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.

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 8
PCIeMini_CAN_FD 37

AlteraDma 16
AlteraSpi 18
TcanInterface 103

BoardVersion 22
IrigDecoder::IrigDate 24
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 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
TransferDesc 106

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) 8
AlteraDma (Low level SPI interface to the SPI hardware) 16
AlteraSpi (Low level SPI interface to the SPI hardware) 18
BoardVersion (Board Hardware identification and version) 22
IrigDecoder::IrigDate 24
IrigDecoder 25
LinearAddress (Memory Segment Descriptor) 26
MINIPCIE_DEV_CTX (Minipcie Device Information Structure) 27
MINIPCIE_INT_RESULT (Interrupt result information structure) 28
ParallelInput (Alphi Avalon Pio controller class) 30
PcieCra (PCIe CRA module controller class) 33
PCIeMini_CAN_FD (PCIeMini_CAN_FD controller board object) 37
TCAN4550 41
TCAN4x5x_Device_Interrupt_Enable (Struct containing the device interrupt enable bit field) 56
TCAN4x5x_Device_Interrupts (Struct containing the device interrupt bit field) 61
TCAN4x5x_MCAN_CCCR_Config (Struct containing the bit fields of the MCAN CCCR register) 67
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) 71
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) 73
TCAN4x5x_MCAN_Interrupt_Enable (Struct containing the MCAN interrupt enable bit field) 75
TCAN4x5x_MCAN_Interrupts (Struct containing the MCAN interrupt bit field) 81
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) 87
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) 89
TCAN4x5x_MCAN_RX_Header (CAN message header) 91
TCAN4x5x_MCAN_SID_Filter (Standard ID filter struct) 94
TCAN4x5x_MCAN_TX_Header (CAN message header for transmitted messages) 96
TCAN4x5x_MCAN_XID_Filter (Extended ID filter struct) 98
TCAN4x5x_MRAM_Config (Defines the number of MRAM elements and the size of the elements) 100
TcanInterface (This class implements the TCAN4550 SPI interface) 103
TransferDesc (Structure containing the details of a DMA transaction) 106

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) 108

C:/Alphi/PCIeMiniSoftware/include/AlphiDll.h (Utility DLL definitions) 111

C:/Alphi/PCIeMiniSoftware/include/AlphiErrorCodes.h (Description of the Error Codes used by the libraries) 112

C:/Alphi/PCIeMiniSoftware/include/AlteraDma.h (Description of the low-level access routines to the SPI) 117

C:/Alphi/PCIeMiniSoftware/include/AlteraSpi.h (Description of the low-level access routines to the SPI) 121

C:/Alphi/PCIeMiniSoftware/include/IrigDecoder.h (Irig Decoder class to get time) 123

C:/Alphi/PCIeMiniSoftware/include/ParallelInput.h (Description of the Alphi Parallel Input class) 124

C:/Alphi/PCIeMiniSoftware/include/PcieCra.h (PCIe interface CRA class) 125

C:/Alphi/PCIeMiniSoftware/include/PCIeMini_CAN_FD.h (Definitition of the PCIeMini_CAN_FD board class) 126

C:/Alphi/PCIeMiniSoftware/include/TCAN4550.h (This file contains TCAN4550 functions) 128

C:/Alphi/PCIeMiniSoftware/include/TCAN4x5x_Data_Structs.h (This file contains the TCAN4x5x data structures) 131

C:/Alphi/PCIeMiniSoftware/include/TCAN4x5x_Reg.h (This file contains the register definitions for the TCAN4x5x Family) 136

C:/Alphi/PCIeMiniSoftware/include/TCAN4x5x_SPI.h (This file is responsible for abstracting the lower-level microcontroller SPI read and write functions) 164

dllmain.cpp 165

PCIe_Mini_CAN_FD.cpp (Implementation of the PCIeMini_CAN_FD board class) 166

TCAN4550.cpp (This file contains TCAN4550 functions, and relies on the TCAN4x5x_SPI abstraction functions Additional Feature Sets of TCAN4550 vs TCAN4x5x:) 167

TCAN4x5x_SPI.cpp (This file is responsible for abstracting the lower-level microcontroller SPI read and write functions) 168

C:/Alphi/PCIeMiniSoftware/PCIeMini_lib/AlphiBoard.cpp (Implementation of the base PCIe board class with Jungo driver and Altera PCIe hardware) 170

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) 171

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) 172

C:/Alphi/PCIeMiniSoftware/PCIeMini_lib/AlteraDma.cpp (Implementation of the DMA block controller) 173

