data

TCAN4x5x\_Data\_Structs.h, 133

MRAM\_8\_Byte\_Data

TCAN4x5x\_Data\_Structs.h, 133

MRAM\_Clear

TCAN4550, 53

MRAM\_Configure

TCAN4550, 53

MRAM\_SIZE

TCAN4x5x\_Reg.h, 145

msgBufferIn

TCAN4550, 55

msgBufferOut

TCAN4550, 55

msgLength

TCAN4550, 55

MsSleep

AlphiBoard, 13

MyDmaIntHandler

AlphiBoard\_dma.cpp, 171

nanosleep

AlphiBoard.h, 109

nbrOfCanInterfaces

PCIeMini\_CAN\_FD, 40

NISO

TCAN4x5x\_MCAN\_CCCR\_Config, 69

NominalBitRatePrescaler

TCAN4x5x\_MCAN\_Nominal\_Timing\_Raw, 87

TCAN4x5x\_MCAN\_Nominal\_Timing\_Simple, 89

NominalSyncJumpWidth

TCAN4x5x\_MCAN\_Nominal\_Timing\_Raw, 87

NominalTimeSeg1andProp

TCAN4x5x\_MCAN\_Nominal\_Timing\_Raw, 88

NominalTimeSeg2

TCAN4x5x\_MCAN\_Nominal\_Timing\_Raw, 88

NominalTqAfterSamplePoint

TCAN4x5x\_MCAN\_Nominal\_Timing\_Simple, 89

NominalTqBeforeSamplePoint

TCAN4x5x\_MCAN\_Nominal\_Timing\_Simple, 89

NUM\_TRANS\_CMDS

AlphiBoard\_irq.cpp, 172

open

PCIeMini\_CAN\_FD, 39

Open

AlphiBoard, 13

ParallelInput, 30

base, 32

CAP\_INPUT, 32

CAP\_INPUT\_OUTPUT, 32

CAP\_OUTPUT, 32

clearIrqStatus, 31

data\_index, 32

dataOut\_index, 32

direction\_index, 32

edgeReg\_Index, 32

getData, 31

getIrqEnable, 31

getIrqStatus, 31

irqDelay\_index, 32

irqEnable\_index, 32

irqStatus\_index, 32

ParallelInput, 31

polarity\_index, 32

reset, 31

resetIrq, 31

setIrqDisable, 31

setIrqEnable, 31

PCIe\_Mini\_CAN\_FD.cpp, 166

PcieCra, 33

getIrqEnableMask, 34

getIrqStatus, 34

getMappedAddress, 34

PcieCra, 34

reset, 34

setIrqEnableMask, 34

setTrEntry, 34

setTxsAvlAddress, 35

PCIeMini\_CAN\_FD, 37

can, 39

canNios, 39

close, 38

controlRegister, 39

dma, 40

dpr, 40

dpr\_length, 40

dpr\_offset, 40

hwDMAInterruptDisable, 38

hwDMAInterruptEnable, 38

hwDMAProgram, 38

hwDMAStart, 38

hwDMAWaitForCompletion, 39

input0, 40

input1, 40

irig, 40

ledPio, 40

mddr, 40

nbrOfCanInterfaces, 40

open, 39

PCIeMini\_CAN\_FD, 38

reset, 39

PCIeMini\_CAN\_FD.h

PI\_GPIO1\_0, 127

PI\_GPIO1\_1, 127

PI\_GPIO1\_2, 127

PI\_GPIO1\_3, 127

PI\_GPO2\_0, 127

PI\_GPO2\_1, 127

PI\_GPO2\_2, 127

PI\_GPO2\_3, 127

PI\_nINT\_0, 127

PI\_nINT\_1, 127

PI\_nINT\_2, 127

PI\_nINT\_3, 127

PI\_nWAKE\_0, 127

PI\_nWAKE\_1, 127

PI\_nWAKE\_2, 127

PI\_nWAKE\_3, 127

PCIeMini\_error.cpp

getAlphiErrorMsg, 176

wdErrorToString, 176

PCIeMini\_status

AlphiErrorCodes.h, 115

pDma

AlphiBoard, 15

PEA

TCAN4x5x\_MCAN\_Interrupts, 84

PEAE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 78

PED

TCAN4x5x\_MCAN\_Interrupts, 84

PEDE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 78

PI\_GPIO1\_0

PCIeMini\_CAN\_FD.h, 127

PI\_GPIO1\_1

PCIeMini\_CAN\_FD.h, 127

PI\_GPIO1\_2

PCIeMini\_CAN\_FD.h, 127

PI\_GPIO1\_3

PCIeMini\_CAN\_FD.h, 127

PI\_GPO2\_0

PCIeMini\_CAN\_FD.h, 127

PI\_GPO2\_1

PCIeMini\_CAN\_FD.h, 127

PI\_GPO2\_2

PCIeMini\_CAN\_FD.h, 127

PI\_GPO2\_3

PCIeMini\_CAN\_FD.h, 127

PI\_nINT\_0

PCIeMini\_CAN\_FD.h, 127

PI\_nINT\_1

PCIeMini\_CAN\_FD.h, 127

PI\_nINT\_2

PCIeMini\_CAN\_FD.h, 127

PI\_nINT\_3

PCIeMini\_CAN\_FD.h, 127

PI\_nWAKE\_0

PCIeMini\_CAN\_FD.h, 127

PI\_nWAKE\_1

PCIeMini\_CAN\_FD.h, 127

PI\_nWAKE\_2

PCIeMini\_CAN\_FD.h, 127

PI\_nWAKE\_3

PCIeMini\_CAN\_FD.h, 127

PMINIPCIE\_DEV\_CTX

AlphiBoard.h, 109

polarity\_index

ParallelInput, 32

prepare

AlteraDma, 17

print

AlteraDma, 17

PWRON

TCAN4x5x\_Device\_Interrupts, 65

PWRONEN

TCAN4x5x\_Device\_Interrupt\_Enable, 59

PXHD

TCAN4x5x\_MCAN\_CCCR\_Config, 69

QT\_CORE\_LIB

AlphiBoard.cpp, 170

REG\_BITS\_DEVICE\_IE\_CANBUSBAT

TCAN4x5x\_Reg.h, 145

REG\_BITS\_DEVICE\_IE\_CANBUSGND

TCAN4x5x\_Reg.h, 145

REG\_BITS\_DEVICE\_IE\_CANBUSNOM

TCAN4x5x\_Reg.h, 145

REG\_BITS\_DEVICE\_IE\_CANBUSOPEN

TCAN4x5x\_Reg.h, 145

REG\_BITS\_DEVICE\_IE\_CANBUSTERMOPEN

TCAN4x5x\_Reg.h, 145

REG\_BITS\_DEVICE\_IE\_CANDOM

TCAN4x5x\_Reg.h, 145

REG\_BITS\_DEVICE\_IE\_CANHBAT

TCAN4x5x\_Reg.h, 145

REG\_BITS\_DEVICE\_IE\_CANHCANL

TCAN4x5x\_Reg.h, 145

REG\_BITS\_DEVICE\_IE\_CANINT

TCAN4x5x\_Reg.h, 146

REG\_BITS\_DEVICE\_IE\_CANLGND

TCAN4x5x\_Reg.h, 146

REG\_BITS\_DEVICE\_IE\_CANSLNT

TCAN4x5x\_Reg.h, 146

REG\_BITS\_DEVICE\_IE\_ECCERR

TCAN4x5x\_Reg.h, 146

REG\_BITS\_DEVICE\_IE\_FRAME\_OVF

TCAN4x5x\_Reg.h, 146

REG\_BITS\_DEVICE\_IE\_LWU

TCAN4x5x\_Reg.h, 146

REG\_BITS\_DEVICE\_IE\_MASK

TCAN4x5x\_Reg.h, 146

REG\_BITS\_DEVICE\_IE\_PWRON

TCAN4x5x\_Reg.h, 146

REG\_BITS\_DEVICE\_IE\_TSD

TCAN4x5x\_Reg.h, 146

REG\_BITS\_DEVICE\_IE\_UVCCOUT

TCAN4x5x\_Reg.h, 146

REG\_BITS\_DEVICE\_IE\_UVIO

TCAN4x5x\_Reg.h, 146

REG\_BITS\_DEVICE\_IE\_UVSUP

TCAN4x5x\_Reg.h, 146

REG\_BITS\_DEVICE\_IE\_WDTO

TCAN4x5x\_Reg.h, 146

REG\_BITS\_DEVICE\_IE\_WKERR

TCAN4x5x\_Reg.h, 146

REG\_BITS\_DEVICE\_IR\_CANBUSBAT

TCAN4x5x\_Reg.h, 146

REG\_BITS\_DEVICE\_IR\_CANBUSFAULT

TCAN4x5x\_Reg.h, 146

REG\_BITS\_DEVICE\_IR\_CANBUSGND

TCAN4x5x\_Reg.h, 146

REG\_BITS\_DEVICE\_IR\_CANBUSNOM

TCAN4x5x\_Reg.h, 146

REG\_BITS\_DEVICE\_IR\_CANBUSOPEN

TCAN4x5x\_Reg.h, 146

REG\_BITS\_DEVICE\_IR\_CANBUSTERMOPEN

TCAN4x5x\_Reg.h, 146

REG\_BITS\_DEVICE\_IR\_CANDOM

TCAN4x5x\_Reg.h, 146

REG\_BITS\_DEVICE\_IR\_CANERR

TCAN4x5x\_Reg.h, 146

REG\_BITS\_DEVICE\_IR\_CANHBAT

TCAN4x5x\_Reg.h, 146

REG\_BITS\_DEVICE\_IR\_CANHCANL

TCAN4x5x\_Reg.h, 146

REG\_BITS\_DEVICE\_IR\_CANINT

TCAN4x5x\_Reg.h, 146

REG\_BITS\_DEVICE\_IR\_CANLGND

TCAN4x5x\_Reg.h, 146

REG\_BITS\_DEVICE\_IR\_CANSLNT

TCAN4x5x\_Reg.h, 147

REG\_BITS\_DEVICE\_IR\_ECCERR

TCAN4x5x\_Reg.h, 147

REG\_BITS\_DEVICE\_IR\_FRAME\_OVF

TCAN4x5x\_Reg.h, 147

REG\_BITS\_DEVICE\_IR\_GLOBALERR

TCAN4x5x\_Reg.h, 147

REG\_BITS\_DEVICE\_IR\_LWU

TCAN4x5x\_Reg.h, 147

REG\_BITS\_DEVICE\_IR\_M\_CAN\_INT

TCAN4x5x\_Reg.h, 147

REG\_BITS\_DEVICE\_IR\_nWKRQ

TCAN4x5x\_Reg.h, 147

REG\_BITS\_DEVICE\_IR\_PWRON

TCAN4x5x\_Reg.h, 147

REG\_BITS\_DEVICE\_IR\_SPIERR

TCAN4x5x\_Reg.h, 147

REG\_BITS\_DEVICE\_IR\_SWERR

TCAN4x5x\_Reg.h, 147

REG\_BITS\_DEVICE\_IR\_TSD

TCAN4x5x\_Reg.h, 147

REG\_BITS\_DEVICE\_IR\_UVIO

TCAN4x5x\_Reg.h, 147

REG\_BITS\_DEVICE\_IR\_UVSUP

TCAN4x5x\_Reg.h, 147

REG\_BITS\_DEVICE\_IR\_VTWD

TCAN4x5x\_Reg.h, 147

REG\_BITS\_DEVICE\_IR\_WDTO

TCAN4x5x\_Reg.h, 147

REG\_BITS\_DEVICE\_IR\_WKERR

TCAN4x5x\_Reg.h, 147

REG\_BITS\_DEVICE\_MODE\_CLKOUT\_DIV1

TCAN4x5x\_Reg.h, 147

REG\_BITS\_DEVICE\_MODE\_CLKOUT\_DIV2

TCAN4x5x\_Reg.h, 147

REG\_BITS\_DEVICE\_MODE\_CLKOUT\_MASK

TCAN4x5x\_Reg.h, 147

REG\_BITS\_DEVICE\_MODE\_DEVICE\_RESET

TCAN4x5x\_Reg.h, 147

REG\_BITS\_DEVICE\_MODE\_DEVICEMODE\_MASK

TCAN4x5x\_Reg.h, 147

REG\_BITS\_DEVICE\_MODE\_DEVICEMODE\_NORMAL

TCAN4x5x\_Reg.h, 147

REG\_BITS\_DEVICE\_MODE\_DEVICEMODE\_SLEEP

TCAN4x5x\_Reg.h, 147

REG\_BITS\_DEVICE\_MODE\_DEVICEMODE\_STANDBY

TCAN4x5x\_Reg.h, 147

REG\_BITS\_DEVICE\_MODE\_FAIL\_SAFE\_DIS

TCAN4x5x\_Reg.h, 147

REG\_BITS\_DEVICE\_MODE\_FAIL\_SAFE\_EN

TCAN4x5x\_Reg.h, 147

REG\_BITS\_DEVICE\_MODE\_FAIL\_SAFE\_MASK

TCAN4x5x\_Reg.h, 148

REG\_BITS\_DEVICE\_MODE\_GPO1\_FUNC\_MASK

TCAN4x5x\_Reg.h, 148

REG\_BITS\_DEVICE\_MODE\_GPO1\_FUNC\_MCAN\_INT1

TCAN4x5x\_Reg.h, 148

REG\_BITS\_DEVICE\_MODE\_GPO1\_FUNC\_SPI\_INT

TCAN4x5x\_Reg.h, 148

REG\_BITS\_DEVICE\_MODE\_GPO1\_FUNC\_UVLO\_THERM

TCAN4x5x\_Reg.h, 148

REG\_BITS\_DEVICE\_MODE\_GPO1\_MODE\_CLKOUT

TCAN4x5x\_Reg.h, 148

REG\_BITS\_DEVICE\_MODE\_GPO1\_MODE\_GPI

TCAN4x5x\_Reg.h, 148

REG\_BITS\_DEVICE\_MODE\_GPO1\_MODE\_GPO

TCAN4x5x\_Reg.h, 148

REG\_BITS\_DEVICE\_MODE\_GPO1\_MODE\_MASK

TCAN4x5x\_Reg.h, 148

REG\_BITS\_DEVICE\_MODE\_GPO2\_CAN\_FAULT

TCAN4x5x\_Reg.h, 148

REG\_BITS\_DEVICE\_MODE\_GPO2\_MASK

TCAN4x5x\_Reg.h, 148

REG\_BITS\_DEVICE\_MODE\_GPO2\_MCAN\_INT0

TCAN4x5x\_Reg.h, 148

REG\_BITS\_DEVICE\_MODE\_GPO2\_NINT

TCAN4x5x\_Reg.h, 148

REG\_BITS\_DEVICE\_MODE\_GPO2\_WDT

TCAN4x5x\_Reg.h, 148

REG\_BITS\_DEVICE\_MODE\_INH\_DIS

TCAN4x5x\_Reg.h, 148

