Can Driver

AUTHOR: **Islam Ehab Ezzat**
Version **1.0.0**
Sun Sep 5 2021

Data Structure Index

Data Structures

Here are the data structures with brief descriptions:

File Index

File List

Here is a list of all files with brief descriptions:

Data Structure Documentation

Can_BitTimingConfig Struct Reference

`#include <CAN_Init.h>`

Data Fields

uint32 SyncPropPhase1Seg
uint32 Phase2Seg
uint32 SJW
uint32 BaudRate

Field Documentation

uint32 Can_BitTimingConfig::BaudRate

Member 'SJW' used to hold (Re-)Sync. Jump Width value

Note

: This value is from 1 to 4

uint32 Can_BitTimingConfig::Phase2Seg

Member 'SyncPropPhase1Seg' used to hold the sum of Sync. , Propagation and Phase Buffer 1 Segments

Note

: This value is from 2 to 16

uint32 Can_BitTimingConfig::SJW

Member 'Phase2Seg' used to hold the value of Phase Buffer 2 Segment in time quanta.

Note

: This value is from 1 to 8

uint32 Can_BitTimingConfig::SyncPropPhase1Seg

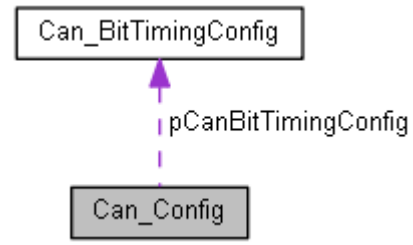
The documentation for this struct was generated from the following file:

Drivers/CAN/**CAN_Init.h**

Can_Config Struct Reference

`#include <CAN_Init.h>`

Collaboration diagram for Can_Config:



Data Fields

Can_BitTimingConfig * pCanBitTimingConfig
Can_Interrupts CanInterrupts
uint32 ClkValue
uint32 CanBitRate
uint8 CanNum

Field Documentation

uint32 Can_Config::CanBitRate

Member 'ClkValue' used to calculate the required Bit Rate for CAN Module

Note

This value should given by a API used to get system clk (Future work)

Can_Interrupts Can_Config::CanInterrupts

Member 'pCanBitTimingConfig' is a pointer to BitTiming structure used to hold CAN Bit Timing values

uint8 Can_Config::CanNum

Member 'CanBitRate' used to indicate the required bit rate for CAN

uint32 Can_Config::ClkValue

Member 'CanInterrupts' used to decide which interrupts should be used in CAN Module You can find this value **Can_Interrupts**

Can_BitTimingConfig* Can_Config::pCanBitTimingConfig

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

Drivers/CAN/**CAN_Init.h**

Can_MessageObjectConfig Struct Reference

`#include <CAN_Init.h>`

Data Fields

Can_MessageObjectType MessageObjectType
Can_MessageObjectSettings MessageObjectSettings
uint32 MessageId
uint32 MessageIdMask
uint8 * pMsgData
uint8 DataLengthCode
uint8 MessageObjectId

Field Documentation

uint8 Can_MessageObjectConfig::DataLengthCode

Member 'pMsgData' used as a pointer to message object's data

uint32 Can_MessageObjectConfig::MessageId

Member 'MessageObjectSettings' used to hold several settings of the message object defined by the user

Note

: You can find this value **Can_MessageObjectSettings**

uint32 Can_MessageObjectConfig::MessageIdMask

Member 'MessageId' used as the CAN Message identifier which is in the CAN Frame. This ID Can be 11 or 29 bit (Extended)

uint8 Can_MessageObjectConfig::MessageObjectId

Member 'DataLengthCode' used to decide the data length needed

Can_MessageObjectSettings Can_MessageObjectConfig::MessageObjectSettings

Member 'Can_MessageObjectType' used to decide which Message object type should be used

Note

: you can find this value **Can_MessageObjectType**

Can_MessageObjectType Can_MessageObjectConfig::MessageObjectType

uint8* Can_MessageObjectConfig::pMsgData

Member 'MessageIdMask' used when identifier filtering is enabled

---

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

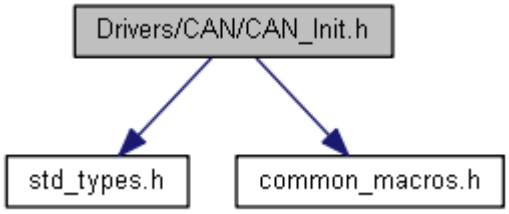
Drivers/CAN/**CAN_Init.h**

File Documentation

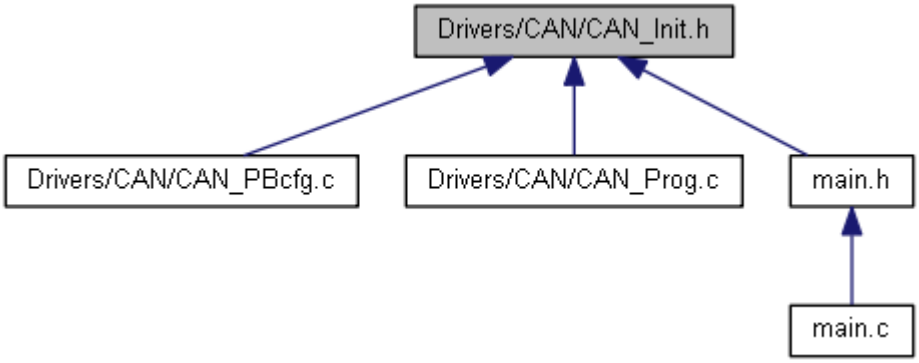
Drivers/CAN/CAN_Init.h File Reference

```
#include "std_types.h"
#include "common_macros.h"
```

Include dependency graph for CAN_Init.h:



This graph shows which files directly or indirectly include this file:



Data Structures

struct **Can_MessageObjectConfig**
struct **Can_BitTimingConfig**
struct **Can_Config**

Macros

#define **INTERRUPT_ENABLE_POS** ((**uint8**)0x1) /* CAN Interrupt Enable Bit Position in CAN_CTL */
#define **STATUS_INTERRUPT_ENABLE_POS** ((**uint8**)0x2) /* Status Interrupt Enable Bit Position in CAN_CTL */
#define **ERROR_INTERRUPT_ENABLE_POS** ((**uint8**)0x3) /* Error Interrupt Enable Bit Position in CAN_CTL */
#define **INTERRUPT_ENABLE_VAL** ((**uint8**)0x1) /* Interrupts enabled Value */
#define **STATUS_INTERRUPT_ENABLE_VAL**
#define **ERROR_INTERRUPT_ENABLE_VAL**
#define **RECEIVE_INTERRUPT_ENABLE_VAL** ((**uint8**)0x1) /* Set INTPND Bit in MCTL reg after receiving frame */
#define **TRANSMIT_INTERRUPT_ENABLE_VAL** ((**uint8**)0x1) /* Set INTPND Bit in MCTL reg after Tx frame */
#define **RECEIVE_INTERRUPT_ENABLE_POS** ((**uint8**)0xA) /* Receive Interrupt Enable Bit Position */
#define **TRANSMIT_INTERRUPT_ENABLE_POS** ((**uint8**)0xB) /* Transmit Interrupt Enable Bit Position */
#define **CAN_MODULE_0** ((**uint8**)0x1)
#define **CAN_MODULE_1** ((**uint8**)0x2)