C:/Alphi/PCIeMiniSoftware/PCIeMini_lib/AlteraSpi.cpp (Implementation of the low-level access routines to the SPI) 174

C:/Alphi/PCIeMiniSoftware/PCIeMini_lib/PcieCra.cpp (PCIe interface CRA class) 175

C:/Alphi/PCIeMiniSoftware/PCIeMini_lib/PCIeMini_error.cpp (Error code to string conversion) 176

C:/Alphi/PCIeMiniSoftware/PCIeMini_lib/TestProgram.cpp (Utility program class for the test programs) 177

Features

CAN Bus

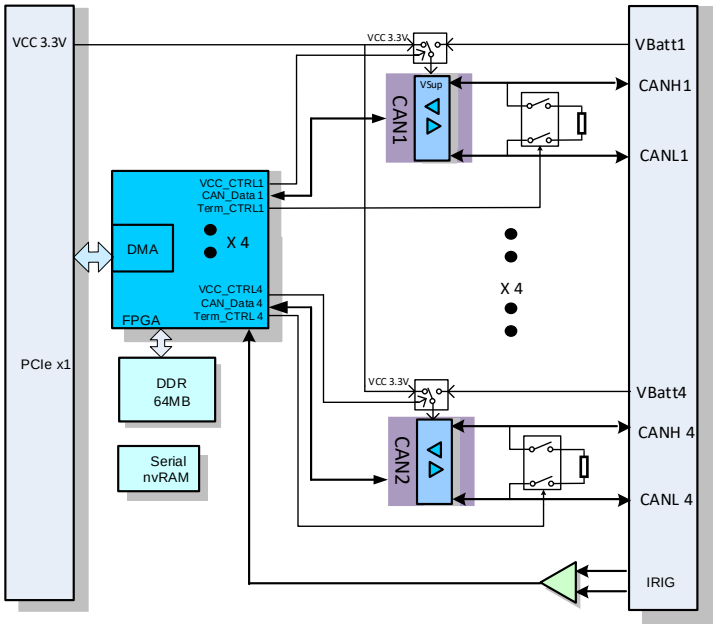
- Four channel CAN controller
- Supports classic CAN and CAN FD
- Complies with
  - ISO11898-1:2015
  - M_CAN Revision 3.2.1.1
- Meets requirements of ISO11898-2:2016
- Data rates from 40Kbps to 8Mbps
- Software programmable speed selection
- Independent Software controlled
  - 120 Ohm Termination
  - 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).

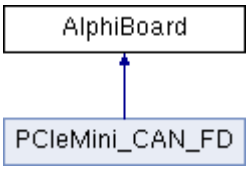
The ***PCIe-Mini-CAN-FD*** device includes many protection features providing device and CAN bus robustness. These features include failsafe mode, internal dominant state timeout, wide bus operating range and a timeout watchdog as examples.



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:

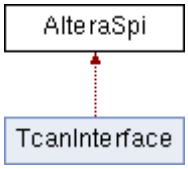
- C:/Alphi/PCIeMiniSoftware/include/**AlteraDma.h**
- C:/Alphi/PCIeMiniSoftware/PCIeMini_lib/**AlteraDma.cpp**

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:

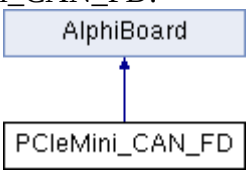
- C:/Alphi/PCIeMiniSoftware/include/**PcieCra.h**
- C:/Alphi/PCIeMiniSoftware/PCIeMini_lib/**PcieCra.cpp**

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

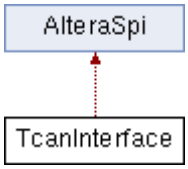
Extended ID Number of Filter Elements: The number of 29-bit filters the user would like
Valid range is: 0 to 64.