REG\_BITS\_DEVICE\_MODE\_INH\_EN

TCAN4x5x\_Reg.h, 148

REG\_BITS\_DEVICE\_MODE\_INH\_MASK

TCAN4x5x\_Reg.h, 148

REG\_BITS\_DEVICE\_MODE\_NWKRQ\_CONFIG\_INH

TCAN4x5x\_Reg.h, 148

REG\_BITS\_DEVICE\_MODE\_NWKRQ\_CONFIG\_MASK

TCAN4x5x\_Reg.h, 148

REG\_BITS\_DEVICE\_MODE\_NWKRQ\_CONFIG\_WKRQ

TCAN4x5x\_Reg.h, 148

REG\_BITS\_DEVICE\_MODE\_NWKRQ\_VOLT\_INTERNAL

TCAN4x5x\_Reg.h, 148

REG\_BITS\_DEVICE\_MODE\_NWKRQ\_VOLT\_MASK

TCAN4x5x\_Reg.h, 148

REG\_BITS\_DEVICE\_MODE\_NWKRQ\_VOLT\_VIO

TCAN4x5x\_Reg.h, 148

REG\_BITS\_DEVICE\_MODE\_SWE\_DIS

TCAN4x5x\_Reg.h, 148

REG\_BITS\_DEVICE\_MODE\_SWE\_EN

TCAN4x5x\_Reg.h, 148

REG\_BITS\_DEVICE\_MODE\_SWE\_MASK

TCAN4x5x\_Reg.h, 148

REG\_BITS\_DEVICE\_MODE\_TESTMODE\_CONTROLLER

TCAN4x5x\_Reg.h, 149

REG\_BITS\_DEVICE\_MODE\_TESTMODE\_DIS

TCAN4x5x\_Reg.h, 149

REG\_BITS\_DEVICE\_MODE\_TESTMODE\_EN

TCAN4x5x\_Reg.h, 149

REG\_BITS\_DEVICE\_MODE\_TESTMODE\_ENMASK

TCAN4x5x\_Reg.h, 149

REG\_BITS\_DEVICE\_MODE\_TESTMODE\_MASK

TCAN4x5x\_Reg.h, 149

REG\_BITS\_DEVICE\_MODE\_TESTMODE\_PHY

TCAN4x5x\_Reg.h, 149

REG\_BITS\_DEVICE\_MODE\_WAKE\_PIN\_BOTHEDGES

TCAN4x5x\_Reg.h, 149

REG\_BITS\_DEVICE\_MODE\_WAKE\_PIN\_DIS

TCAN4x5x\_Reg.h, 149

REG\_BITS\_DEVICE\_MODE\_WAKE\_PIN\_FALLING

TCAN4x5x\_Reg.h, 149

REG\_BITS\_DEVICE\_MODE\_WAKE\_PIN\_MASK

TCAN4x5x\_Reg.h, 149

REG\_BITS\_DEVICE\_MODE\_WAKE\_PIN\_RISING

TCAN4x5x\_Reg.h, 149

REG\_BITS\_DEVICE\_MODE\_WD\_CLK\_20MHZ

TCAN4x5x\_Reg.h, 149

REG\_BITS\_DEVICE\_MODE\_WD\_CLK\_40MHZ

TCAN4x5x\_Reg.h, 149

REG\_BITS\_DEVICE\_MODE\_WD\_CLK\_MASK

TCAN4x5x\_Reg.h, 149

REG\_BITS\_DEVICE\_MODE\_WD\_TIMER\_3S

TCAN4x5x\_Reg.h, 149

REG\_BITS\_DEVICE\_MODE\_WD\_TIMER\_600MS

TCAN4x5x\_Reg.h, 149

REG\_BITS\_DEVICE\_MODE\_WD\_TIMER\_60MS

TCAN4x5x\_Reg.h, 149

REG\_BITS\_DEVICE\_MODE\_WD\_TIMER\_6S

TCAN4x5x\_Reg.h, 149

REG\_BITS\_DEVICE\_MODE\_WD\_TIMER\_MASK

TCAN4x5x\_Reg.h, 149

REG\_BITS\_DEVICE\_MODE\_WDT\_ACTION\_INH\_PULSE

TCAN4x5x\_Reg.h, 149

REG\_BITS\_DEVICE\_MODE\_WDT\_ACTION\_INT

TCAN4x5x\_Reg.h, 149

REG\_BITS\_DEVICE\_MODE\_WDT\_ACTION\_MASK

TCAN4x5x\_Reg.h, 149

REG\_BITS\_DEVICE\_MODE\_WDT\_ACTION\_WDT\_PULSE

TCAN4x5x\_Reg.h, 149

REG\_BITS\_DEVICE\_MODE\_WDT\_DIS

TCAN4x5x\_Reg.h, 149

REG\_BITS\_DEVICE\_MODE\_WDT\_EN

TCAN4x5x\_Reg.h, 149

REG\_BITS\_DEVICE\_MODE\_WDT\_MASK

TCAN4x5x\_Reg.h, 149

REG\_BITS\_DEVICE\_MODE\_WDT\_RESET\_BIT

TCAN4x5x\_Reg.h, 150

REG\_BITS\_MCAN\_CCCR\_ASM

TCAN4x5x\_Reg.h, 150

REG\_BITS\_MCAN\_CCCR\_BRSE

TCAN4x5x\_Reg.h, 150

REG\_BITS\_MCAN\_CCCR\_CCE

TCAN4x5x\_Reg.h, 150

REG\_BITS\_MCAN\_CCCR\_CSA

TCAN4x5x\_Reg.h, 150

REG\_BITS\_MCAN\_CCCR\_CSR

TCAN4x5x\_Reg.h, 150

REG\_BITS\_MCAN\_CCCR\_DAR\_DIS

TCAN4x5x\_Reg.h, 150

REG\_BITS\_MCAN\_CCCR\_EFBI

TCAN4x5x\_Reg.h, 150

REG\_BITS\_MCAN\_CCCR\_FDOE

TCAN4x5x\_Reg.h, 150

REG\_BITS\_MCAN\_CCCR\_INIT

TCAN4x5x\_Reg.h, 150

REG\_BITS\_MCAN\_CCCR\_MON

TCAN4x5x\_Reg.h, 150

REG\_BITS\_MCAN\_CCCR\_NISO\_BOSCH

TCAN4x5x\_Reg.h, 150

REG\_BITS\_MCAN\_CCCR\_NISO\_ISO

TCAN4x5x\_Reg.h, 150

REG\_BITS\_MCAN\_CCCR\_PXHD\_DIS

TCAN4x5x\_Reg.h, 150

REG\_BITS\_MCAN\_CCCR\_RESERVED\_MASK

TCAN4x5x\_Reg.h, 150

REG\_BITS\_MCAN\_CCCR\_TEST

TCAN4x5x\_Reg.h, 150

REG\_BITS\_MCAN\_CCCR\_TXP

TCAN4x5x\_Reg.h, 150

REG\_BITS\_MCAN\_DBTP\_TDC\_EN

TCAN4x5x\_Reg.h, 150

REG\_BITS\_MCAN\_GFC\_ANFE\_FIFO0

TCAN4x5x\_Reg.h, 150

REG\_BITS\_MCAN\_GFC\_ANFE\_FIFO1

TCAN4x5x\_Reg.h, 150

REG\_BITS\_MCAN\_GFC\_ANFS\_FIFO0

TCAN4x5x\_Reg.h, 150

REG\_BITS\_MCAN\_GFC\_ANFS\_FIFO1

TCAN4x5x\_Reg.h, 150

REG\_BITS\_MCAN\_GFC\_RRFE

TCAN4x5x\_Reg.h, 150

REG\_BITS\_MCAN\_GFC\_RRFS

TCAN4x5x\_Reg.h, 150

REG\_BITS\_MCAN\_IE\_ARAE

TCAN4x5x\_Reg.h, 150

REG\_BITS\_MCAN\_IE\_ARAL

TCAN4x5x\_Reg.h, 150

REG\_BITS\_MCAN\_IE\_BECE

TCAN4x5x\_Reg.h, 151

REG\_BITS\_MCAN\_IE\_BECL

TCAN4x5x\_Reg.h, 151

REG\_BITS\_MCAN\_IE\_BEUE

TCAN4x5x\_Reg.h, 151

REG\_BITS\_MCAN\_IE\_BEUL

TCAN4x5x\_Reg.h, 151

REG\_BITS\_MCAN\_IE\_BOE

TCAN4x5x\_Reg.h, 151

REG\_BITS\_MCAN\_IE\_BOL

TCAN4x5x\_Reg.h, 151

REG\_BITS\_MCAN\_IE\_DRXE

TCAN4x5x\_Reg.h, 151

REG\_BITS\_MCAN\_IE\_DRXL

TCAN4x5x\_Reg.h, 151

REG\_BITS\_MCAN\_IE\_ELOE

TCAN4x5x\_Reg.h, 151

REG\_BITS\_MCAN\_IE\_ELOL

TCAN4x5x\_Reg.h, 151

REG\_BITS\_MCAN\_IE\_EPE

TCAN4x5x\_Reg.h, 151

REG\_BITS\_MCAN\_IE\_EPL

TCAN4x5x\_Reg.h, 151

REG\_BITS\_MCAN\_IE\_EWE

TCAN4x5x\_Reg.h, 151

REG\_BITS\_MCAN\_IE\_EWL

TCAN4x5x\_Reg.h, 151

REG\_BITS\_MCAN\_IE\_HPME

TCAN4x5x\_Reg.h, 151

REG\_BITS\_MCAN\_IE\_HPML

TCAN4x5x\_Reg.h, 151

REG\_BITS\_MCAN\_IE\_MRAFE

TCAN4x5x\_Reg.h, 151

REG\_BITS\_MCAN\_IE\_MRAFL

TCAN4x5x\_Reg.h, 151

REG\_BITS\_MCAN\_IE\_PEAE

TCAN4x5x\_Reg.h, 151

REG\_BITS\_MCAN\_IE\_PEAL

TCAN4x5x\_Reg.h, 151

REG\_BITS\_MCAN\_IE\_PEDE

TCAN4x5x\_Reg.h, 151

REG\_BITS\_MCAN\_IE\_PEDL

TCAN4x5x\_Reg.h, 151

REG\_BITS\_MCAN\_IE\_RF0FE

TCAN4x5x\_Reg.h, 151

REG\_BITS\_MCAN\_IE\_RF0FL

TCAN4x5x\_Reg.h, 151

REG\_BITS\_MCAN\_IE\_RF0LE

TCAN4x5x\_Reg.h, 151

REG\_BITS\_MCAN\_IE\_RF0LL

TCAN4x5x\_Reg.h, 151

REG\_BITS\_MCAN\_IE\_RF0NE

TCAN4x5x\_Reg.h, 152

REG\_BITS\_MCAN\_IE\_RF0NL

TCAN4x5x\_Reg.h, 152

REG\_BITS\_MCAN\_IE\_RF0WE

TCAN4x5x\_Reg.h, 152

REG\_BITS\_MCAN\_IE\_RF0WL

TCAN4x5x\_Reg.h, 152

REG\_BITS\_MCAN\_IE\_RF1FE

TCAN4x5x\_Reg.h, 152

REG\_BITS\_MCAN\_IE\_RF1FL

TCAN4x5x\_Reg.h, 152

REG\_BITS\_MCAN\_IE\_RF1LE

TCAN4x5x\_Reg.h, 152

REG\_BITS\_MCAN\_IE\_RF1LL

TCAN4x5x\_Reg.h, 152

REG\_BITS\_MCAN\_IE\_RF1NE

TCAN4x5x\_Reg.h, 152

REG\_BITS\_MCAN\_IE\_RF1NL

TCAN4x5x\_Reg.h, 152

REG\_BITS\_MCAN\_IE\_RF1WE

TCAN4x5x\_Reg.h, 152

REG\_BITS\_MCAN\_IE\_RF1WL

TCAN4x5x\_Reg.h, 152

REG\_BITS\_MCAN\_IE\_TCE

TCAN4x5x\_Reg.h, 152

REG\_BITS\_MCAN\_IE\_TCFE

TCAN4x5x\_Reg.h, 152

REG\_BITS\_MCAN\_IE\_TCFL

TCAN4x5x\_Reg.h, 152

REG\_BITS\_MCAN\_IE\_TCL

TCAN4x5x\_Reg.h, 152

REG\_BITS\_MCAN\_IE\_TEFFE

TCAN4x5x\_Reg.h, 152

REG\_BITS\_MCAN\_IE\_TEFFL

TCAN4x5x\_Reg.h, 152

REG\_BITS\_MCAN\_IE\_TEFLE

TCAN4x5x\_Reg.h, 152

REG\_BITS\_MCAN\_IE\_TEFLL

TCAN4x5x\_Reg.h, 152

REG\_BITS\_MCAN\_IE\_TEFNE

TCAN4x5x\_Reg.h, 152

REG\_BITS\_MCAN\_IE\_TEFNL

TCAN4x5x\_Reg.h, 152

REG\_BITS\_MCAN\_IE\_TEFWE

TCAN4x5x\_Reg.h, 152

REG\_BITS\_MCAN\_IE\_TEFWL

TCAN4x5x\_Reg.h, 152

REG\_BITS\_MCAN\_IE\_TFEE

TCAN4x5x\_Reg.h, 152

REG\_BITS\_MCAN\_IE\_TFEL

TCAN4x5x\_Reg.h, 152

REG\_BITS\_MCAN\_IE\_TOOE

TCAN4x5x\_Reg.h, 153

REG\_BITS\_MCAN\_IE\_TOOL

TCAN4x5x\_Reg.h, 153

REG\_BITS\_MCAN\_IE\_TSWE

TCAN4x5x\_Reg.h, 153

REG\_BITS\_MCAN\_IE\_TSWL

TCAN4x5x\_Reg.h, 153

REG\_BITS\_MCAN\_IE\_WDIE

TCAN4x5x\_Reg.h, 153

REG\_BITS\_MCAN\_IE\_WDIL

TCAN4x5x\_Reg.h, 153

REG\_BITS\_MCAN\_ILE\_EINT0

TCAN4x5x\_Reg.h, 153

REG\_BITS\_MCAN\_ILE\_EINT1

TCAN4x5x\_Reg.h, 153

REG\_BITS\_MCAN\_IR\_ARA

TCAN4x5x\_Reg.h, 153

REG\_BITS\_MCAN\_IR\_BEC

TCAN4x5x\_Reg.h, 153

REG\_BITS\_MCAN\_IR\_BEU

TCAN4x5x\_Reg.h, 153

REG\_BITS\_MCAN\_IR\_BO

TCAN4x5x\_Reg.h, 153

REG\_BITS\_MCAN\_IR\_DRX

TCAN4x5x\_Reg.h, 153

REG\_BITS\_MCAN\_IR\_ELO

TCAN4x5x\_Reg.h, 153

REG\_BITS\_MCAN\_IR\_EP

TCAN4x5x\_Reg.h, 153

REG\_BITS\_MCAN\_IR\_EW

TCAN4x5x\_Reg.h, 153

REG\_BITS\_MCAN\_IR\_HPM

TCAN4x5x\_Reg.h, 153

REG\_BITS\_MCAN\_IR\_MRAF

TCAN4x5x\_Reg.h, 153

REG\_BITS\_MCAN\_IR\_PEA

TCAN4x5x\_Reg.h, 153

REG\_BITS\_MCAN\_IR\_PED

TCAN4x5x\_Reg.h, 153

REG\_BITS\_MCAN\_IR\_RF0F

TCAN4x5x\_Reg.h, 153

REG\_BITS\_MCAN\_IR\_RF0L

TCAN4x5x\_Reg.h, 153

REG\_BITS\_MCAN\_IR\_RF0N

TCAN4x5x\_Reg.h, 153

REG\_BITS\_MCAN\_IR\_RF0W

TCAN4x5x\_Reg.h, 153

REG\_BITS\_MCAN\_IR\_RF1F

TCAN4x5x\_Reg.h, 153

REG\_BITS\_MCAN\_IR\_RF1L

TCAN4x5x\_Reg.h, 153

REG\_BITS\_MCAN\_IR\_RF1N

TCAN4x5x\_Reg.h, 154

REG\_BITS\_MCAN\_IR\_RF1W

TCAN4x5x\_Reg.h, 154

REG\_BITS\_MCAN\_IR\_TC

TCAN4x5x\_Reg.h, 154

REG\_BITS\_MCAN\_IR\_TCF

TCAN4x5x\_Reg.h, 154

REG\_BITS\_MCAN\_IR\_TEFF

TCAN4x5x\_Reg.h, 154

REG\_BITS\_MCAN\_IR\_TEFL