Enumerations

enum **Can_MessageObjectSettings** {
    CanSettingsTransmitInterruptEnable = 0x001,
    CanSettingsReceiveInterruptEnable = 0x002,
    CanSettingsExtendedId = 0x004,
    CanSettingsUseIdFilter = 0x008,
    CanSettingsNewData = 0x080,
    CanSettingsDataLost = 0x100,

CanSettingsUseDirectionFilter = 0x018,
CanSettingsUseExtendedFilter = 0x028,
CanSettingsRemoteFrame = 0x040,
CanSettingsFIFO = 0x200,
CanSettingsNoSettingsSet = 0x000,
CanSettingsClearPendingInterrupt = 0x400
}

enum **Can_Interrupts** { **CanInterruptsDisable** = 0, **CanInterruptMasterEnable** = ((INTERRUPT_ENABLE_VAL) << (INTERRUPT_ENABLE_POS)), **CanInterruptStatusEnable** = ((STATUS_INTERRUPT_ENABLE_VAL) << (STATUS_INTERRUPT_ENABLE_POS)), **CanInterruptErrorEnable** = ((ERROR_INTERRUPT_ENABLE_VAL) << (ERROR_INTERRUPT_ENABLE_POS)), **CanInterruptTransmit** = ((TRANSMIT_INTERRUPT_ENABLE_VAL) << (TRANSMIT_INTERRUPT_ENABLE_POS)), **CanInterruptReceive** = ((RECEIVE_INTERRUPT_ENABLE_VAL) << (RECEIVE_INTERRUPT_ENABLE_POS)) }

enum **Can_MessageObjectType** { **MessageObjectTypeTransmit**, **MessageObjectTypeTransmitRemote**, **MessageObjectTypeReceive**, **MessageObjectTypeReceiveRemote**, **MessageObjectTypeReceiveRemoteAutoTransmit** }

Functions

uint8 Can_Init (const **Can_Config** *ConfigPtr)
uint8 Can_Transmit (const **Can_Config** *ConfigPtr, const **Can_MessageObjectConfig** *MessageConfigPtr)
uint8 Can_Receive (const **Can_Config** *ConfigPtr, **Can_MessageObjectConfig** *MessageConfigPtr)

Variables

Can_Config CanConfiguration

Macro Definition Documentation

#define CAN_MODULE_0 ((uint8)0x1)

#define CAN_MODULE_1 ((uint8)0x2)

#define ERROR_INTERRUPT_ENABLE_POS ((uint8)0x3) /* Error Interrupt Enable Bit Position in CAN_CTL */

#define ERROR_INTERRUPT_ENABLE_VAL

```
Value:((uint8)0x1)    /* A change in the BOFF or EWARN bits in the CANSTS
                                                    register generates an interrupt
*/
```

#define INTERRUPT_ENABLE_POS ((uint8)0x1) /* CAN Interrupt Enable Bit Position in CAN_CTL */

#define INTERRUPT_ENABLE_VAL ((uint8)0x1) /* Interrupts enabled Value */

#define RECEIVE_INTERRUPT_ENABLE_POS ((uint8)0xA) /* Receive Interrupt Enable Bit Position */

#define RECEIVE_INTERRUPT_ENABLE_VAL ((uint8)0x1) /* Set INTPND Bit in MCTL reg after receiving frame */

#define STATUS_INTERRUPT_ENABLE_POS ((uint8)0x2) /* Status Interrupt Enable Bit Position in CAN_CTL */

#define STATUS_INTERRUPT_ENABLE_VAL

```
Value:((uint8)0x1)    /* A change in the TXOK, RXOK or LEC bits in the
                                                    CANSTS register generates an interrupt
*/
```

#define TRANSMIT_INTERRUPT_ENABLE_POS ((uint8)0xB) /* Transmit Interrupt Enable Bit Position */

#define TRANSMIT_INTERRUPT_ENABLE_VAL ((uint8)0x1) /* Set INTPND Bit in MCTL reg after Tx frame*/

Enumeration Type Documentation

enum Can_Interrupts

Enumerator:

CanInterruptsDisable	CanInterruptsDisable.
CanInterruptMasterEnable	CanInterruptMasterEnable.
CanInterruptStatusEnable	CanInterruptStatusEnable.
CanInterruptErrorEnable	CanInterruptErrorEnable.
CanInterruptTransmit	CanInterruptTransmit.
CanInterruptReceive	CanInterruptReceive.

enum Can_MessageObjectSettings

Enumerator:

CanSettingsTransmitInterruptEnable	
CanSettingsReceiveInterruptEnable	
CanSettingsExtendedId	
CanSettingsUseIdFilter	
CanSettingsNewData	
CanSettingsDataLost	
CanSettingsUseDirectionFilter	
CanSettingsUseExtendedFilter	
CanSettingsRemoteFrame	
CanSettingsFIFO	
CanSettingsNoSettingsSet	
CanSettingsClearPendingInterrupt	

enum Can_MessageObjectType

Enumerator:

MessageObjectTypeTransmit	MessageObjectTypeTransmit.

MessageObjectTypeTransmitRemote	MessageObjectTypeTransmitRemote.
MessageObjectTypeReceive	MessageObjectTypeReceive.
MessageObjectTypeReceiveRemote	MessageObjectTypeReceiveRemote.
MessageObjectTypeReceiveRemoteAutoTransmit	MessageObjectTypeReceiveRemoteAutoTransmit.

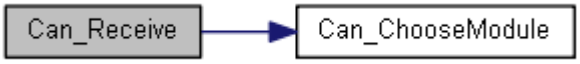
Function Documentation

uint8 Can_Init (const Can_Config * *ConfigPtr*)

Here is the call graph for this function:

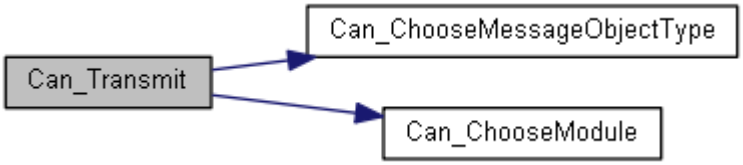
uint8 Can_Receive (const Can_Config * *ConfigPtr*, Can_MessageObjectConfig * *MessageConfigPtr*)

Here is the call graph for this function:



uint8 Can_Transmit (const Can_Config * *ConfigPtr*, const Can_MessageObjectConfig * *MessageConfigPtr*)

Here is the call graph for this function:



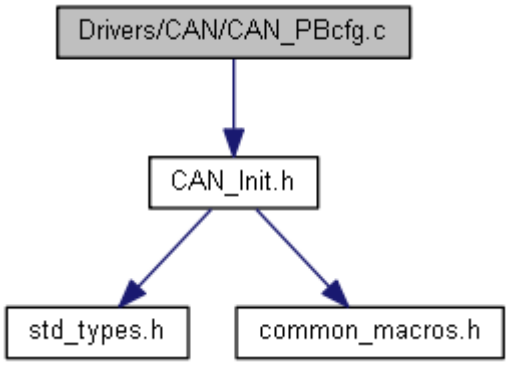
Variable Documentation

Can_Config CanConfiguration

Drivers/CAN/CAN_PBcfg.c File Reference

```
#include "CAN_Init.h"
```

Include dependency graph for CAN_PBcfg.c:



Variables

Can_Config CanConfiguration

Variable Documentation

Can_Config CanConfiguration

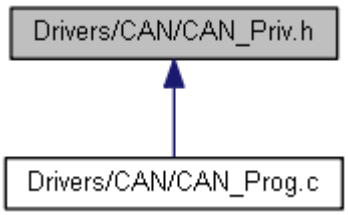
Initial value:= {

```
                                .pCanBitTimingConfig = &(Can_BitTimingConfig){13 ,2 ,1, 2},
                                .CanBitRate          = 500000,
                                 .CanInterrupts      = CanInterruptsDisable,
                                .CanNum              = CAN_MODULE_0,
                                 .ClkValue           = 16000000
```

}

Drivers/CAN/CAN_Priv.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

#define **CTL_REG_INIT_POS** ((**uint8**)0x0) /* Initialization Bit Position */
#define **CTL_REG_IE_POS** ((**uint8**)0x1) /* CAN Interrupt Enable Bit Position */
#define **CTL_REG_SIE_POS** ((**uint8**)0x2) /* Status Interrupt Enable Bit Position */
#define **CTL_REG_EIE_POS** ((**uint8**)0x3) /* Error Interrupt Enable Bit Position */
#define **CTL_REG_DAR_POS** ((**uint8**)0x5) /* Disable Automatic-Retransmission Bit Position */
#define **CTL_REG_CCE_POS** ((**uint8**)0x6) /* Configuration Change Enable Bit Position */
#define **CTL_REG_TEST_POS** ((**uint8**)0x7) /* Test Mode Enable Bit Position */
#define **CTL_REG_INIT_ENABLE** ((**uint8**)0x1) /* Initialization started Value */
#define **CTL_REG_IE_ENABLE** ((**uint8**)0x1) /* Interrupts enabled Value */
#define **CTL_REG_SIE_ENABLE**
#define **CTL_REG_EIE_ENABLE**
#define **CTL_REG_DAR_ENABLE** ((**uint8**)0x1) /* Auto-retransmission is disabled */
#define **CTL_REG_CCE_ENABLE**
#define **CTL_REG_TEST_ENABLE** ((**uint8**)0x1) /* CAN controller is in test mode */
#define **IFCR1_REG_MNUM_POS** ((**uint8**)0x0) /* Message Object Number Bit Position */
#define **IFCRQ_REG_BUSY_POS** ((**uint8**)0xF) /* Busy Flag bit position */
#define **IFCRQ_REG_MNUM_MASK** ((**uint8**)0x3F) /* Message Object Number Mask Value */
#define **IFCRQ_REG_BUSY_ENABLE** ((**uint8**)0x1) /* Set when a write occurs to the message number */
#define **CMSK_REG_DATAB_POS** ((**uint8**)0x0) /* Access Data Byte 4 to 7 Bit Position */
#define **CMSK_REG_DATAA_POS** ((**uint8**)0x1) /* Access Data Byte 0 to 3 Bit Position */
#define **CMSK_REG_NEWDAT_TXRQST_POS** ((**uint8**)0x2) /* NEWDAT / TXRQST Bit Bit Position */
#define **CMSK_REG_CLRINTPND_POS** ((**uint8**)0x3) /* Clear Interrupt Pending Bit Bit Position */
#define **CMSK_REG_CONTROL_POS** ((**uint8**)0x4) /* Access Control Bit Bit Position */
#define **CMSK_REG_ARB_POS** ((**uint8**)0x5) /* Access Arbitration Bits Bit Position */

```
#define CMSK_REG_MASK_POS   ((uint8)0x6)          /* Access Mask Bits Bit Position
    */
#define CMSK_REG_WRNRD_POS   ((uint8)0x7)         /* Write, Not Read Bit Position
    */
#define CMSK_REG_DATAB_ENABLE
#define CMSK_REG_DATAA_ENABLE
#define CMSK_REG_NEWDAT_TXRQST_ENABLE
#define CMSK_REG_CLRINTPND_ENABLE
#define CMSK_REG_CONTROL_ENABLE   ((uint8)0x1)          /* Tx Control bits from MCTL
    Reg to IF Regs              */
#define CMSK_REG_ARB_ENABLE   ((uint8)0x1)          /* Tx ID+DIR+XTD+MSGVAL of msg
    object to IF Regs       */
#define CMSK_REG_MASK_ENABLE   ((uint8)0x1)         /* Tx IDMASK, DIR, MXTD of msg
    object to IF Regs       */
#define CMSK_REG_WRNRD_ENABLE   ((uint8)0x1)        /* Tx Data in IF Regs to CAN Msg
    Object                  */
#define MCTL_REG_DLC_POS   ((uint8)0x0)          /* Data Length Code Bit Position
    */
#define MCTL_REG_EOB_POS   ((uint8)0x7)          /* End of Buffer Bit Position
    */
#define MCTL_REG_TXRQST_POS   ((uint8)0x8)         /* Transmit Request Bit Position
    */
#define MCTL_REG_RMTEN_POS   ((uint8)0x9)         /* Remote Enable Bit Position
    */
#define MCTL_REG_RXIE_POS   ((uint8)0xA)          /* ReceiveInterrupt Enable Bit Position
    */
#define MCTL_REG_TXIE_POS   ((uint8)0xB)          /* Transmit Interrupt Enable Bit Position
    */
#define MCTL_REG_UMASK_POS   ((uint8)0xC)         /* Use Acceptance Mask Bit Position
    */
#define MCTL_REG_INTPND_POS   ((uint8)0xD)         /* Interrupt Pending Bit Position
    */
#define MCTL_REG_MSGLST_POS   ((uint8)0xE)         /* Message Lost Bit Position
    */
#define MCTL_REG_NEWDAT_POS   ((uint8)0xF)         /* New Data Bit Position
    */
#define MCTL_REG_DLC_MASK   ((uint8)0xF)          /* Data Length Code Mask Value
    */
#define MCTL_REG_EOB_ENABLE   ((uint8)0x1)          /* Single or last message object of a FIFO
    Buffer      */
#define MCTL_REG_TXRQST_ENABLE   ((uint8)0x1)         /* Transmission of Msg object is
    req.& not yet done   */
#define MCTL_REG_RMTEN_ENABLE   ((uint8)0x1)         /* Set TXRQST Bit when remote
    frame received          */
#define MCTL_REG_RXIE_ENABLE   ((uint8)0x1)          /* Set INTPND Bit after receiving a
    frame               */
#define MCTL_REG_TXIE_ENABLE   ((uint8)0x1)          /* Set INTPND Bit after Transmitting
    frame               */
#define MCTL_REG_UMASK_ENABLE   ((uint8)0x1)         /* Use mask (MSK, MXTD, MDIR)
    for acceptance filter   */
#define MCTL_REG_INTPND_ENABLE   ((uint8)0x1)         /* This message object. is the source
    of Interrupt    */
#define MCTL_REG_MSGLST_ENABLE
#define MCTL_REG_NEWDAT_ENABLE   ((uint8)0x1)         /* Msg Handler or CPU write new
    data into msg object */
#define ARB1_REG_ID_POS   ((uint8)0x0)          /* Message ID Bit Position
    */
#define ARB1_REG_ID_MASK   ((uint16)0xFFFF)    /* Message ID Mask Value
    */
```

```
#define ARB2_REG_ID_POS   ((uint8)0x0)          /* Message ID Bit Position
    */
#define ARB2_REG_DIR_POS   ((uint8)0xD)         /* Message Direction Bit Position
    */
#define ARB2_REG_XTD_POS   ((uint8)0xE)         /* Extended ID Bit Position
    */
#define ARB2_REG_MSGVAL_POS   ((uint8)0xF)          /* Message Valid Bit Position
    */
#define ARB2_REG_ID_MASK   ((uint16)0x1FFF)    /* Message ID Mask Value
    */
#define ARB2_REG_DIR_ENABLE   ((uint8)0x1)          /* Message Direction is Transmit
    */
#define ARB2_REG_XTD_ENABLE   ((uint8)0x1)          /* Use Extended ID (29 bits)
    */
#define ARB2_REG_MSGVAL_ENABLE   ((uint8)0x1)          /* This Message object is ready for
    msg handler        */
#define MSK1_REG_MSK_POS   ((uint8)0x0)          /* Mask ID Bit Position
    */
#define MSK1_REG_MSK_MASK   ((uint16)0xFFFF)    /* Mask ID Masking Value
    */
#define MSK2_REG_MSK_POS   ((uint8)0x0)          /* Mask ID Bit Position
    */
#define MSK2_REG_MDIR_POS   ((uint8)0xE)          /* Mask Message Direction Bit Position
    */
#define MSK2_REG_MXTD_POS   ((uint8)0xF)          /* Mask Extended ID Bit Position
    */
#define MSK2_REG_MSK_MASK   ((uint16)0x1FFF)    /* Mask Id Masking Value
    */
#define MSK2_REG_MDIR_ENABLE   ((uint8)0x1)          /* DIR bit is used for acceptance
    filtering              */
#define MSK2_REG_MXTD_ENABLE   ((uint8)0x1)          /* XTD bit is used for acceptance
    filtering              */
#define BIT_REG_BRP_POS   ((uint8)0x0)          /* Baud Rate PreScaler Bit Position
    */
#define BIT_REG_SJW_POS   ((uint8)0x6)          /* (Re)Synchronization Jump Width Bit Position
    */
#define BIT_REG_TSEG1_POS   ((uint8)0x8)          /* Time Segment Before Sample Point Bit
    Position       */
#define BIT_REG_TSEG2_POS   ((uint8)0xC)          /* Time Segment after Sample Point Bit
    Position       */
#define BIT_REG_BRP_MASK   ((uint8)0x3F)          /* Baud Rate Mask value
    */
#define BIT_REG_SJW_MASK   ((uint8)0xC0)          /* (Re)Synchronization Jump Width Mask
    value          */
#define BIT_REG_TSEG1_MASK   ((uint16)0x0F00)    /* Time Segment Before Sample Mask
    value                  */
#define BIT_REG_TSEG2_MASK   ((uint16)0x7000)    /* Time Segment After Sample Mask value
    */
#define BRPE_REG_BRPE_POS   ((uint8)0x0)          /* Baud Rate PreScaler Extension bit position
    */
#define BREP_REG_BREP_MASK   ((uint8)0xF)          /* Baud Rate PreScaler Extension Mask
    Value          */
#define RCGC0_REG_CAN_POS   ((uint8)0x18)        /* CAN Clock Bit Position (Bit 24)
    */
#define BAUDRATE_EXTENSION_SHIFT   ((uint8)0x6)          /* CAN Bit timing baud rate
    extension shift value       */
#define ID_11_BIT_MAX_VALUE   ((uint16)0x7FF)    /* ID 11-bit Mode Max Value
    */
#define ID_29_BIT_MSK2_SHIFT_VALUE   ((uint8)0x10)          /* ID 29-Bit Shift value for MSK2
    Register               */
```

```
#define ID_11_BIT_MSK2_SHIFT_VALUE   ((uint8)0x02)       /* ID 11-Bit Shift value for MSK2
    Register                */
#define DATA_SHIFT_VALUE   ((uint8)0x08)       /* Reading data from user buffer shift left value
    */
#define ARB1_REG_FULL_ID   ARB1_REG_ID_MASK   /* use Full ID in Arbitration ARB1 Reg
    value            */
#define ARB2_REG_FULL_ID   ARB2_REG_ID_MASK   /* Use Full ID in Arbitration ARB2 Reg
    value            */
#define ID_29_BIT_FULLY_SPECIFIED   0x1FFFFFFF          /* ID 29-Bit Fully Specified Mask
    Value                */
#define ID_11_BIT_FULLY_SPECIFIED   0x7FF               /* ID 11-Bit Fully Specified Mask
    Value                */
#define DATA_REG_OFFSET_VALUE   ((uint8)0x04)       /* Data Register offset value
    */
```