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

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_DRIV
    ER_NAME, 170
  MINIPCIE_ARINC429_DEFAULT_LICE
    NSE_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_ENABL
E_VAL
AlphiErrorCodes.h, 115
ERRCODE_INVALID_SSM
AlphiErrorCodes.h, 115
ERRCODE_INVALID_STROBE_MODE
AlphiErrorCodes.h, 115
ERRCODE_INVALID_TIME_BOUNCE_VA
L
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_CANBUSTERMOP
EN
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_CANBUSTERMOP
EN
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_DI
V1
TCAN4x5x_Reg.h, 147
REG_BITS_DEVICE_MODE_CLKOUT_DI
V2
TCAN4x5x_Reg.h, 147
REG_BITS_DEVICE_MODE_CLKOUT_MA
SK
TCAN4x5x_Reg.h, 147
REG_BITS_DEVICE_MODE_DEVICE_RES
ET
TCAN4x5x_Reg.h, 147
REG_BITS_DEVICE_MODE_DEVICEMOD
E_MASK
TCAN4x5x_Reg.h, 147
REG_BITS_DEVICE_MODE_DEVICEMOD
E_NORMAL
TCAN4x5x_Reg.h, 147
REG_BITS_DEVICE_MODE_DEVICEMOD
E_SLEEP
TCAN4x5x_Reg.h, 147
REG_BITS_DEVICE_MODE_DEVICEMOD
E_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_OVER
WRITE
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_AL
WAYS_0
TCAN4x5x_Reg.h, 155
REG_BITS_MCAN_TSCC_COUNTER_EXT
ERNAL
TCAN4x5x_Reg.h, 155
REG_BITS_MCAN_TSCC_COUNTER_USE
_TCP
TCAN4x5x_Reg.h, 155
REG_BITS_MCAN_TSCC_PRESCALER_M
ASK
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_CONT
ROLLER
TCAN4550.h, 129
TCAN4x5x_Device_Test_Mode_Enum
TCAN4550.h, 129
TCAN4x5x_DEVICE_TEST_MODE_NORM
AL
TCAN4550.h, 129
TCAN4x5x_DEVICE_TEST_MODE_PHY
TCAN4550.h, 129
TCAN4x5x_DEVICE_VERIFY_CONFIGUR
ATION_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_CANBUSFAUL
T, 146
REG_BITS_DEVICE_IR_CANBUSGND,
146
REG_BITS_DEVICE_IR_CANBUSNOM,
146
REG_BITS_DEVICE_IR_CANBUSOPEN,
146
REG_BITS_DEVICE_IR_CANBUSTERM
OPEN, 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_R
ESET, 147
REG_BITS_DEVICE_MODE_DEVICEM
ODE_MASK, 147
REG_BITS_DEVICE_MODE_DEVICEM
ODE_NORMAL, 147
REG_BITS_DEVICE_MODE_DEVICEM
ODE_SLEEP, 147
REG_BITS_DEVICE_MODE_DEVICEM
ODE_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_FUN
C_MASK, 148
REG_BITS_DEVICE_MODE_GPO1_FUN
C_MCAN_INT1, 148
REG_BITS_DEVICE_MODE_GPO1_FUN
C_SPI_INT, 148
REG_BITS_DEVICE_MODE_GPO1_FUN
C_UVLO_THERM, 148
REG_BITS_DEVICE_MODE_GPO1_MO
DE_CLKOUT, 148
REG_BITS_DEVICE_MODE_GPO1_MO
DE_GPI, 148
REG_BITS_DEVICE_MODE_GPO1_MO
DE_GPO, 148
REG_BITS_DEVICE_MODE_GPO1_MO
DE_MASK, 148
REG_BITS_DEVICE_MODE_GPO2_CAN
_FAULT, 148
REG_BITS_DEVICE_MODE_GPO2_MAS
K, 148
REG_BITS_DEVICE_MODE_GPO2_MC
AN_INT0, 148
REG_BITS_DEVICE_MODE_GPO2_NIN
T, 148
REG_BITS_DEVICE_MODE_GPO2_WD
T, 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_C
ONFIG_INH, 148
REG_BITS_DEVICE_MODE_NWKRQ_C
ONFIG_MASK, 148
REG_BITS_DEVICE_MODE_NWKRQ_C
ONFIG_WKRQ, 148
REG_BITS_DEVICE_MODE_NWKRQ_V
OLT_INTERNAL, 148
REG_BITS_DEVICE_MODE_NWKRQ_V
OLT_MASK, 148
REG_BITS_DEVICE_MODE_NWKRQ_V
OLT_VIO, 148
REG_BITS_DEVICE_MODE_SWE_DIS,
148
REG_BITS_DEVICE_MODE_SWE_EN,
148
REG_BITS_DEVICE_MODE_SWE_MAS
K, 148
REG_BITS_DEVICE_MODE_TESTMOD
E_CONTROLLER, 149
REG_BITS_DEVICE_MODE_TESTMOD
E_DIS, 149
REG_BITS_DEVICE_MODE_TESTMOD
E_EN, 149
REG_BITS_DEVICE_MODE_TESTMOD
E_ENMASK, 149
REG_BITS_DEVICE_MODE_TESTMOD
E_MASK, 149
REG_BITS_DEVICE_MODE_TESTMOD
E_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