TCAN4x5x\_Reg.h, 154

REG\_BITS\_MCAN\_IR\_TEFN

TCAN4x5x\_Reg.h, 154

REG\_BITS\_MCAN\_IR\_TEFW

TCAN4x5x\_Reg.h, 154

REG\_BITS\_MCAN\_IR\_TFE

TCAN4x5x\_Reg.h, 154

REG\_BITS\_MCAN\_IR\_TOO

TCAN4x5x\_Reg.h, 154

REG\_BITS\_MCAN\_IR\_TSW

TCAN4x5x\_Reg.h, 154

REG\_BITS\_MCAN\_IR\_WDI

TCAN4x5x\_Reg.h, 154

REG\_BITS\_MCAN\_RXESC\_F0DS\_12B

TCAN4x5x\_Reg.h, 154

REG\_BITS\_MCAN\_RXESC\_F0DS\_16B

TCAN4x5x\_Reg.h, 154

REG\_BITS\_MCAN\_RXESC\_F0DS\_20B

TCAN4x5x\_Reg.h, 154

REG\_BITS\_MCAN\_RXESC\_F0DS\_24B

TCAN4x5x\_Reg.h, 154

REG\_BITS\_MCAN\_RXESC\_F0DS\_32B

TCAN4x5x\_Reg.h, 154

REG\_BITS\_MCAN\_RXESC\_F0DS\_48B

TCAN4x5x\_Reg.h, 154

REG\_BITS\_MCAN\_RXESC\_F0DS\_64B

TCAN4x5x\_Reg.h, 154

REG\_BITS\_MCAN\_RXESC\_F0DS\_8B

TCAN4x5x\_Reg.h, 154

REG\_BITS\_MCAN\_RXESC\_F1DS\_12B

TCAN4x5x\_Reg.h, 154

REG\_BITS\_MCAN\_RXESC\_F1DS\_16B

TCAN4x5x\_Reg.h, 154

REG\_BITS\_MCAN\_RXESC\_F1DS\_20B

TCAN4x5x\_Reg.h, 154

REG\_BITS\_MCAN\_RXESC\_F1DS\_24B

TCAN4x5x\_Reg.h, 154

REG\_BITS\_MCAN\_RXESC\_F1DS\_32B

TCAN4x5x\_Reg.h, 154

REG\_BITS\_MCAN\_RXESC\_F1DS\_48B

TCAN4x5x\_Reg.h, 154

REG\_BITS\_MCAN\_RXESC\_F1DS\_64B

TCAN4x5x\_Reg.h, 155

REG\_BITS\_MCAN\_RXESC\_F1DS\_8B

TCAN4x5x\_Reg.h, 155

REG\_BITS\_MCAN\_RXESC\_RBDS\_12B

TCAN4x5x\_Reg.h, 155

REG\_BITS\_MCAN\_RXESC\_RBDS\_16B

TCAN4x5x\_Reg.h, 155

REG\_BITS\_MCAN\_RXESC\_RBDS\_20B

TCAN4x5x\_Reg.h, 155

REG\_BITS\_MCAN\_RXESC\_RBDS\_24B

TCAN4x5x\_Reg.h, 155

REG\_BITS\_MCAN\_RXESC\_RBDS\_32B

TCAN4x5x\_Reg.h, 155

REG\_BITS\_MCAN\_RXESC\_RBDS\_48B

TCAN4x5x\_Reg.h, 155

REG\_BITS\_MCAN\_RXESC\_RBDS\_64B

TCAN4x5x\_Reg.h, 155

REG\_BITS\_MCAN\_RXESC\_RBDS\_8B

TCAN4x5x\_Reg.h, 155

REG\_BITS\_MCAN\_RXF0C\_F0OM\_OVERWRITE

TCAN4x5x\_Reg.h, 155

REG\_BITS\_MCAN\_TEST\_LOOP\_BACK

TCAN4x5x\_Reg.h, 155

REG\_BITS\_MCAN\_TEST\_RX\_DOM

TCAN4x5x\_Reg.h, 155

REG\_BITS\_MCAN\_TEST\_RX\_REC

TCAN4x5x\_Reg.h, 155

REG\_BITS\_MCAN\_TEST\_TX\_DOM

TCAN4x5x\_Reg.h, 155

REG\_BITS\_MCAN\_TEST\_TX\_REC

TCAN4x5x\_Reg.h, 155

REG\_BITS\_MCAN\_TEST\_TX\_SP

TCAN4x5x\_Reg.h, 155

REG\_BITS\_MCAN\_TSCC\_COUNTER\_ALWAYS\_0

TCAN4x5x\_Reg.h, 155

REG\_BITS\_MCAN\_TSCC\_COUNTER\_EXTERNAL

TCAN4x5x\_Reg.h, 155

REG\_BITS\_MCAN\_TSCC\_COUNTER\_USE\_TCP

TCAN4x5x\_Reg.h, 155

REG\_BITS\_MCAN\_TSCC\_PRESCALER\_MASK

TCAN4x5x\_Reg.h, 155

REG\_BITS\_MCAN\_TXBAR\_AR0

TCAN4x5x\_Reg.h, 155

REG\_BITS\_MCAN\_TXBAR\_AR1

TCAN4x5x\_Reg.h, 155

REG\_BITS\_MCAN\_TXBAR\_AR10

TCAN4x5x\_Reg.h, 155

REG\_BITS\_MCAN\_TXBAR\_AR11

TCAN4x5x\_Reg.h, 155

REG\_BITS\_MCAN\_TXBAR\_AR12

TCAN4x5x\_Reg.h, 155

REG\_BITS\_MCAN\_TXBAR\_AR13

TCAN4x5x\_Reg.h, 156

REG\_BITS\_MCAN\_TXBAR\_AR14

TCAN4x5x\_Reg.h, 156

REG\_BITS\_MCAN\_TXBAR\_AR15

TCAN4x5x\_Reg.h, 156

REG\_BITS\_MCAN\_TXBAR\_AR16

TCAN4x5x\_Reg.h, 156

REG\_BITS\_MCAN\_TXBAR\_AR17

TCAN4x5x\_Reg.h, 156

REG\_BITS\_MCAN\_TXBAR\_AR18

TCAN4x5x\_Reg.h, 156

REG\_BITS\_MCAN\_TXBAR\_AR19

TCAN4x5x\_Reg.h, 156

REG\_BITS\_MCAN\_TXBAR\_AR2

TCAN4x5x\_Reg.h, 156

REG\_BITS\_MCAN\_TXBAR\_AR20

TCAN4x5x\_Reg.h, 156

REG\_BITS\_MCAN\_TXBAR\_AR21

TCAN4x5x\_Reg.h, 156

REG\_BITS\_MCAN\_TXBAR\_AR22

TCAN4x5x\_Reg.h, 156

REG\_BITS\_MCAN\_TXBAR\_AR23

TCAN4x5x\_Reg.h, 156

REG\_BITS\_MCAN\_TXBAR\_AR24

TCAN4x5x\_Reg.h, 156

REG\_BITS\_MCAN\_TXBAR\_AR25

TCAN4x5x\_Reg.h, 156

REG\_BITS\_MCAN\_TXBAR\_AR26

TCAN4x5x\_Reg.h, 156

REG\_BITS\_MCAN\_TXBAR\_AR27

TCAN4x5x\_Reg.h, 156

REG\_BITS\_MCAN\_TXBAR\_AR28

TCAN4x5x\_Reg.h, 156

REG\_BITS\_MCAN\_TXBAR\_AR29

TCAN4x5x\_Reg.h, 156

REG\_BITS\_MCAN\_TXBAR\_AR3

TCAN4x5x\_Reg.h, 156

REG\_BITS\_MCAN\_TXBAR\_AR30

TCAN4x5x\_Reg.h, 156

REG\_BITS\_MCAN\_TXBAR\_AR31

TCAN4x5x\_Reg.h, 156

REG\_BITS\_MCAN\_TXBAR\_AR4

TCAN4x5x\_Reg.h, 156

REG\_BITS\_MCAN\_TXBAR\_AR5

TCAN4x5x\_Reg.h, 156

REG\_BITS\_MCAN\_TXBAR\_AR6

TCAN4x5x\_Reg.h, 156

REG\_BITS\_MCAN\_TXBAR\_AR7

TCAN4x5x\_Reg.h, 156

REG\_BITS\_MCAN\_TXBAR\_AR8

TCAN4x5x\_Reg.h, 156

REG\_BITS\_MCAN\_TXBAR\_AR9

TCAN4x5x\_Reg.h, 157

REG\_BITS\_MCAN\_TXBC\_TFQM

TCAN4x5x\_Reg.h, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE0

TCAN4x5x\_Reg.h, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE1

TCAN4x5x\_Reg.h, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE10

TCAN4x5x\_Reg.h, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE11

TCAN4x5x\_Reg.h, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE12

TCAN4x5x\_Reg.h, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE13

TCAN4x5x\_Reg.h, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE14

TCAN4x5x\_Reg.h, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE15

TCAN4x5x\_Reg.h, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE16

TCAN4x5x\_Reg.h, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE17

TCAN4x5x\_Reg.h, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE18

TCAN4x5x\_Reg.h, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE19

TCAN4x5x\_Reg.h, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE2

TCAN4x5x\_Reg.h, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE20

TCAN4x5x\_Reg.h, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE21

TCAN4x5x\_Reg.h, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE22

TCAN4x5x\_Reg.h, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE23

TCAN4x5x\_Reg.h, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE24

TCAN4x5x\_Reg.h, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE25

TCAN4x5x\_Reg.h, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE26

TCAN4x5x\_Reg.h, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE27

TCAN4x5x\_Reg.h, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE28

TCAN4x5x\_Reg.h, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE29

TCAN4x5x\_Reg.h, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE3

TCAN4x5x\_Reg.h, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE30

TCAN4x5x\_Reg.h, 158

REG\_BITS\_MCAN\_TXBCIE\_CFIE31

TCAN4x5x\_Reg.h, 158

REG\_BITS\_MCAN\_TXBCIE\_CFIE4

TCAN4x5x\_Reg.h, 158

REG\_BITS\_MCAN\_TXBCIE\_CFIE5

TCAN4x5x\_Reg.h, 158

REG\_BITS\_MCAN\_TXBCIE\_CFIE6

TCAN4x5x\_Reg.h, 158

REG\_BITS\_MCAN\_TXBCIE\_CFIE7

TCAN4x5x\_Reg.h, 158

REG\_BITS\_MCAN\_TXBCIE\_CFIE8

TCAN4x5x\_Reg.h, 158

REG\_BITS\_MCAN\_TXBCIE\_CFIE9

TCAN4x5x\_Reg.h, 158

REG\_BITS\_MCAN\_TXBCR\_CR0

TCAN4x5x\_Reg.h, 158

REG\_BITS\_MCAN\_TXBCR\_CR1

TCAN4x5x\_Reg.h, 158

REG\_BITS\_MCAN\_TXBCR\_CR10

TCAN4x5x\_Reg.h, 158

REG\_BITS\_MCAN\_TXBCR\_CR11

TCAN4x5x\_Reg.h, 158

REG\_BITS\_MCAN\_TXBCR\_CR12

TCAN4x5x\_Reg.h, 158

REG\_BITS\_MCAN\_TXBCR\_CR13

TCAN4x5x\_Reg.h, 158

REG\_BITS\_MCAN\_TXBCR\_CR14

TCAN4x5x\_Reg.h, 158

REG\_BITS\_MCAN\_TXBCR\_CR15

TCAN4x5x\_Reg.h, 158

REG\_BITS\_MCAN\_TXBCR\_CR16

TCAN4x5x\_Reg.h, 158

REG\_BITS\_MCAN\_TXBCR\_CR17

TCAN4x5x\_Reg.h, 158

REG\_BITS\_MCAN\_TXBCR\_CR18

TCAN4x5x\_Reg.h, 158

REG\_BITS\_MCAN\_TXBCR\_CR19

TCAN4x5x\_Reg.h, 158

REG\_BITS\_MCAN\_TXBCR\_CR2

TCAN4x5x\_Reg.h, 158

REG\_BITS\_MCAN\_TXBCR\_CR20

TCAN4x5x\_Reg.h, 158

REG\_BITS\_MCAN\_TXBCR\_CR21

TCAN4x5x\_Reg.h, 158

REG\_BITS\_MCAN\_TXBCR\_CR22

TCAN4x5x\_Reg.h, 158

REG\_BITS\_MCAN\_TXBCR\_CR23

TCAN4x5x\_Reg.h, 158

REG\_BITS\_MCAN\_TXBCR\_CR24

TCAN4x5x\_Reg.h, 158

REG\_BITS\_MCAN\_TXBCR\_CR25

TCAN4x5x\_Reg.h, 159

REG\_BITS\_MCAN\_TXBCR\_CR26

TCAN4x5x\_Reg.h, 159

REG\_BITS\_MCAN\_TXBCR\_CR27

TCAN4x5x\_Reg.h, 159

REG\_BITS\_MCAN\_TXBCR\_CR28

TCAN4x5x\_Reg.h, 159

REG\_BITS\_MCAN\_TXBCR\_CR29

TCAN4x5x\_Reg.h, 159

REG\_BITS\_MCAN\_TXBCR\_CR3

TCAN4x5x\_Reg.h, 159

REG\_BITS\_MCAN\_TXBCR\_CR30

TCAN4x5x\_Reg.h, 159

REG\_BITS\_MCAN\_TXBCR\_CR31

TCAN4x5x\_Reg.h, 159

REG\_BITS\_MCAN\_TXBCR\_CR4

TCAN4x5x\_Reg.h, 159

REG\_BITS\_MCAN\_TXBCR\_CR5

TCAN4x5x\_Reg.h, 159

REG\_BITS\_MCAN\_TXBCR\_CR6

TCAN4x5x\_Reg.h, 159

REG\_BITS\_MCAN\_TXBCR\_CR7

TCAN4x5x\_Reg.h, 159