Macro Definition Documentation

#define ARB1_REG_FULL_ID ARB1_REG_ID_MASK /* use Full ID in Arbitration ARB1 Reg value */

#define ARB1_REG_ID_MASK ((uint16)0xFFFF) /* Message ID Mask Value */

#define ARB1_REG_ID_POS ((uint8)0x0) /* Message ID Bit Position */

#define ARB2_REG_DIR_ENABLE ((uint8)0x1) /* Message Direction is Transmit */

#define ARB2_REG_DIR_POS ((uint8)0xD) /* Message Direction Bit Position */

#define ARB2_REG_FULL_ID ARB2_REG_ID_MASK /* Use Full ID in Arbitration ARB2 Reg value */

#define ARB2_REG_ID_MASK ((uint16)0x1FFF) /* Message ID Mask Value */

#define ARB2_REG_ID_POS ((uint8)0x0) /* Message ID Bit Position */

#define ARB2_REG_MSGVAL_ENABLE ((uint8)0x1) /* This Message object is ready for msg handler */

#define ARB2_REG_MSGVAL_POS ((uint8)0xF) /* Message Valid Bit Position */

#define ARB2_REG_XTD_ENABLE ((uint8)0x1) /* Use Extended ID (29 bits) */

#define ARB2_REG_XTD_POS ((uint8)0xE) /* Extended ID Bit Position */

#define BAUDRATE_EXTENSION_SHIFT ((uint8)0x6) /* CAN Bit timing baud rate extension shift value */

#define BIT_REG_BRP_MASK ((uint8)0x3F) /* Baud Rate Mask value */

#define BIT_REG_BRP_POS ((uint8)0x0) /* Baud Rate PreScaler Bit Position */

#define BIT_REG_SJW_MASK ((uint8)0xC0) /* (Re)Synchronization Jump Width Mask value */

#define BIT_REG_SJW_POS ((uint8)0x6) /* (Re)Synchronization Jump Width Bit Position */

#define BIT_REG_TSEG1_MASK ((uint16)0x0F00) /* Time Segment Before Sample Mask value */

#define BIT_REG_TSEG1_POS ((uint8)0x8) /* Time Segment Before Sample Point Bit Position */

#define BIT_REG_TSEG2_MASK ((uint16)0x7000) /* Time Segment After Sample Mask value */