REG\_BITS\_MCAN\_TXBCR\_CR8

TCAN4x5x\_Reg.h, 159

REG\_BITS\_MCAN\_TXBCR\_CR9

TCAN4x5x\_Reg.h, 159

REG\_BITS\_MCAN\_TXBTIE\_TIE0

TCAN4x5x\_Reg.h, 159

REG\_BITS\_MCAN\_TXBTIE\_TIE1

TCAN4x5x\_Reg.h, 159

REG\_BITS\_MCAN\_TXBTIE\_TIE10

TCAN4x5x\_Reg.h, 159

REG\_BITS\_MCAN\_TXBTIE\_TIE11

TCAN4x5x\_Reg.h, 159

REG\_BITS\_MCAN\_TXBTIE\_TIE12

TCAN4x5x\_Reg.h, 159

REG\_BITS\_MCAN\_TXBTIE\_TIE13

TCAN4x5x\_Reg.h, 159

REG\_BITS\_MCAN\_TXBTIE\_TIE14

TCAN4x5x\_Reg.h, 159

REG\_BITS\_MCAN\_TXBTIE\_TIE15

TCAN4x5x\_Reg.h, 159

REG\_BITS\_MCAN\_TXBTIE\_TIE16

TCAN4x5x\_Reg.h, 159

REG\_BITS\_MCAN\_TXBTIE\_TIE17

TCAN4x5x\_Reg.h, 159

REG\_BITS\_MCAN\_TXBTIE\_TIE18

TCAN4x5x\_Reg.h, 159

REG\_BITS\_MCAN\_TXBTIE\_TIE19

TCAN4x5x\_Reg.h, 159

REG\_BITS\_MCAN\_TXBTIE\_TIE2

TCAN4x5x\_Reg.h, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE20

TCAN4x5x\_Reg.h, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE21

TCAN4x5x\_Reg.h, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE22

TCAN4x5x\_Reg.h, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE23

TCAN4x5x\_Reg.h, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE24

TCAN4x5x\_Reg.h, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE25

TCAN4x5x\_Reg.h, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE26

TCAN4x5x\_Reg.h, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE27

TCAN4x5x\_Reg.h, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE28

TCAN4x5x\_Reg.h, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE29

TCAN4x5x\_Reg.h, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE3

TCAN4x5x\_Reg.h, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE30

TCAN4x5x\_Reg.h, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE31

TCAN4x5x\_Reg.h, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE4

TCAN4x5x\_Reg.h, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE5

TCAN4x5x\_Reg.h, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE6

TCAN4x5x\_Reg.h, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE7

TCAN4x5x\_Reg.h, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE8

TCAN4x5x\_Reg.h, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE9

TCAN4x5x\_Reg.h, 160

REG\_BITS\_MCAN\_TXESC\_TBDS\_12

TCAN4x5x\_Reg.h, 160

REG\_BITS\_MCAN\_TXESC\_TBDS\_16

TCAN4x5x\_Reg.h, 160

REG\_BITS\_MCAN\_TXESC\_TBDS\_20

TCAN4x5x\_Reg.h, 160

REG\_BITS\_MCAN\_TXESC\_TBDS\_24

TCAN4x5x\_Reg.h, 160

REG\_BITS\_MCAN\_TXESC\_TBDS\_32

TCAN4x5x\_Reg.h, 160

REG\_BITS\_MCAN\_TXESC\_TBDS\_48

TCAN4x5x\_Reg.h, 160

REG\_BITS\_MCAN\_TXESC\_TBDS\_64

TCAN4x5x\_Reg.h, 161

REG\_BITS\_MCAN\_TXESC\_TBDS\_8

TCAN4x5x\_Reg.h, 161

REG\_DEV\_CONFIG

TCAN4x5x\_Reg.h, 161

REG\_DEV\_IE

TCAN4x5x\_Reg.h, 161

REG\_DEV\_IR

TCAN4x5x\_Reg.h, 161

REG\_DEV\_MODES\_AND\_PINS

TCAN4x5x\_Reg.h, 161

REG\_DEV\_TEST\_REGISTERS

TCAN4x5x\_Reg.h, 161

REG\_DEV\_TIMESTAMP\_PRESCALER

TCAN4x5x\_Reg.h, 161

REG\_MCAN

TCAN4x5x\_Reg.h, 161

REG\_MCAN\_CCCR

TCAN4x5x\_Reg.h, 161

REG\_MCAN\_CREL

TCAN4x5x\_Reg.h, 161

REG\_MCAN\_CUST

TCAN4x5x\_Reg.h, 161

REG\_MCAN\_DBTP

TCAN4x5x\_Reg.h, 161

REG\_MCAN\_ECR

TCAN4x5x\_Reg.h, 161

REG\_MCAN\_ENDN

TCAN4x5x\_Reg.h, 161

REG\_MCAN\_GFC

TCAN4x5x\_Reg.h, 161

REG\_MCAN\_HPMS

TCAN4x5x\_Reg.h, 161

REG\_MCAN\_IE

TCAN4x5x\_Reg.h, 161

REG\_MCAN\_ILE

TCAN4x5x\_Reg.h, 161

REG\_MCAN\_ILS

TCAN4x5x\_Reg.h, 161

REG\_MCAN\_IR

TCAN4x5x\_Reg.h, 161

REG\_MCAN\_NBTP

TCAN4x5x\_Reg.h, 161

REG\_MCAN\_NDAT1

TCAN4x5x\_Reg.h, 161

REG\_MCAN\_NDAT2

TCAN4x5x\_Reg.h, 161

REG\_MCAN\_PSR

TCAN4x5x\_Reg.h, 161

REG\_MCAN\_RWD

TCAN4x5x\_Reg.h, 161

REG\_MCAN\_RXBC

TCAN4x5x\_Reg.h, 162

REG\_MCAN\_RXESC

TCAN4x5x\_Reg.h, 162

REG\_MCAN\_RXF0A

TCAN4x5x\_Reg.h, 162

REG\_MCAN\_RXF0C

TCAN4x5x\_Reg.h, 162

REG\_MCAN\_RXF0S

TCAN4x5x\_Reg.h, 162

REG\_MCAN\_RXF1A

TCAN4x5x\_Reg.h, 162

REG\_MCAN\_RXF1C

TCAN4x5x\_Reg.h, 162

REG\_MCAN\_RXF1S

TCAN4x5x\_Reg.h, 162

REG\_MCAN\_SIDFC

TCAN4x5x\_Reg.h, 162

REG\_MCAN\_TDCR

TCAN4x5x\_Reg.h, 162

REG\_MCAN\_TEST

TCAN4x5x\_Reg.h, 162

REG\_MCAN\_TOCC

TCAN4x5x\_Reg.h, 162

REG\_MCAN\_TOCV

TCAN4x5x\_Reg.h, 162

REG\_MCAN\_TSCC

TCAN4x5x\_Reg.h, 162

REG\_MCAN\_TSCV

TCAN4x5x\_Reg.h, 162

REG\_MCAN\_TXBAR

TCAN4x5x\_Reg.h, 162

REG\_MCAN\_TXBC

TCAN4x5x\_Reg.h, 162

REG\_MCAN\_TXBCF

TCAN4x5x\_Reg.h, 162

REG\_MCAN\_TXBCIE

TCAN4x5x\_Reg.h, 162

REG\_MCAN\_TXBCR

TCAN4x5x\_Reg.h, 162

REG\_MCAN\_TXBRP

TCAN4x5x\_Reg.h, 162

REG\_MCAN\_TXBTIE

TCAN4x5x\_Reg.h, 162

REG\_MCAN\_TXBTO

TCAN4x5x\_Reg.h, 162

REG\_MCAN\_TXEFA

TCAN4x5x\_Reg.h, 162

REG\_MCAN\_TXEFC

TCAN4x5x\_Reg.h, 162

REG\_MCAN\_TXEFS

TCAN4x5x\_Reg.h, 162

REG\_MCAN\_TXESC

TCAN4x5x\_Reg.h, 163

REG\_MCAN\_TXFQS

TCAN4x5x\_Reg.h, 163

REG\_MCAN\_XIDAM

TCAN4x5x\_Reg.h, 163

REG\_MCAN\_XIDFC

TCAN4x5x\_Reg.h, 163

REG\_MRAM

TCAN4x5x\_Reg.h, 163

REG\_SPI\_CONFIG

TCAN4x5x\_Reg.h, 163

REG\_SPI\_DEVICE\_ID0

TCAN4x5x\_Reg.h, 163

REG\_SPI\_DEVICE\_ID1

TCAN4x5x\_Reg.h, 163

REG\_SPI\_ERROR\_STATUS\_MASK

TCAN4x5x\_Reg.h, 163

REG\_SPI\_REVISION

TCAN4x5x\_Reg.h, 163

REG\_SPI\_STATUS

TCAN4x5x\_Reg.h, 163

reserved

TCAN4x5x\_MCAN\_CCCR\_Config, 69

TCAN4x5x\_MCAN\_Interrupt\_Enable, 78

TCAN4x5x\_MCAN\_Interrupts, 84

TCAN4x5x\_MCAN\_RX\_Header, 92

TCAN4x5x\_MCAN\_SID\_Filter, 94

TCAN4x5x\_MCAN\_TX\_Header, 97

TCAN4x5x\_MCAN\_XID\_Filter, 99

RESERVED

TCAN4x5x\_Device\_Interrupts, 65

RESERVED1

TCAN4x5x\_Device\_Interrupt\_Enable, 59

reserved2

TCAN4x5x\_MCAN\_CCCR\_Config, 69

RESERVED2

TCAN4x5x\_Device\_Interrupt\_Enable, 59

TCAN4x5x\_Device\_Interrupts, 65

reserved3

TCAN4x5x\_MCAN\_CCCR\_Config, 69

RESERVED3

TCAN4x5x\_Device\_Interrupt\_Enable, 59

TCAN4x5x\_Device\_Interrupts, 65

RESERVED4

TCAN4x5x\_Device\_Interrupt\_Enable, 59

reset

AlphiBoard, 14

AlteraDma, 17

ParallelInput, 31

PcieCra, 34

PCIeMini\_CAN\_FD, 39

TCAN4550, 54

TcanInterface, 104

resetIrq

ParallelInput, 31

resetStatus

AlteraSpi, 19

RF0F

TCAN4x5x\_MCAN\_Interrupts, 84

RF0FE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 78

RF0L

TCAN4x5x\_MCAN\_Interrupts, 84

RF0LE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 78

RF0N

TCAN4x5x\_MCAN\_Interrupts, 84

RF0NE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 78

RF0W

TCAN4x5x\_MCAN\_Interrupts, 84

RF0WE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 78

RF1F

TCAN4x5x\_MCAN\_Interrupts, 84

RF1FE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 78

RF1L

TCAN4x5x\_MCAN\_Interrupts, 84

RF1LE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 78

RF1N

TCAN4x5x\_MCAN\_Interrupts, 85

RF1NE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 79

RF1W

TCAN4x5x\_MCAN\_Interrupts, 85

RF1WE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 79

rstPio

TcanInterface, 105

RTR

TCAN4x5x\_MCAN\_RX\_Header, 92

TCAN4x5x\_MCAN\_TX\_Header, 97

rx\_ioctl

AlteraDma, 17

Rx0ElementSize

TCAN4x5x\_MRAM\_Config, 101

Rx0NumElements

TCAN4x5x\_MRAM\_Config, 101

Rx1ElementSize

TCAN4x5x\_MRAM\_Config, 101

Rx1NumElements

TCAN4x5x\_MRAM\_Config, 101

RxBufElementSize

TCAN4x5x\_MRAM\_Config, 101

RxBufNumElements

TCAN4x5x\_MRAM\_Config, 101

rxData\_index

AlteraSpi, 20

RXFIFO0

TCAN4550.h, 129

RXFIFO1

TCAN4550.h, 129

RXTS

TCAN4x5x\_MCAN\_RX\_Header, 92

selectSlave

AlteraSpi, 19

send

AlteraDma, 17

sendSpiCommand

AlteraSpi, 19

setControl

AlteraSpi, 20

setIrqDisable

ParallelInput, 31

setIrqEnable

ParallelInput, 31

setIrqEnableMask

PcieCra, 34

setTrEntry

PcieCra, 34

setTxData

AlteraSpi, 20

setTxsAvlAddress

PcieCra, 35

setVerbose

AlphiBoard, 14

SFEC

TCAN4x5x\_MCAN\_SID\_Filter, 94

SFID1

TCAN4x5x\_MCAN\_SID\_Filter, 95

SFID2

TCAN4x5x\_MCAN\_SID\_Filter, 95

SFT

TCAN4x5x\_MCAN\_SID\_Filter, 95

SIDNumElements

TCAN4x5x\_MRAM\_Config, 101

slave

TcanInterface, 105

slaveNbr

TCAN4550, 55

slaveSelect\_index

AlteraSpi, 20

SMS

TCAN4x5x\_Device\_Interrupts, 65

SMSEN

TCAN4x5x\_Device\_Interrupt\_Enable, 59

space

AlteraDma, 17

SPIERR

TCAN4x5x\_Device\_Interrupts, 65

src\_offset

TransferDesc, 107

status

TCAN4550, 55

status\_Index

AlteraSpi, 20

status\_ROE\_mask

AlteraSpi, 20

status\_RRDY\_mask

AlteraSpi, 20

status\_TMT\_mask

AlteraSpi, 20

status\_TOE\_mask

AlteraSpi, 20

status\_TRDY\_mask

AlteraSpi, 20

statusToString

AlteraDma, 17

SWERR

TCAN4x5x\_Device\_Interrupts, 65

sysid

AlphiBoard, 15

TC

TCAN4x5x\_MCAN\_Interrupts, 85

TCAN4550, 41

BUFF\_LEN, 55

can, 55

Device\_ClearInterrupts, 44

Device\_ClearInterruptsAll, 44

Device\_ConfigureInterruptEnable, 44

Device\_DisableTestMode, 45

Device\_EnableTestMode, 45

Device\_ReadDeviceIdent, 45

Device\_ReadDeviceVersion, 45

Device\_ReadInterruptEnable, 45

Device\_ReadInterrupts, 46

Device\_ReadMode, 46

Device\_ReadTestMode, 46

Device\_SetMode, 46

disableIrq, 46

enableIrq, 46

MCAN\_ClearInterrupts, 46

MCAN\_ClearInterruptsAll, 47

MCAN\_ConfigureCCCRRegister, 47

MCAN\_ConfigureDataTiming\_Raw, 47

MCAN\_ConfigureDataTiming\_Simple, 48

MCAN\_ConfigureInterruptEnable, 48

MCAN\_ConfigureNominalTiming\_Raw, 48

MCAN\_ConfigureNominalTiming\_Simple, 49

MCAN\_DisableProtectedRegisters, 49

MCAN\_DLCtoBytes, 49

MCAN\_EnableProtectedRegisters, 49

MCAN\_ReadCCCRRegister, 50

MCAN\_ReadDataTimingFD\_Raw, 50

MCAN\_ReadDataTimingFD\_Simple, 50

MCAN\_ReadInterruptEnable, 50

MCAN\_ReadInterrupts, 50

MCAN\_ReadNextFIFO, 51

MCAN\_ReadNominalTiming\_Raw, 51

MCAN\_ReadNominalTiming\_Simple, 51

MCAN\_ReadRXBuffer, 51

MCAN\_TransmitBufferContents, 52

MCAN\_TXRXESC\_DataByteValue, 52

MCAN\_WriteSIDFilter, 52

MCAN\_WriteTXBuffer, 53

MCAN\_WriteXIDFilter, 53

MRAM\_Clear, 53

MRAM\_Configure, 53

msgBufferIn, 55

msgBufferOut, 55

msgLength, 55

reset, 54

slaveNbr, 55

status, 55

TCAN4550, 44

WDT\_Configure, 54

WDT\_Disable, 54

WDT\_Enable, 54

WDT\_Read, 54

WDT\_Reset, 54

TCAN4550.cpp, 167

TCAN4550.h

RXFIFO0, 129

RXFIFO1, 129

TCAN4x5x\_Device\_Mode\_Enum, 129

TCAN4x5x\_DEVICE\_MODE\_NORMAL, 129

TCAN4x5x\_DEVICE\_MODE\_SLEEP, 129

TCAN4x5x\_DEVICE\_MODE\_STANDBY, 129

TCAN4x5x\_DEVICE\_TEST\_MODE\_CONTROLLER, 129

TCAN4x5x\_Device\_Test\_Mode\_Enum, 129

TCAN4x5x\_DEVICE\_TEST\_MODE\_NORMAL, 129

TCAN4x5x\_DEVICE\_TEST\_MODE\_PHY, 129

TCAN4x5x\_DEVICE\_VERIFY\_CONFIGURATION\_WRITES, 128

TCAN4x5x\_MCAN\_FIFO\_Enum, 129

TCAN4x5x\_MCAN\_VERIFY\_CONFIGURATION\_WRITES, 129

TCAN4x5x\_WDT\_3S, 129

TCAN4x5x\_WDT\_600MS, 129

TCAN4x5x\_WDT\_60MS, 129

TCAN4x5x\_WDT\_6S, 130

TCAN4x5x\_WDT\_Timer\_Enum, 129

TCAN4x5x\_Data\_Structs.h

MRAM\_12\_Byte\_Data, 133

MRAM\_16\_Byte\_Data, 133

MRAM\_20\_Byte\_Data, 133

MRAM\_24\_Byte\_Data, 133

MRAM\_32\_Byte\_Data, 133

MRAM\_48\_Byte\_Data, 133

MRAM\_64\_Byte\_Data, 133

MRAM\_8\_Byte\_Data, 133

TCAN4x5x\_MRAM\_Element\_Data\_Size, 133

TCAN4x5x\_SID\_SFEC\_DISABLED, 133

TCAN4x5x\_SID\_SFEC\_PRIORITY, 133

TCAN4x5x\_SID\_SFEC\_PRIORITYSTORERX0, 133

TCAN4x5x\_SID\_SFEC\_PRIORITYSTORERX1, 133

TCAN4x5x\_SID\_SFEC\_REJECTMATCH, 133

TCAN4x5x\_SID\_SFEC\_STORERX0, 133

TCAN4x5x\_SID\_SFEC\_STORERX1, 133

TCAN4x5x\_SID\_SFEC\_STORERXBUFORDEBUG, 134

TCAN4x5x\_SID\_SFEC\_Values, 133

TCAN4x5x\_SID\_SFT\_CLASSIC, 134

TCAN4x5x\_SID\_SFT\_DISABLED, 134

TCAN4x5x\_SID\_SFT\_DUALID, 134

TCAN4x5x\_SID\_SFT\_RANGE, 134

TCAN4x5x\_SID\_SFT\_Values, 134

TCAN4x5x\_XID\_EFEC\_DISABLED, 134

TCAN4x5x\_XID\_EFEC\_PRIORITY, 134

TCAN4x5x\_XID\_EFEC\_PRIORITYSTORERX0, 134

TCAN4x5x\_XID\_EFEC\_PRIORITYSTORERX1, 134

TCAN4x5x\_XID\_EFEC\_REJECTMATCH, 134

TCAN4x5x\_XID\_EFEC\_STORERX0, 134

TCAN4x5x\_XID\_EFEC\_STORERX1, 134

TCAN4x5x\_XID\_EFEC\_STORERXBUFORDEBUG, 134

TCAN4x5x\_XID\_EFEC\_Values, 134

TCAN4x5x\_XID\_EFT\_CLASSIC, 135

TCAN4x5x\_XID\_EFT\_DUALID, 135

TCAN4x5x\_XID\_EFT\_RANGE, 135

TCAN4x5x\_XID\_EFT\_RANGENOMASK, 135

TCAN4x5x\_XID\_EFT\_Values, 134

TCAN4x5x\_Device\_Interrupt\_Enable, 56

CANBUSBATEN, 58

CANBUSGNDEN, 58

CANBUSNORMEN, 58

CANBUSOPENEN, 58

CANBUSTERMOPENEN, 58

CANDOMEN, 58

CANHBATEN, 58

CANHCANLEN, 58

CANINTEN, 58

CANLGNDEN, 58

CANTOEN, 58

ECCERREN, 59

FRAME\_OVFEN, 59

LWUEN, 59

PWRONEN, 59

RESERVED1, 59

RESERVED2, 59

RESERVED3, 59

RESERVED4, 59

SMSEN, 59

TSDEN, 59

UVIOEN, 59

UVSUPEN, 59

WDTOEN, 60

WKERREN, 60

word, 60

TCAN4x5x\_Device\_Interrupts, 61

CANBUSBAT, 63

CANBUSGND, 63

CANBUSNORM, 63

CANBUSOPEN, 63

CANBUSTERMOPEN, 63

CANDOM, 63

CANERR, 64

CANHBAT, 64

CANHCANL, 64

CANINT, 64

CANLGND, 64

CANTO, 64

CBF, 64

ECCERR, 64

FRAME\_OVF, 64

GLOBALERR, 64

LWU, 64

M\_CAN\_INT, 64

PWRON, 65

RESERVED, 65

RESERVED2, 65

RESERVED3, 65

SMS, 65

SPIERR, 65

SWERR, 65

TSD, 65

UVIO, 65

UVSUP, 65

VTWD, 65

WDTO, 66

WKERR, 66

WKRQ, 66

word, 66

TCAN4x5x\_Device\_Mode\_Enum

TCAN4550.h, 129

TCAN4x5x\_DEVICE\_MODE\_NORMAL

TCAN4550.h, 129

TCAN4x5x\_DEVICE\_MODE\_SLEEP

TCAN4550.h, 129

TCAN4x5x\_DEVICE\_MODE\_STANDBY

TCAN4550.h, 129

TCAN4x5x\_DEVICE\_TEST\_MODE\_CONTROLLER

TCAN4550.h, 129

TCAN4x5x\_Device\_Test\_Mode\_Enum

TCAN4550.h, 129

TCAN4x5x\_DEVICE\_TEST\_MODE\_NORMAL

TCAN4550.h, 129

TCAN4x5x\_DEVICE\_TEST\_MODE\_PHY

TCAN4550.h, 129

TCAN4x5x\_DEVICE\_VERIFY\_CONFIGURATION\_WRITES

TCAN4550.h, 128

TCAN4x5x\_MCAN\_CCCR\_Config, 67

ASM, 68

BRSE, 68

CSR, 68

DAR, 68

EFBI, 68

FDOE, 69

MON, 69

NISO, 69

PXHD, 69

reserved, 69

reserved2, 69

reserved3, 69

TEST, 69

TXP, 69

word, 69

TCAN4x5x\_MCAN\_Data\_Timing\_Raw, 71

DataBitRatePrescaler, 71

DataSyncJumpWidth, 72

DataTimeSeg1andProp, 72

DataTimeSeg2, 72

TDCFilter, 72

TDCOffset, 72

TCAN4x5x\_MCAN\_Data\_Timing\_Simple, 73

DataBitRatePrescaler, 73

DataTqAfterSamplePoint, 73

DataTqBeforeSamplePoint, 73

TCAN4x5x\_MCAN\_FIFO\_Enum

TCAN4550.h, 129

TCAN4x5x\_MCAN\_Interrupt\_Enable, 75

ARAE, 77

BECE, 77

BEUE, 77

BOE, 77

DRXE, 77

ELOE, 77

EPE, 77

EWE, 78

HPME, 78

MRAFE, 78

PEAE, 78

PEDE, 78

reserved, 78

RF0FE, 78

RF0LE, 78

RF0NE, 78

RF0WE, 78

RF1FE, 78

RF1LE, 78

RF1NE, 79

RF1WE, 79

TCE, 79

TCFE, 79

TEFFE, 79

TEFLE, 79

TEFNE, 79

TEFWE, 79

TFEE, 79

TOOE, 79

TSWE, 79

WDIE, 79

word, 80

TCAN4x5x\_MCAN\_Interrupts, 81

ARA, 83

BEC, 83

BEU, 83

BO, 83

DRX, 83

ELO, 83

EP, 83

EW, 84

HPM, 84

MRAF, 84

PEA, 84

PED, 84

reserved, 84

RF0F, 84

RF0L, 84

RF0N, 84

RF0W, 84

RF1F, 84

RF1L, 84

RF1N, 85

RF1W, 85

TC, 85

TCF, 85

TEFF, 85

TEFL, 85

TEFN, 85

TEFW, 85

TFE, 85

TOO, 85

TSW, 85

WDI, 85

word, 86

TCAN4x5x\_MCAN\_Nominal\_Timing\_Raw, 87

NominalBitRatePrescaler, 87

NominalSyncJumpWidth, 87

NominalTimeSeg1andProp, 88

NominalTimeSeg2, 88

TCAN4x5x\_MCAN\_Nominal\_Timing\_Simple, 89

NominalBitRatePrescaler, 89

NominalTqAfterSamplePoint, 89

NominalTqBeforeSamplePoint, 89

TCAN4x5x\_MCAN\_RX\_Header, 91

ANMF, 92

BRS, 92

DLCode, 92

ESI, 92

FDF, 92

FIDX, 92

ID, 92

reserved, 92

RTR, 92

RXTS, 92

XTD, 92

TCAN4x5x\_MCAN\_SID\_Filter, 94

reserved, 94

SFEC, 94

SFID1, 95

SFID2, 95

SFT, 95

word, 95

TCAN4x5x\_MCAN\_TX\_Header, 96

BRS, 96

DLCode, 97

EFC, 97

ESI, 97

FDF, 97

ID, 97

MM, 97

reserved, 97

RTR, 97

XTD, 97

TCAN4x5x\_MCAN\_VERIFY\_CONFIGURATION\_WRITES

TCAN4550.h, 129

TCAN4x5x\_MCAN\_XID\_Filter, 98

EFEC, 98

EFID1, 98

EFID2, 98

EFT, 98

reserved, 99

TCAN4x5x\_MRAM\_Config, 100

Rx0ElementSize, 101

Rx0NumElements, 101

Rx1ElementSize, 101

Rx1NumElements, 101

RxBufElementSize, 101

RxBufNumElements, 101

SIDNumElements, 101

TxBufferElementSize, 101

TxBufferNumElements, 102

TxEventFIFONumElements, 102

XIDNumElements, 102

TCAN4x5x\_MRAM\_Element\_Data\_Size

TCAN4x5x\_Data\_Structs.h, 133

TCAN4x5x\_Reg.h

MCAN\_DLC\_0B, 145

MCAN\_DLC\_12B, 145

MCAN\_DLC\_16B, 145

MCAN\_DLC\_1B, 145

MCAN\_DLC\_20B, 145