#define BIT_REG_TSEG2_POS ((uint8)0xC) /* Time Segment after Sample Point Bit Position */

#define BREP_REG_BREP_MASK ((uint8)0xF) /* Baud Rate PreScaler Extension Mask Value */

#define BRPE_REG_BRPE_POS ((uint8)0x0) /* Baud Rate PreScaler Extension bit position */

#define CMSK_REG_ARB_ENABLE ((uint8)0x1) /* Tx ID+DIR+XTD+MSGVAL of msg object to IF Regs */

#define CMSK_REG_ARB_POS ((uint8)0x5) /* Access Arbitration Bits Bit Position */

#define CMSK_REG_CLRINTPND_ENABLE

```
Value:((uint8)0x1)       /* if WRNRD=0,INT Pending status clrd in msg buffer
                                                    if WRNRD=1,INTPND
Bit is cleared in msg object    */
```

#define CMSK_REG_CLRINTPND_POS ((uint8)0x3) /* Clear Interrupt Pending Bit Bit Position */

#define CMSK_REG_CONTROL_ENABLE ((uint8)0x1) /* Tx Control bits from MCTL Reg to IF Regs */

#define CMSK_REG_CONTROL_POS ((uint8)0x4) /* Access Control Bit Bit Position */

#define CMSK_REG_DATAA_ENABLE

```
Value:((uint8)0x1)       /* if WRNRD=0, Tx data bytes 0->3 to msg object
                                                    if WRNRD=1, Tx data
bytes 0->3 to DA, DB Regs     */
```

#define CMSK_REG_DATAA_POS ((uint8)0x1) /* Access Data Byte 0 to 3 Bit Position */

#define CMSK_REG_DATAB_ENABLE

```
Value:((uint8)0x1)       /* if WRNRD=0, Tx data bytes 4->7 to msg object
                                                    if WRNRD=1, Tx data
bytes 4->7 to DA, DB Regs     */
```

#define CMSK_REG_DATAB_POS ((uint8)0x0) /* Access Data Byte 4 to 7 Bit Position */

#define CMSK_REG_MASK_ENABLE ((uint8)0x1) /* Tx IDMASK, DIR, MXTD of msg object to IF Regs */

#define CMSK_REG_MASK_POS ((uint8)0x6) /* Access Mask Bits Bit Position */

#define CMSK_REG_NEWDAT_TXRQST_ENABLE

```
Value:((uint8)0x1)       /* if WRNRD=0, New Data status clrd in msg buffer
                                                    if WRNRD=1, Tx is
requested                        */
```

#define CMSK_REG_NEWDAT_TXRQST_POS ((uint8)0x2) /* NEWDAT / TXRQST Bit Bit Position */

#define CMSK_REG_WRNRD_ENABLE ((uint8)0x1) /* Tx Data in IF Regs to CAN Msg Object */

#define CMSK_REG_WRNRD_POS ((uint8)0x7) /* Write, Not Read Bit Position */

#define CTL_REG_CCE_ENABLE

```
Value:((uint8)0x1)       /* Write accesses to the CANBIT register are
                                                    allowed if the INIT
bit is 1                      */
```

#define CTL_REG_CCE_POS ((uint8)0x6) /* Configuration Change Enable Bit Position */

#define CTL_REG_DAR_ENABLE ((uint8)0x1) /* Auto-retransmission is disabled */

#define CTL_REG_DAR_POS ((uint8)0x5) /* Disable Automatic-Retransmission Bit Position */

#define CTL_REG_EIE_ENABLE

```
Value:((uint8)0x1)       /* A change in the BOFF or EWARN bits in the CANSTS
                                                    register generates
an interrupt                   */
```

#define CTL_REG_EIE_POS ((uint8)0x3) /* Error Interrupt Enable Bit Position */

#define CTL_REG_IE_ENABLE ((uint8)0x1) /* Interrupts enabled Value */

#define CTL_REG_IE_POS ((uint8)0x1) /* CAN Interrupt Enable Bit Position */

#define CTL_REG_INIT_ENABLE ((uint8)0x1) /* Initialization started Value */

#define CTL_REG_INIT_POS ((uint8)0x0) /* Initialization Bit Position */

#define CTL_REG_SIE_ENABLE

```
Value:((uint8)0x1)       /* A change in the TXOK, RXOK or LEC bits in the
                                                        CANSTS register
generates an interrupt            */
```

```
#define CTL_REG_SIE_POS   ((uint8)0x2)          /* Status Interrupt Enable Bit Position
*/

#define CTL_REG_TEST_ENABLE   ((uint8)0x1)          /* CAN controller is in test mode
*/

#define CTL_REG_TEST_POS   ((uint8)0x7)          /* Test Mode Enable Bit Position
*/

#define DATA_REG_OFFSET_VALUE   ((uint8)0x04)        /* Data Register offset value
*/

#define DATA_SHIFT_VALUE   ((uint8)0x08)        /* Reading data from user buffer shift
left value        */

#define ID_11_BIT_FULLY_SPECIFIED   0x7FF                    /* ID 11-Bit Fully Specified
Mask Value                    */

#define ID_11_BIT_MAX_VALUE   ((uint16)0x7FF)     /* ID 11-bit Mode Max Value
*/

#define ID_11_BIT_MSK2_SHIFT_VALUE   ((uint8)0x02)        /* ID 11-Bit Shift value for
MSK2 Register                 */

#define ID_29_BIT_FULLY_SPECIFIED   0x1FFFFFFF            /* ID 29-Bit Fully Specified
Mask Value                    */

#define ID_29_BIT_MSK2_SHIFT_VALUE   ((uint8)0x10)        /* ID 29-Bit Shift value for
MSK2 Register                 */

#define IFCR1_REG_MNUM_POS   ((uint8)0x0)          /* Message Object Number Bit
Position                      */

#define IFCRQ_REG_BUSY_ENABLE   ((uint8)0x1)          /* Set when a write occurs to
the message number        */

#define IFCRQ_REG_BUSY_POS   ((uint8)0xF)          /* Busy Flag bit position
*/

#define IFCRQ_REG_MNUM_MASK   ((uint8)0x3F)         /* Message Object Number
Mask Value                    */

#define MCTL_REG_DLC_MASK   ((uint8)0xF)          /* Data Length Code Mask Value
*/

#define MCTL_REG_DLC_POS   ((uint8)0x0)          /* Data Length Code Bit Position
*/

#define MCTL_REG_EOB_ENABLE   ((uint8)0x1)          /* Single or last message object
of a FIFO Buffer       */

#define MCTL_REG_EOB_POS   ((uint8)0x7)          /* End of Buffer Bit Position
*/
```

#define MCTL_REG_INTPND_ENABLE ((uint8)0x1) /* This message object. is the source of Interrupt */

#define MCTL_REG_INTPND_POS ((uint8)0xD) /* Interrupt Pending Bit Position */

#define MCTL_REG_MSGLST_ENABLE

```
Value:((uint8)0x1)       /* This msg handler stored new msg into this object
                                                    When NEWDAT Was Set,
The CPU has LOST A MESSAGE   */
```