MCAN\_DLC\_24B, 145

MCAN\_DLC\_2B, 145

MCAN\_DLC\_32B, 145

MCAN\_DLC\_3B, 145

MCAN\_DLC\_48B, 145

MCAN\_DLC\_4B, 145

MCAN\_DLC\_5B, 145

MCAN\_DLC\_64B, 145

MCAN\_DLC\_6B, 145

MCAN\_DLC\_7B, 145

MCAN\_DLC\_8B, 145

MRAM\_SIZE, 145

REG\_BITS\_DEVICE\_IE\_CANBUSBAT, 145

REG\_BITS\_DEVICE\_IE\_CANBUSGND, 145

REG\_BITS\_DEVICE\_IE\_CANBUSNOM, 145

REG\_BITS\_DEVICE\_IE\_CANBUSOPEN, 145

REG\_BITS\_DEVICE\_IE\_CANBUSTERMOPEN, 145

REG\_BITS\_DEVICE\_IE\_CANDOM, 145

REG\_BITS\_DEVICE\_IE\_CANHBAT, 145

REG\_BITS\_DEVICE\_IE\_CANHCANL, 145

REG\_BITS\_DEVICE\_IE\_CANINT, 146

REG\_BITS\_DEVICE\_IE\_CANLGND, 146

REG\_BITS\_DEVICE\_IE\_CANSLNT, 146

REG\_BITS\_DEVICE\_IE\_ECCERR, 146

REG\_BITS\_DEVICE\_IE\_FRAME\_OVF, 146

REG\_BITS\_DEVICE\_IE\_LWU, 146

REG\_BITS\_DEVICE\_IE\_MASK, 146

REG\_BITS\_DEVICE\_IE\_PWRON, 146

REG\_BITS\_DEVICE\_IE\_TSD, 146

REG\_BITS\_DEVICE\_IE\_UVCCOUT, 146

REG\_BITS\_DEVICE\_IE\_UVIO, 146

REG\_BITS\_DEVICE\_IE\_UVSUP, 146

REG\_BITS\_DEVICE\_IE\_WDTO, 146

REG\_BITS\_DEVICE\_IE\_WKERR, 146

REG\_BITS\_DEVICE\_IR\_CANBUSBAT, 146

REG\_BITS\_DEVICE\_IR\_CANBUSFAULT, 146

REG\_BITS\_DEVICE\_IR\_CANBUSGND, 146

REG\_BITS\_DEVICE\_IR\_CANBUSNOM, 146

REG\_BITS\_DEVICE\_IR\_CANBUSOPEN, 146

REG\_BITS\_DEVICE\_IR\_CANBUSTERMOPEN, 146

REG\_BITS\_DEVICE\_IR\_CANDOM, 146

REG\_BITS\_DEVICE\_IR\_CANERR, 146

REG\_BITS\_DEVICE\_IR\_CANHBAT, 146

REG\_BITS\_DEVICE\_IR\_CANHCANL, 146

REG\_BITS\_DEVICE\_IR\_CANINT, 146

REG\_BITS\_DEVICE\_IR\_CANLGND, 146

REG\_BITS\_DEVICE\_IR\_CANSLNT, 147

REG\_BITS\_DEVICE\_IR\_ECCERR, 147

REG\_BITS\_DEVICE\_IR\_FRAME\_OVF, 147

REG\_BITS\_DEVICE\_IR\_GLOBALERR, 147

REG\_BITS\_DEVICE\_IR\_LWU, 147

REG\_BITS\_DEVICE\_IR\_M\_CAN\_INT, 147

REG\_BITS\_DEVICE\_IR\_nWKRQ, 147

REG\_BITS\_DEVICE\_IR\_PWRON, 147

REG\_BITS\_DEVICE\_IR\_SPIERR, 147

REG\_BITS\_DEVICE\_IR\_SWERR, 147

REG\_BITS\_DEVICE\_IR\_TSD, 147

REG\_BITS\_DEVICE\_IR\_UVIO, 147

REG\_BITS\_DEVICE\_IR\_UVSUP, 147

REG\_BITS\_DEVICE\_IR\_VTWD, 147

REG\_BITS\_DEVICE\_IR\_WDTO, 147

REG\_BITS\_DEVICE\_IR\_WKERR, 147

REG\_BITS\_DEVICE\_MODE\_CLKOUT\_DIV1, 147

REG\_BITS\_DEVICE\_MODE\_CLKOUT\_DIV2, 147

REG\_BITS\_DEVICE\_MODE\_CLKOUT\_MASK, 147

REG\_BITS\_DEVICE\_MODE\_DEVICE\_RESET, 147

REG\_BITS\_DEVICE\_MODE\_DEVICEMODE\_MASK, 147

REG\_BITS\_DEVICE\_MODE\_DEVICEMODE\_NORMAL, 147

REG\_BITS\_DEVICE\_MODE\_DEVICEMODE\_SLEEP, 147

REG\_BITS\_DEVICE\_MODE\_DEVICEMODE\_STANDBY, 147

REG\_BITS\_DEVICE\_MODE\_FAIL\_SAFE\_DIS, 147

REG\_BITS\_DEVICE\_MODE\_FAIL\_SAFE\_EN, 147

REG\_BITS\_DEVICE\_MODE\_FAIL\_SAFE\_MASK, 148

REG\_BITS\_DEVICE\_MODE\_GPO1\_FUNC\_MASK, 148

REG\_BITS\_DEVICE\_MODE\_GPO1\_FUNC\_MCAN\_INT1, 148

REG\_BITS\_DEVICE\_MODE\_GPO1\_FUNC\_SPI\_INT, 148

REG\_BITS\_DEVICE\_MODE\_GPO1\_FUNC\_UVLO\_THERM, 148

REG\_BITS\_DEVICE\_MODE\_GPO1\_MODE\_CLKOUT, 148

REG\_BITS\_DEVICE\_MODE\_GPO1\_MODE\_GPI, 148

REG\_BITS\_DEVICE\_MODE\_GPO1\_MODE\_GPO, 148

REG\_BITS\_DEVICE\_MODE\_GPO1\_MODE\_MASK, 148

REG\_BITS\_DEVICE\_MODE\_GPO2\_CAN\_FAULT, 148

REG\_BITS\_DEVICE\_MODE\_GPO2\_MASK, 148

REG\_BITS\_DEVICE\_MODE\_GPO2\_MCAN\_INT0, 148

REG\_BITS\_DEVICE\_MODE\_GPO2\_NINT, 148

REG\_BITS\_DEVICE\_MODE\_GPO2\_WDT, 148

REG\_BITS\_DEVICE\_MODE\_INH\_DIS, 148

REG\_BITS\_DEVICE\_MODE\_INH\_EN, 148

REG\_BITS\_DEVICE\_MODE\_INH\_MASK, 148

REG\_BITS\_DEVICE\_MODE\_NWKRQ\_CONFIG\_INH, 148

REG\_BITS\_DEVICE\_MODE\_NWKRQ\_CONFIG\_MASK, 148

REG\_BITS\_DEVICE\_MODE\_NWKRQ\_CONFIG\_WKRQ, 148

REG\_BITS\_DEVICE\_MODE\_NWKRQ\_VOLT\_INTERNAL, 148

REG\_BITS\_DEVICE\_MODE\_NWKRQ\_VOLT\_MASK, 148

REG\_BITS\_DEVICE\_MODE\_NWKRQ\_VOLT\_VIO, 148

REG\_BITS\_DEVICE\_MODE\_SWE\_DIS, 148

REG\_BITS\_DEVICE\_MODE\_SWE\_EN, 148

REG\_BITS\_DEVICE\_MODE\_SWE\_MASK, 148

REG\_BITS\_DEVICE\_MODE\_TESTMODE\_CONTROLLER, 149

REG\_BITS\_DEVICE\_MODE\_TESTMODE\_DIS, 149

REG\_BITS\_DEVICE\_MODE\_TESTMODE\_EN, 149

REG\_BITS\_DEVICE\_MODE\_TESTMODE\_ENMASK, 149

REG\_BITS\_DEVICE\_MODE\_TESTMODE\_MASK, 149

REG\_BITS\_DEVICE\_MODE\_TESTMODE\_PHY, 149

REG\_BITS\_DEVICE\_MODE\_WAKE\_PIN\_BOTHEDGES, 149

REG\_BITS\_DEVICE\_MODE\_WAKE\_PIN\_DIS, 149

REG\_BITS\_DEVICE\_MODE\_WAKE\_PIN\_FALLING, 149

REG\_BITS\_DEVICE\_MODE\_WAKE\_PIN\_MASK, 149

REG\_BITS\_DEVICE\_MODE\_WAKE\_PIN\_RISING, 149

REG\_BITS\_DEVICE\_MODE\_WD\_CLK\_20MHZ, 149

REG\_BITS\_DEVICE\_MODE\_WD\_CLK\_40MHZ, 149

REG\_BITS\_DEVICE\_MODE\_WD\_CLK\_MASK, 149

REG\_BITS\_DEVICE\_MODE\_WD\_TIMER\_3S, 149

REG\_BITS\_DEVICE\_MODE\_WD\_TIMER\_600MS, 149

REG\_BITS\_DEVICE\_MODE\_WD\_TIMER\_60MS, 149

REG\_BITS\_DEVICE\_MODE\_WD\_TIMER\_6S, 149

REG\_BITS\_DEVICE\_MODE\_WD\_TIMER\_MASK, 149

REG\_BITS\_DEVICE\_MODE\_WDT\_ACTION\_INH\_PULSE, 149

REG\_BITS\_DEVICE\_MODE\_WDT\_ACTION\_INT, 149

REG\_BITS\_DEVICE\_MODE\_WDT\_ACTION\_MASK, 149

REG\_BITS\_DEVICE\_MODE\_WDT\_ACTION\_WDT\_PULSE, 149

REG\_BITS\_DEVICE\_MODE\_WDT\_DIS, 149

REG\_BITS\_DEVICE\_MODE\_WDT\_EN, 149

REG\_BITS\_DEVICE\_MODE\_WDT\_MASK, 149

REG\_BITS\_DEVICE\_MODE\_WDT\_RESET\_BIT, 150

REG\_BITS\_MCAN\_CCCR\_ASM, 150

REG\_BITS\_MCAN\_CCCR\_BRSE, 150

REG\_BITS\_MCAN\_CCCR\_CCE, 150

REG\_BITS\_MCAN\_CCCR\_CSA, 150

REG\_BITS\_MCAN\_CCCR\_CSR, 150

REG\_BITS\_MCAN\_CCCR\_DAR\_DIS, 150

REG\_BITS\_MCAN\_CCCR\_EFBI, 150

REG\_BITS\_MCAN\_CCCR\_FDOE, 150

REG\_BITS\_MCAN\_CCCR\_INIT, 150

REG\_BITS\_MCAN\_CCCR\_MON, 150

REG\_BITS\_MCAN\_CCCR\_NISO\_BOSCH, 150

REG\_BITS\_MCAN\_CCCR\_NISO\_ISO, 150

REG\_BITS\_MCAN\_CCCR\_PXHD\_DIS, 150

REG\_BITS\_MCAN\_CCCR\_RESERVED\_MASK, 150

REG\_BITS\_MCAN\_CCCR\_TEST, 150

REG\_BITS\_MCAN\_CCCR\_TXP, 150

REG\_BITS\_MCAN\_DBTP\_TDC\_EN, 150

REG\_BITS\_MCAN\_GFC\_ANFE\_FIFO0, 150

REG\_BITS\_MCAN\_GFC\_ANFE\_FIFO1, 150

REG\_BITS\_MCAN\_GFC\_ANFS\_FIFO0, 150

REG\_BITS\_MCAN\_GFC\_ANFS\_FIFO1, 150

REG\_BITS\_MCAN\_GFC\_RRFE, 150

REG\_BITS\_MCAN\_GFC\_RRFS, 150

REG\_BITS\_MCAN\_IE\_ARAE, 150

REG\_BITS\_MCAN\_IE\_ARAL, 150

REG\_BITS\_MCAN\_IE\_BECE, 151

REG\_BITS\_MCAN\_IE\_BECL, 151

REG\_BITS\_MCAN\_IE\_BEUE, 151

REG\_BITS\_MCAN\_IE\_BEUL, 151

REG\_BITS\_MCAN\_IE\_BOE, 151

REG\_BITS\_MCAN\_IE\_BOL, 151

REG\_BITS\_MCAN\_IE\_DRXE, 151

REG\_BITS\_MCAN\_IE\_DRXL, 151

REG\_BITS\_MCAN\_IE\_ELOE, 151

REG\_BITS\_MCAN\_IE\_ELOL, 151

REG\_BITS\_MCAN\_IE\_EPE, 151

REG\_BITS\_MCAN\_IE\_EPL, 151

REG\_BITS\_MCAN\_IE\_EWE, 151

REG\_BITS\_MCAN\_IE\_EWL, 151

REG\_BITS\_MCAN\_IE\_HPME, 151

REG\_BITS\_MCAN\_IE\_HPML, 151

REG\_BITS\_MCAN\_IE\_MRAFE, 151

REG\_BITS\_MCAN\_IE\_MRAFL, 151

REG\_BITS\_MCAN\_IE\_PEAE, 151

REG\_BITS\_MCAN\_IE\_PEAL, 151

REG\_BITS\_MCAN\_IE\_PEDE, 151

REG\_BITS\_MCAN\_IE\_PEDL, 151

REG\_BITS\_MCAN\_IE\_RF0FE, 151

REG\_BITS\_MCAN\_IE\_RF0FL, 151

REG\_BITS\_MCAN\_IE\_RF0LE, 151

REG\_BITS\_MCAN\_IE\_RF0LL, 151

REG\_BITS\_MCAN\_IE\_RF0NE, 152

REG\_BITS\_MCAN\_IE\_RF0NL, 152

REG\_BITS\_MCAN\_IE\_RF0WE, 152

REG\_BITS\_MCAN\_IE\_RF0WL, 152

REG\_BITS\_MCAN\_IE\_RF1FE, 152

REG\_BITS\_MCAN\_IE\_RF1FL, 152

REG\_BITS\_MCAN\_IE\_RF1LE, 152

REG\_BITS\_MCAN\_IE\_RF1LL, 152

REG\_BITS\_MCAN\_IE\_RF1NE, 152

REG\_BITS\_MCAN\_IE\_RF1NL, 152

REG\_BITS\_MCAN\_IE\_RF1WE, 152

REG\_BITS\_MCAN\_IE\_RF1WL, 152

REG\_BITS\_MCAN\_IE\_TCE, 152

REG\_BITS\_MCAN\_IE\_TCFE, 152

REG\_BITS\_MCAN\_IE\_TCFL, 152

REG\_BITS\_MCAN\_IE\_TCL, 152

REG\_BITS\_MCAN\_IE\_TEFFE, 152

REG\_BITS\_MCAN\_IE\_TEFFL, 152

REG\_BITS\_MCAN\_IE\_TEFLE, 152

REG\_BITS\_MCAN\_IE\_TEFLL, 152

REG\_BITS\_MCAN\_IE\_TEFNE, 152

REG\_BITS\_MCAN\_IE\_TEFNL, 152

REG\_BITS\_MCAN\_IE\_TEFWE, 152

REG\_BITS\_MCAN\_IE\_TEFWL, 152

REG\_BITS\_MCAN\_IE\_TFEE, 152

REG\_BITS\_MCAN\_IE\_TFEL, 152

REG\_BITS\_MCAN\_IE\_TOOE, 153

REG\_BITS\_MCAN\_IE\_TOOL, 153

REG\_BITS\_MCAN\_IE\_TSWE, 153

REG\_BITS\_MCAN\_IE\_TSWL, 153

REG\_BITS\_MCAN\_IE\_WDIE, 153

REG\_BITS\_MCAN\_IE\_WDIL, 153

REG\_BITS\_MCAN\_ILE\_EINT0, 153

REG\_BITS\_MCAN\_ILE\_EINT1, 153

REG\_BITS\_MCAN\_IR\_ARA, 153

REG\_BITS\_MCAN\_IR\_BEC, 153

REG\_BITS\_MCAN\_IR\_BEU, 153

REG\_BITS\_MCAN\_IR\_BO, 153

REG\_BITS\_MCAN\_IR\_DRX, 153

REG\_BITS\_MCAN\_IR\_ELO, 153

REG\_BITS\_MCAN\_IR\_EP, 153

REG\_BITS\_MCAN\_IR\_EW, 153

REG\_BITS\_MCAN\_IR\_HPM, 153

REG\_BITS\_MCAN\_IR\_MRAF, 153

REG\_BITS\_MCAN\_IR\_PEA, 153

REG\_BITS\_MCAN\_IR\_PED, 153

REG\_BITS\_MCAN\_IR\_RF0F, 153

REG\_BITS\_MCAN\_IR\_RF0L, 153

REG\_BITS\_MCAN\_IR\_RF0N, 153

REG\_BITS\_MCAN\_IR\_RF0W, 153

REG\_BITS\_MCAN\_IR\_RF1F, 153

REG\_BITS\_MCAN\_IR\_RF1L, 153

REG\_BITS\_MCAN\_IR\_RF1N, 154

REG\_BITS\_MCAN\_IR\_RF1W, 154

REG\_BITS\_MCAN\_IR\_TC, 154

REG\_BITS\_MCAN\_IR\_TCF, 154

REG\_BITS\_MCAN\_IR\_TEFF, 154

REG\_BITS\_MCAN\_IR\_TEFL, 154

REG\_BITS\_MCAN\_IR\_TEFN, 154

REG\_BITS\_MCAN\_IR\_TEFW, 154

REG\_BITS\_MCAN\_IR\_TFE, 154

REG\_BITS\_MCAN\_IR\_TOO, 154

REG\_BITS\_MCAN\_IR\_TSW, 154

REG\_BITS\_MCAN\_IR\_WDI, 154

REG\_BITS\_MCAN\_RXESC\_F0DS\_12B, 154

REG\_BITS\_MCAN\_RXESC\_F0DS\_16B, 154

REG\_BITS\_MCAN\_RXESC\_F0DS\_20B, 154

REG\_BITS\_MCAN\_RXESC\_F0DS\_24B, 154

REG\_BITS\_MCAN\_RXESC\_F0DS\_32B, 154

REG\_BITS\_MCAN\_RXESC\_F0DS\_48B, 154

REG\_BITS\_MCAN\_RXESC\_F0DS\_64B, 154

REG\_BITS\_MCAN\_RXESC\_F0DS\_8B, 154

REG\_BITS\_MCAN\_RXESC\_F1DS\_12B, 154

REG\_BITS\_MCAN\_RXESC\_F1DS\_16B, 154

REG\_BITS\_MCAN\_RXESC\_F1DS\_20B, 154

REG\_BITS\_MCAN\_RXESC\_F1DS\_24B, 154

REG\_BITS\_MCAN\_RXESC\_F1DS\_32B, 154

REG\_BITS\_MCAN\_RXESC\_F1DS\_48B, 154

REG\_BITS\_MCAN\_RXESC\_F1DS\_64B, 155

REG\_BITS\_MCAN\_RXESC\_F1DS\_8B, 155

REG\_BITS\_MCAN\_RXESC\_RBDS\_12B, 155

REG\_BITS\_MCAN\_RXESC\_RBDS\_16B, 155

REG\_BITS\_MCAN\_RXESC\_RBDS\_20B, 155

REG\_BITS\_MCAN\_RXESC\_RBDS\_24B, 155

REG\_BITS\_MCAN\_RXESC\_RBDS\_32B, 155

REG\_BITS\_MCAN\_RXESC\_RBDS\_48B, 155

REG\_BITS\_MCAN\_RXESC\_RBDS\_64B, 155

REG\_BITS\_MCAN\_RXESC\_RBDS\_8B, 155

REG\_BITS\_MCAN\_RXF0C\_F0OM\_OVERWRITE, 155

REG\_BITS\_MCAN\_TEST\_LOOP\_BACK, 155

REG\_BITS\_MCAN\_TEST\_RX\_DOM, 155

REG\_BITS\_MCAN\_TEST\_RX\_REC, 155

REG\_BITS\_MCAN\_TEST\_TX\_DOM, 155

REG\_BITS\_MCAN\_TEST\_TX\_REC, 155

REG\_BITS\_MCAN\_TEST\_TX\_SP, 155

REG\_BITS\_MCAN\_TSCC\_COUNTER\_ALWAYS\_0, 155

REG\_BITS\_MCAN\_TSCC\_COUNTER\_EXTERNAL, 155

REG\_BITS\_MCAN\_TSCC\_COUNTER\_USE\_TCP, 155

REG\_BITS\_MCAN\_TSCC\_PRESCALER\_MASK, 155

REG\_BITS\_MCAN\_TXBAR\_AR0, 155

REG\_BITS\_MCAN\_TXBAR\_AR1, 155

REG\_BITS\_MCAN\_TXBAR\_AR10, 155

REG\_BITS\_MCAN\_TXBAR\_AR11, 155

REG\_BITS\_MCAN\_TXBAR\_AR12, 155

REG\_BITS\_MCAN\_TXBAR\_AR13, 156

REG\_BITS\_MCAN\_TXBAR\_AR14, 156

REG\_BITS\_MCAN\_TXBAR\_AR15, 156

REG\_BITS\_MCAN\_TXBAR\_AR16, 156

REG\_BITS\_MCAN\_TXBAR\_AR17, 156

REG\_BITS\_MCAN\_TXBAR\_AR18, 156

REG\_BITS\_MCAN\_TXBAR\_AR19, 156

REG\_BITS\_MCAN\_TXBAR\_AR2, 156

REG\_BITS\_MCAN\_TXBAR\_AR20, 156

REG\_BITS\_MCAN\_TXBAR\_AR21, 156

REG\_BITS\_MCAN\_TXBAR\_AR22, 156

REG\_BITS\_MCAN\_TXBAR\_AR23, 156

REG\_BITS\_MCAN\_TXBAR\_AR24, 156

REG\_BITS\_MCAN\_TXBAR\_AR25, 156

REG\_BITS\_MCAN\_TXBAR\_AR26, 156

REG\_BITS\_MCAN\_TXBAR\_AR27, 156

REG\_BITS\_MCAN\_TXBAR\_AR28, 156

REG\_BITS\_MCAN\_TXBAR\_AR29, 156

REG\_BITS\_MCAN\_TXBAR\_AR3, 156

REG\_BITS\_MCAN\_TXBAR\_AR30, 156

REG\_BITS\_MCAN\_TXBAR\_AR31, 156

REG\_BITS\_MCAN\_TXBAR\_AR4, 156

REG\_BITS\_MCAN\_TXBAR\_AR5, 156

REG\_BITS\_MCAN\_TXBAR\_AR6, 156

REG\_BITS\_MCAN\_TXBAR\_AR7, 156

REG\_BITS\_MCAN\_TXBAR\_AR8, 156

REG\_BITS\_MCAN\_TXBAR\_AR9, 157

REG\_BITS\_MCAN\_TXBC\_TFQM, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE0, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE1, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE10, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE11, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE12, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE13, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE14, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE15, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE16, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE17, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE18, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE19, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE2, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE20, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE21, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE22, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE23, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE24, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE25, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE26, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE27, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE28, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE29, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE3, 157

REG\_BITS\_MCAN\_TXBCIE\_CFIE30, 158

REG\_BITS\_MCAN\_TXBCIE\_CFIE31, 158

REG\_BITS\_MCAN\_TXBCIE\_CFIE4, 158

REG\_BITS\_MCAN\_TXBCIE\_CFIE5, 158

REG\_BITS\_MCAN\_TXBCIE\_CFIE6, 158

REG\_BITS\_MCAN\_TXBCIE\_CFIE7, 158

REG\_BITS\_MCAN\_TXBCIE\_CFIE8, 158

REG\_BITS\_MCAN\_TXBCIE\_CFIE9, 158

REG\_BITS\_MCAN\_TXBCR\_CR0, 158

REG\_BITS\_MCAN\_TXBCR\_CR1, 158

REG\_BITS\_MCAN\_TXBCR\_CR10, 158

REG\_BITS\_MCAN\_TXBCR\_CR11, 158

REG\_BITS\_MCAN\_TXBCR\_CR12, 158

REG\_BITS\_MCAN\_TXBCR\_CR13, 158

REG\_BITS\_MCAN\_TXBCR\_CR14, 158

REG\_BITS\_MCAN\_TXBCR\_CR15, 158

REG\_BITS\_MCAN\_TXBCR\_CR16, 158

REG\_BITS\_MCAN\_TXBCR\_CR17, 158

REG\_BITS\_MCAN\_TXBCR\_CR18, 158

REG\_BITS\_MCAN\_TXBCR\_CR19, 158

REG\_BITS\_MCAN\_TXBCR\_CR2, 158

REG\_BITS\_MCAN\_TXBCR\_CR20, 158

REG\_BITS\_MCAN\_TXBCR\_CR21, 158

REG\_BITS\_MCAN\_TXBCR\_CR22, 158

REG\_BITS\_MCAN\_TXBCR\_CR23, 158

REG\_BITS\_MCAN\_TXBCR\_CR24, 158

REG\_BITS\_MCAN\_TXBCR\_CR25, 159

REG\_BITS\_MCAN\_TXBCR\_CR26, 159

REG\_BITS\_MCAN\_TXBCR\_CR27, 159

REG\_BITS\_MCAN\_TXBCR\_CR28, 159

REG\_BITS\_MCAN\_TXBCR\_CR29, 159

REG\_BITS\_MCAN\_TXBCR\_CR3, 159

REG\_BITS\_MCAN\_TXBCR\_CR30, 159

REG\_BITS\_MCAN\_TXBCR\_CR31, 159

REG\_BITS\_MCAN\_TXBCR\_CR4, 159

REG\_BITS\_MCAN\_TXBCR\_CR5, 159

REG\_BITS\_MCAN\_TXBCR\_CR6, 159

REG\_BITS\_MCAN\_TXBCR\_CR7, 159

REG\_BITS\_MCAN\_TXBCR\_CR8, 159

REG\_BITS\_MCAN\_TXBCR\_CR9, 159

REG\_BITS\_MCAN\_TXBTIE\_TIE0, 159

REG\_BITS\_MCAN\_TXBTIE\_TIE1, 159

REG\_BITS\_MCAN\_TXBTIE\_TIE10, 159

REG\_BITS\_MCAN\_TXBTIE\_TIE11, 159

REG\_BITS\_MCAN\_TXBTIE\_TIE12, 159

REG\_BITS\_MCAN\_TXBTIE\_TIE13, 159

REG\_BITS\_MCAN\_TXBTIE\_TIE14, 159

REG\_BITS\_MCAN\_TXBTIE\_TIE15, 159

REG\_BITS\_MCAN\_TXBTIE\_TIE16, 159

REG\_BITS\_MCAN\_TXBTIE\_TIE17, 159

REG\_BITS\_MCAN\_TXBTIE\_TIE18, 159

REG\_BITS\_MCAN\_TXBTIE\_TIE19, 159

REG\_BITS\_MCAN\_TXBTIE\_TIE2, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE20, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE21, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE22, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE23, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE24, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE25, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE26, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE27, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE28, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE29, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE3, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE30, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE31, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE4, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE5, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE6, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE7, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE8, 160