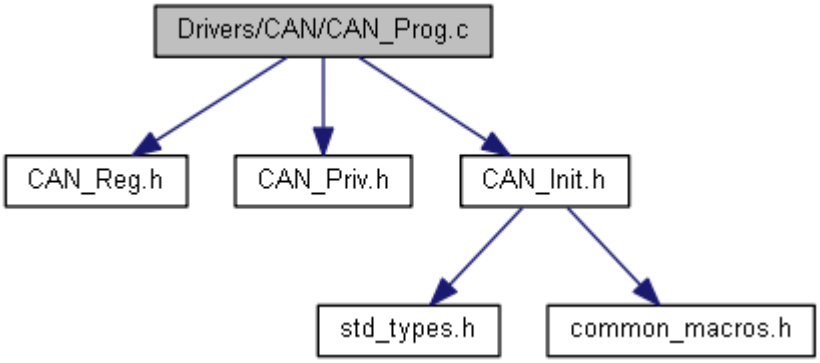
```
#define MCTL_REG_MSGLST_POS   ((uint8)0xE)          /* Message Lost Bit Position
*/

#define MCTL_REG_NEWDAT_ENABLE   ((uint8)0x1)          /* Msg Handler or CPU
write new data into msg object */

#define MCTL_REG_NEWDAT_POS   ((uint8)0xF)          /* New Data Bit Position
*/

#define MCTL_REG_RMTEN_ENABLE   ((uint8)0x1)          /* Set TXRQST Bit when
remote frame received             */

#define MCTL_REG_RMTEN_POS   ((uint8)0x9)          /* Remote Enable Bit Position
*/

#define MCTL_REG_RXIE_ENABLE   ((uint8)0x1)          /* Set INTPND Bit after
receiving a frame                 */

#define MCTL_REG_RXIE_POS   ((uint8)0xA)          /* ReceiveInterrupt Enable Bit
Position                  */

#define MCTL_REG_TXIE_ENABLE   ((uint8)0x1)          /* Set INTPND Bit after
Transmitting frame                */

#define MCTL_REG_TXIE_POS   ((uint8)0xB)          /* Transmit Interrupt Enable Bit
Position                */

#define MCTL_REG_TXRQST_ENABLE   ((uint8)0x1)          /* Transmission of Msg
object is req.& not yet done   */

#define MCTL_REG_TXRQST_POS   ((uint8)0x8)          /* Transmit Request Bit Position
*/

#define MCTL_REG_UMASK_ENABLE   ((uint8)0x1)          /* Use mask (MSK, MXTD,
MDIR) for acceptance filter   */

#define MCTL_REG_UMASK_POS   ((uint8)0xC)          /* Use Acceptance Mask Bit
Position                  */

#define MSK1_REG_MSK_MASK   ((uint16)0xFFFF)     /* Mask ID Masking Value
*/

#define MSK1_REG_MSK_POS   ((uint8)0x0)          /* Mask ID Bit Position
*/

#define MSK2_REG_MDIR_ENABLE   ((uint8)0x1)          /* DIR bit is used for
acceptance filtering              */

#define MSK2_REG_MDIR_POS   ((uint8)0xE)          /* Mask Message Direction Bit
Position                  */

#define MSK2_REG_MSK_MASK   ((uint16)0x1FFF)     /* Mask Id Masking Value
*/
```

```
#define MSK2_REG_MSK_POS   ((uint8)0x0)          /* Mask ID Bit Position
*/

#define MSK2_REG_MXTD_ENABLE   ((uint8)0x1)          /* XTD bit is used for
acceptance filtering                */

#define MSK2_REG_MXTD_POS   ((uint8)0xF)          /* Mask Extended ID Bit Position
*/

#define RCGC0_REG_CAN_POS   ((uint8)0x18)        /* CAN Clock Bit Position (Bit 24)
*/
```

Drivers/CAN/CAN_Prog.c File Reference

```
#include "CAN_Reg.h"
#include "CAN_Priv.h"
#include "CAN_Init.h"
```

Include dependency graph for CAN_Prog.c:



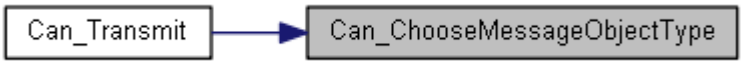
Functions

static void **Can_ChooseModule** (const **Can_Config** *ConfigPtr, volatile **uint32** **Ptr)
static **uint8 Can_ChooseMessageObjectType** (volatile **uint32** **Can_Ptr, const **Can_MessageObjectConfig** *MessageConfigPtr, **uint8** *DataTransmitFlag)
uint8 Can_Init (const **Can_Config** *ConfigPtr)
uint8 Can_Transmit (const **Can_Config** *ConfigPtr, const **Can_MessageObjectConfig** *MessageConfigPtr)
uint8 Can_Receive (const **Can_Config** *ConfigPtr, **Can_MessageObjectConfig** *MessageConfigPtr)

Function Documentation

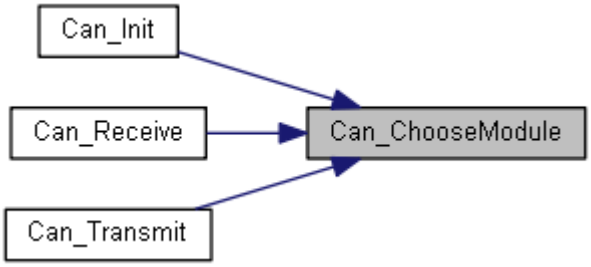
static uint8 Can_ChooseMessageObjectType (volatile uint32 ** *Can_Ptr*, const Can_MessageObjectConfig * *MessageConfigPtr*, uint8 * *DataTransmitFlag*) [static]

Here is the caller graph for this function:



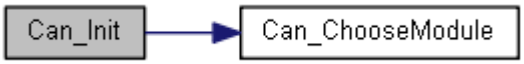
static void Can_ChooseModule (const Can_Config * *ConfigPtr*, volatile uint32 ** *Ptr*) [static]

Here is the caller graph for this function:



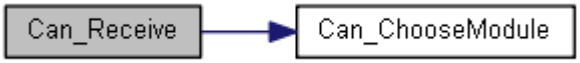
uint8 Can_Init (const Can_Config * *ConfigPtr*)

Here is the call graph for this function:



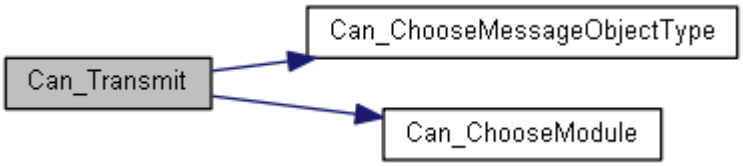
uint8 Can_Receive (const Can_Config * *ConfigPtr*, Can_MessageObjectConfig * *MessageConfigPtr*)

Here is the call graph for this function:



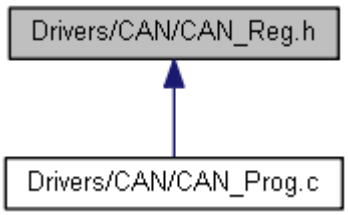
uint8 Can_Transmit (const Can_Config * *ConfigPtr*, const Can_MessageObjectConfig * *MessageConfigPtr*)

Here is the call graph for this function:



Drivers/CAN/CAN_Reg.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

#define **CAN_0_BASE_ADDRESS** 0x40040000
#define **CAN_1_BASE_ADDRESS** 0x40041000
#define **CAN_CTL_REG_OFFSET** 0x000 /* CAN Control */
#define **CAN_STS_REG_OFFSET** 0x004 /* CAN Status */
#define **CAN_ERR_REG_OFFSET** 0x008 /* CAN Error Counter */
#define **CAN_BIT_REG_OFFSET** 0x00C /* CAN Bit Timing */
#define **CAN_INT_REG_OFFSET** 0x010 /* CAN Interrupt */
#define **CAN_TST_REG_OFFSET** 0x014 /* CAN Test */
#define **CAN_BPRE_REG_OFFSET** 0x018 /* CAN Baud Rate PreScaler Extension */
#define **CAN_IF1CRQ_REG_OFFSET** 0x020 /* CAN IF1 Command Request */
#define **CAN_IF1CMSK_REG_OFFSET** 0x024 /* CAN IF1 Command Mask */
#define **CAN_IF1MSK1_REG_OFFSET** 0x028 /* CAN IF1 Mask 1 */
#define **CAN_IF1MSK2_REG_OFFSET** 0x02C /* CAN IF1 Mask 2 */
#define **CAN_IF1ARB1_REG_OFFSET** 0x030 /* CAN IF1 Arbitration 1 */
#define **CAN_IF1ARB2_REG_OFFSET** 0x034 /* CAN IF1 Arbitration 2 */
#define **CAN_IF1MCTL_REG_OFFSET** 0x038 /* CAN IF1 Message Control */
#define **CAN_IF1DA1_REG_OFFSET** 0x03C /* CAN IF1 Data A1 */
#define **CAN_IF1DA2_REG_OFFSET** 0x040 /* CAN IF1 Data A2 */
#define **CAN_IF1DB1_REG_OFFSET** 0x044 /* CAN IF1 Data B1 */
#define **CAN_IF1DB2_REG_OFFSET** 0x048 /* CAN IF1 Data B2 */
#define **CAN_IF2CRQ_REG_OFFSET** 0x080 /* CAN IF2 Command Request */
#define **CAN_IF2CMSK_REG_OFFSET** 0x084 /* CAN IF2 Command Mask */
#define **CAN_IF2MSK1_REG_OFFSET** 0x088 /* CAN IF2 Mask 1 */
#define **CAN_IF2MSK2_REG_OFFSET** 0x08C /* CAN IF2 Mask 2 */
#define **CAN_IF2ARB1_REG_OFFSET** 0x090 /* CAN IF2 Arbitration 1 */
#define **CAN_IF2ARB2_REG_OFFSET** 0x094 /* CAN IF2 Arbitration 2 */
#define **CAN_IF2MCTL_REG_OFFSET** 0x098 /* CAN IF2 Message Control */

```
#define CAN_IF2DA1_REG_OFFSET   0x09C               /* CAN IF2 Data A1
    */
#define CAN_IF2DA2_REG_OFFSET   0x0A0               /* CAN IF2 Data A2
    */
#define CAN_IF2DB1_REG_OFFSET   0x0A4               /* CAN IF2 Data B1
    */
#define CAN_IF2DB2_REG_OFFSET   0x0A8               /* CAN IF2 Data B2
    */
#define CAN_TXRQ1_REG_OFFSET   0x100               /* CAN Transmission Request 1
    */
#define CAN_TXRQ2_REG_OFFSET   0x104               /* CAN Transmission Request 2
    */
#define CAN_NWDA1_REG_OFFSET   0x120               /* CAN New Data 1
    */
#define CAN_NWDA2_REG_OFFSET   0x124               /* CAN New Data 2
    */
#define CAN_MSG1INT_REG_OFFSET   0x140               /* CAN Message 1 Interrupt Pending
    */
#define CAN_MSG2INT_REG_OFFSET   0x144               /* CAN Message 2 Interrupt Pending
    */
#define CAN_MSG1VAL_REG_OFFSET   0x160               /* CAN Message 1 Valid
    */
#define CAN_MSG2VAL_REG_OFFSET   0x164               /* CAN Message 2 Valid
    */
#define SYSCTL_RCGC0_REG   (*((volatile unsigned long*)0x400FE100))        /* Note: CAN Bits
    from bit 24 */
```