REG\_BITS\_MCAN\_TXBTIE\_TIE9, 160

REG\_BITS\_MCAN\_TXESC\_TBDS\_12, 160

REG\_BITS\_MCAN\_TXESC\_TBDS\_16, 160

REG\_BITS\_MCAN\_TXESC\_TBDS\_20, 160

REG\_BITS\_MCAN\_TXESC\_TBDS\_24, 160

REG\_BITS\_MCAN\_TXESC\_TBDS\_32, 160

REG\_BITS\_MCAN\_TXESC\_TBDS\_48, 160

REG\_BITS\_MCAN\_TXESC\_TBDS\_64, 161

REG\_BITS\_MCAN\_TXESC\_TBDS\_8, 161

REG\_DEV\_CONFIG, 161

REG\_DEV\_IE, 161

REG\_DEV\_IR, 161

REG\_DEV\_MODES\_AND\_PINS, 161

REG\_DEV\_TEST\_REGISTERS, 161

REG\_DEV\_TIMESTAMP\_PRESCALER, 161

REG\_MCAN, 161

REG\_MCAN\_CCCR, 161

REG\_MCAN\_CREL, 161

REG\_MCAN\_CUST, 161

REG\_MCAN\_DBTP, 161

REG\_MCAN\_ECR, 161

REG\_MCAN\_ENDN, 161

REG\_MCAN\_GFC, 161

REG\_MCAN\_HPMS, 161

REG\_MCAN\_IE, 161

REG\_MCAN\_ILE, 161

REG\_MCAN\_ILS, 161

REG\_MCAN\_IR, 161

REG\_MCAN\_NBTP, 161

REG\_MCAN\_NDAT1, 161

REG\_MCAN\_NDAT2, 161

REG\_MCAN\_PSR, 161

REG\_MCAN\_RWD, 161

REG\_MCAN\_RXBC, 162

REG\_MCAN\_RXESC, 162

REG\_MCAN\_RXF0A, 162

REG\_MCAN\_RXF0C, 162

REG\_MCAN\_RXF0S, 162

REG\_MCAN\_RXF1A, 162

REG\_MCAN\_RXF1C, 162

REG\_MCAN\_RXF1S, 162

REG\_MCAN\_SIDFC, 162

REG\_MCAN\_TDCR, 162

REG\_MCAN\_TEST, 162

REG\_MCAN\_TOCC, 162

REG\_MCAN\_TOCV, 162

REG\_MCAN\_TSCC, 162

REG\_MCAN\_TSCV, 162

REG\_MCAN\_TXBAR, 162

REG\_MCAN\_TXBC, 162

REG\_MCAN\_TXBCF, 162

REG\_MCAN\_TXBCIE, 162

REG\_MCAN\_TXBCR, 162

REG\_MCAN\_TXBRP, 162

REG\_MCAN\_TXBTIE, 162

REG\_MCAN\_TXBTO, 162

REG\_MCAN\_TXEFA, 162

REG\_MCAN\_TXEFC, 162

REG\_MCAN\_TXEFS, 162

REG\_MCAN\_TXESC, 163

REG\_MCAN\_TXFQS, 163

REG\_MCAN\_XIDAM, 163

REG\_MCAN\_XIDFC, 163

REG\_MRAM, 163

REG\_SPI\_CONFIG, 163

REG\_SPI\_DEVICE\_ID0, 163

REG\_SPI\_DEVICE\_ID1, 163

REG\_SPI\_ERROR\_STATUS\_MASK, 163

REG\_SPI\_REVISION, 163

REG\_SPI\_STATUS, 163

TCAN4x5x\_SID\_SFEC\_DISABLED

TCAN4x5x\_Data\_Structs.h, 133

TCAN4x5x\_SID\_SFEC\_PRIORITY

TCAN4x5x\_Data\_Structs.h, 133

TCAN4x5x\_SID\_SFEC\_PRIORITYSTORERX0

TCAN4x5x\_Data\_Structs.h, 133

TCAN4x5x\_SID\_SFEC\_PRIORITYSTORERX1

TCAN4x5x\_Data\_Structs.h, 133

TCAN4x5x\_SID\_SFEC\_REJECTMATCH

TCAN4x5x\_Data\_Structs.h, 133

TCAN4x5x\_SID\_SFEC\_STORERX0

TCAN4x5x\_Data\_Structs.h, 133

TCAN4x5x\_SID\_SFEC\_STORERX1

TCAN4x5x\_Data\_Structs.h, 133

TCAN4x5x\_SID\_SFEC\_STORERXBUFORDEBUG

TCAN4x5x\_Data\_Structs.h, 134

TCAN4x5x\_SID\_SFEC\_Values

TCAN4x5x\_Data\_Structs.h, 133

TCAN4x5x\_SID\_SFT\_CLASSIC

TCAN4x5x\_Data\_Structs.h, 134

TCAN4x5x\_SID\_SFT\_DISABLED

TCAN4x5x\_Data\_Structs.h, 134

TCAN4x5x\_SID\_SFT\_DUALID

TCAN4x5x\_Data\_Structs.h, 134

TCAN4x5x\_SID\_SFT\_RANGE

TCAN4x5x\_Data\_Structs.h, 134

TCAN4x5x\_SID\_SFT\_Values

TCAN4x5x\_Data\_Structs.h, 134

TCAN4x5x\_SPI.cpp, 168

USE\_AHB\_CODE, 168

WAIT\_FOR\_IDLE, 168

TCAN4x5x\_SPI.h

AHB\_READ\_OPCODE, 164

AHB\_WRITE\_OPCODE, 164

TCAN4x5x\_WDT\_3S

TCAN4550.h, 129

TCAN4x5x\_WDT\_600MS

TCAN4550.h, 129

TCAN4x5x\_WDT\_60MS

TCAN4550.h, 129

TCAN4x5x\_WDT\_6S

TCAN4550.h, 130

TCAN4x5x\_WDT\_Timer\_Enum

TCAN4550.h, 129

TCAN4x5x\_XID\_EFEC\_DISABLED

TCAN4x5x\_Data\_Structs.h, 134

TCAN4x5x\_XID\_EFEC\_PRIORITY

TCAN4x5x\_Data\_Structs.h, 134

TCAN4x5x\_XID\_EFEC\_PRIORITYSTORERX0

TCAN4x5x\_Data\_Structs.h, 134

TCAN4x5x\_XID\_EFEC\_PRIORITYSTORERX1

TCAN4x5x\_Data\_Structs.h, 134

TCAN4x5x\_XID\_EFEC\_REJECTMATCH

TCAN4x5x\_Data\_Structs.h, 134

TCAN4x5x\_XID\_EFEC\_STORERX0

TCAN4x5x\_Data\_Structs.h, 134

TCAN4x5x\_XID\_EFEC\_STORERX1

TCAN4x5x\_Data\_Structs.h, 134

TCAN4x5x\_XID\_EFEC\_STORERXBUFORDEBUG

TCAN4x5x\_Data\_Structs.h, 134

TCAN4x5x\_XID\_EFEC\_Values

TCAN4x5x\_Data\_Structs.h, 134

TCAN4x5x\_XID\_EFT\_CLASSIC

TCAN4x5x\_Data\_Structs.h, 135

TCAN4x5x\_XID\_EFT\_DUALID

TCAN4x5x\_Data\_Structs.h, 135

TCAN4x5x\_XID\_EFT\_RANGE

TCAN4x5x\_Data\_Structs.h, 135

TCAN4x5x\_XID\_EFT\_RANGENOMASK

TCAN4x5x\_Data\_Structs.h, 135

TCAN4x5x\_XID\_EFT\_Values

TCAN4x5x\_Data\_Structs.h, 134

TcanInterface, 103

AHB\_READ\_32, 104

AHB\_READ\_BURST\_END, 104

AHB\_READ\_BURST\_READ, 104

AHB\_READ\_BURST\_START, 104

AHB\_WRITE\_32, 104

AHB\_WRITE\_BURST\_END, 104

AHB\_WRITE\_BURST\_START, 104

AHB\_WRITE\_BURST\_WRITE, 104

getAddress, 104

getStatus, 104

reset, 104

rstPio, 105

slave, 105

TcanInterface, 103

TCE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 79

TCF

TCAN4x5x\_MCAN\_Interrupts, 85

TCFE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 79

TDCFilter

TCAN4x5x\_MCAN\_Data\_Timing\_Raw, 72

TDCOffset

TCAN4x5x\_MCAN\_Data\_Timing\_Raw, 72

TEFF

TCAN4x5x\_MCAN\_Interrupts, 85

TEFFE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 79

TEFL

TCAN4x5x\_MCAN\_Interrupts, 85

TEFLE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 79

TEFN

TCAN4x5x\_MCAN\_Interrupts, 85

TEFNE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 79

TEFW

TCAN4x5x\_MCAN\_Interrupts, 85

TEFWE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 79

TEST

TCAN4x5x\_MCAN\_CCCR\_Config, 69

TFE

TCAN4x5x\_MCAN\_Interrupts, 85

TFEE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 79

tfr\_length

TransferDesc, 107

tm\_hour

IrigDecoder::IrigDate, 24

tm\_min

IrigDecoder::IrigDate, 24

tm\_sec

IrigDecoder::IrigDate, 24

tm\_yday

IrigDecoder::IrigDate, 24

tm\_year

IrigDecoder::IrigDate, 24

TOO

TCAN4x5x\_MCAN\_Interrupts, 85

TOOE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 79

TraceLog

AlphiBoard.h, 109

TransferDesc, 106

bufLength, 106

dest\_offset, 106

flags, 107

fPolling, 107

src\_offset, 107

tfr\_length, 107

txs\_offset, 107

userSpaceBuffer, 107

TRUE

AlphiDll.h, 111

TSD

TCAN4x5x\_Device\_Interrupts, 65

TSDEN

TCAN4x5x\_Device\_Interrupt\_Enable, 59

TSW

TCAN4x5x\_MCAN\_Interrupts, 85

TSWE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 79

tx\_ioctl

AlteraDma, 17

TxBufferElementSize

TCAN4x5x\_MRAM\_Config, 101

TxBufferNumElements

TCAN4x5x\_MRAM\_Config, 102

txData\_index

AlteraSpi, 21

TxEventFIFONumElements

TCAN4x5x\_MRAM\_Config, 102

TXP

TCAN4x5x\_MCAN\_CCCR\_Config, 69

txs\_offset

TransferDesc, 107

unhookInterruptServiceRoutine

AlphiBoard, 14

UNIX

AlphiDll.h, 111

USE\_AHB\_CODE

TCAN4x5x\_SPI.cpp, 168

userData

MINIPCIE\_DEV\_CTX, 27

userSpaceBuffer

TransferDesc, 107

usleep

AlphiBoard.h, 109

UVIO

TCAN4x5x\_Device\_Interrupts, 65

UVIOEN

TCAN4x5x\_Device\_Interrupt\_Enable, 59

UVSUP

TCAN4x5x\_Device\_Interrupts, 65

UVSUPEN

TCAN4x5x\_Device\_Interrupt\_Enable, 59

verbose

AlphiBoard, 15

VTWD

TCAN4x5x\_Device\_Interrupts, 65

WAIT\_FOR\_IDLE

TCAN4x5x\_SPI.cpp, 168

waitResult

MINIPCIE\_INT\_RESULT, 28

wdErrorToString

PCIeMini\_error.cpp, 176

WDI

TCAN4x5x\_MCAN\_Interrupts, 85

WDIE

TCAN4x5x\_MCAN\_Interrupt\_Enable, 79

WDT\_Configure

TCAN4550, 54

WDT\_Disable

TCAN4550, 54

WDT\_Enable

TCAN4550, 54

WDT\_Read

TCAN4550, 54

WDT\_Reset

TCAN4550, 54

WDTO

TCAN4x5x\_Device\_Interrupts, 66

WDTOEN

TCAN4x5x\_Device\_Interrupt\_Enable, 60

WKERR

TCAN4x5x\_Device\_Interrupts, 66

WKERREN

TCAN4x5x\_Device\_Interrupt\_Enable, 60

WKRQ

TCAN4x5x\_Device\_Interrupts, 66

word

TCAN4x5x\_Device\_Interrupt\_Enable, 60

TCAN4x5x\_Device\_Interrupts, 66

TCAN4x5x\_MCAN\_CCCR\_Config, 69

TCAN4x5x\_MCAN\_Interrupt\_Enable, 80

TCAN4x5x\_MCAN\_Interrupts, 86

TCAN4x5x\_MCAN\_SID\_Filter, 95

wordSize

AlteraSpi, 21

x64/Debug/PCIeMini\_CAN\_FD\_lib.vcxproj.FileListAbsolute.txt, 169

XIDNumElements

TCAN4x5x\_MRAM\_Config, 102

XTD

TCAN4x5x\_MCAN\_RX\_Header, 92

TCAN4x5x\_MCAN\_TX\_Header, 97