Macro Definition Documentation

#define CAN_0_BASE_ADDRESS 0x40040000

#define CAN_1_BASE_ADDRESS 0x40041000

**#define CAN_BIT_REG_OFFSET 0x00C /* CAN Bit Timing
*/**

**#define CAN_BPRE_REG_OFFSET 0x018 /* CAN Baud Rate PreScaler
Extension */**

**#define CAN_CTL_REG_OFFSET 0x000 /* CAN Control
*/**

**#define CAN_ERR_REG_OFFSET 0x008 /* CAN Error Counter
*/**

**#define CAN_IF1ARB1_REG_OFFSET 0x030 /* CAN IF1 Arbitration 1
*/**

**#define CAN_IF1ARB2_REG_OFFSET 0x034 /* CAN IF1 Arbitration 2
*/**

**#define CAN_IF1CMSK_REG_OFFSET 0x024 /* CAN IF1 Command Mask
*/**

**#define CAN_IF1CRQ_REG_OFFSET 0x020 /* CAN IF1 Command Request
*/**

**#define CAN_IF1DA1_REG_OFFSET 0x03C /* CAN IF1 Data A1
*/**

**#define CAN_IF1DA2_REG_OFFSET 0x040 /* CAN IF1 Data A2
*/**

**#define CAN_IF1DB1_REG_OFFSET 0x044 /* CAN IF1 Data B1
*/**

**#define CAN_IF1DB2_REG_OFFSET 0x048 /* CAN IF1 Data B2
*/**

**#define CAN_IF1MCTL_REG_OFFSET 0x038 /* CAN IF1 Message Control
*/**

**#define CAN_IF1MSK1_REG_OFFSET 0x028 /* CAN IF1 Mask 1
*/**

**#define CAN_IF1MSK2_REG_OFFSET 0x02C /* CAN IF1 Mask 2
*/**

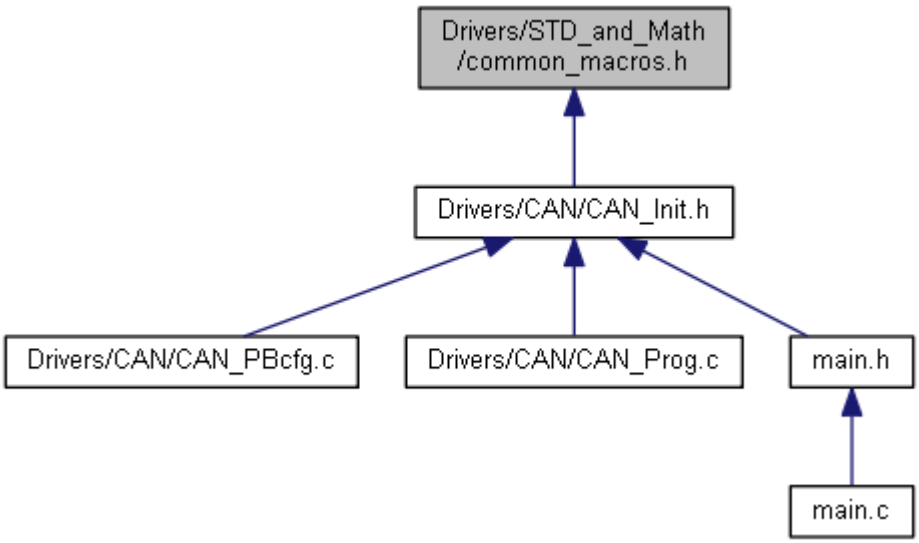
**#define CAN_IF2ARB1_REG_OFFSET 0x090 /* CAN IF2 Arbitration 1
*/**

```
#define CAN_IF2ARB2_REG_OFFSET   0x094                /* CAN IF2 Arbitration 2
*/

#define CAN_IF2CMSK_REG_OFFSET   0x084                /* CAN IF2 Command Mask
*/

#define CAN_IF2CRQ_REG_OFFSET   0x080                 /* CAN IF2 Command Request
*/

#define CAN_IF2DA1_REG_OFFSET   0x09C                 /* CAN IF2 Data A1
*/

#define CAN_IF2DA2_REG_OFFSET   0x0A0                 /* CAN IF2 Data A2
*/

#define CAN_IF2DB1_REG_OFFSET   0x0A4                 /* CAN IF2 Data B1
*/

#define CAN_IF2DB2_REG_OFFSET   0x0A8                 /* CAN IF2 Data B2
*/

#define CAN_IF2MCTL_REG_OFFSET   0x098                /* CAN IF2 Message Control
*/

#define CAN_IF2MSK1_REG_OFFSET   0x088                /* CAN IF2 Mask 1
*/

#define CAN_IF2MSK2_REG_OFFSET   0x08C                /* CAN IF2 Mask 2
*/

#define CAN_INT_REG_OFFSET   0x010                  /* CAN Interrupt
*/

#define CAN_MSG1INT_REG_OFFSET   0x140                /* CAN Message 1 Interrupt
Pending      */

#define CAN_MSG1VAL_REG_OFFSET   0x160                /* CAN Message 1 Valid
*/

#define CAN_MSG2INT_REG_OFFSET   0x144                /* CAN Message 2 Interrupt
Pending      */

#define CAN_MSG2VAL_REG_OFFSET   0x164                /* CAN Message 2 Valid
*/

#define CAN_NWDA1_REG_OFFSET   0x120                  /* CAN New Data 1
*/

#define CAN_NWDA2_REG_OFFSET   0x124                  /* CAN New Data 2
*/

#define CAN_STS_REG_OFFSET   0x004                   /* CAN Status
*/
```

```
#define CAN_TST_REG_OFFSET   0x014                  /* CAN Test
*/

#define CAN_TXRQ1_REG_OFFSET   0x100                /* CAN Transmission Request
1            */

#define CAN_TXRQ2_REG_OFFSET   0x104                /* CAN Transmission Request
2            */

#define SYSCTL_RCGC0_REG   (*((volatile unsigned long*)0x400FE100))        /* Note:
CAN Bits from bit 24 */
```

Drivers/STD_and_Math/common_macros.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

#define **SET_BIT**(REG, BIT) ((REG) |=(1<<(BIT)))
#define **CLEAR_BIT**(REG, BIT) ((REG)&=(~(1<< (BIT))))
#define **TOGGLE_BIT**(REG, BIT) (REG^=(1<<BIT))
#define **ROR**(REG, num) (REG= (REG>>num) | (REG<<(8-num)))
#define **ROL**(REG, num) (REG= (REG<<num) | (REG>>(8-num)))
#define **BIT_IS_SET**(REG, BIT) ((REG) & (1<<(BIT)))
#define **BIT_IS_CLEAR**(REG, BIT) (!(REG & (1<<BIT)))

Macro Definition Documentation

#define BIT_IS_CLEAR(REG, BIT) (!(REG & (1<<BIT)))

#define BIT_IS_SET(REG, BIT) ((REG) & (1<<(BIT)))

#define CLEAR_BIT(REG, BIT) ((REG)&=(~(1<< (BIT))))

#define ROL(REG, num) (REG= (REG<<num) | (REG>>(8-num)))

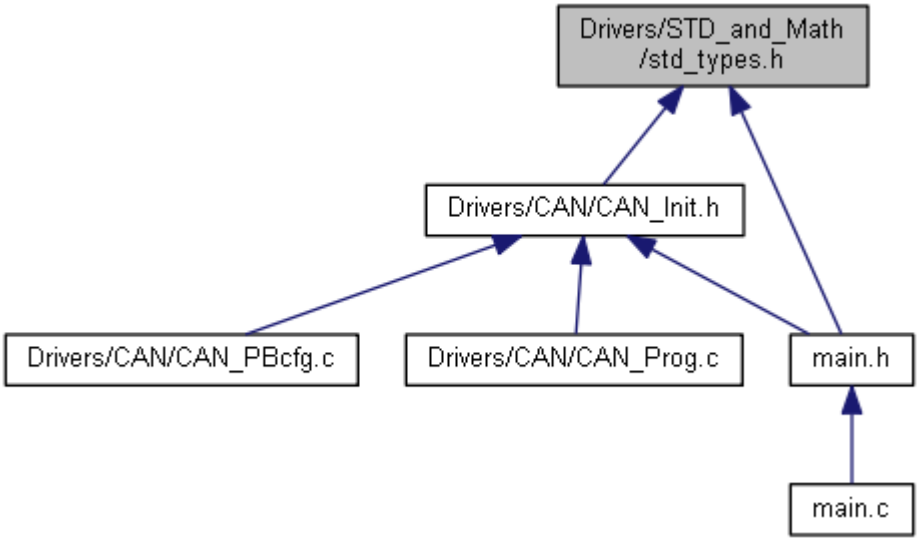
#define ROR(REG, num) (REG= (REG>>num) | (REG<<(8-num)))

#define SET_BIT(REG, BIT) ((REG) |=(1<<(BIT)))

#define TOGGLE_BIT(REG, BIT) (REG^=(1<<BIT))

Drivers/STD_and_Math/std_types.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

#define **FALSE** (0u)
#define **TRUE** (1u)
#define **HIGH** (1u)
#define **LOW** (0u)
#define **NULL_PTR** ((void*)0)
#define **RET_OK** ((**uint8**) 0x0U)
#define **RET_NOT_OK** ((**uint8**) 0x1U)

Typedefs

typedef unsigned char **bool**
typedef unsigned char **uint8**
typedef signed char **sint8**
typedef unsigned short **uint16**
typedef signed short **sint16**
typedef unsigned long **uint32**
typedef signed long **sint32**
typedef unsigned long long **uint64**
typedef signed long long **sint64**
typedef float **float32**
typedef double **float64**

Macro Definition Documentation

#define FALSE (0u)

#define HIGH (1u)

#define LOW (0u)

#define NULL_PTR ((void*)0)

#define RET_NOT_OK ((uint8) 0x1U)

#define RET_OK ((uint8) 0x0U)

#define TRUE (1u)

Typedef Documentation

typedef unsigned char bool

typedef float float32

typedef double float64

typedef signed short sint16

typedef signed long sint32

typedef signed long long sint64

typedef signed char sint8

typedef unsigned short uint16

typedef unsigned long uint32

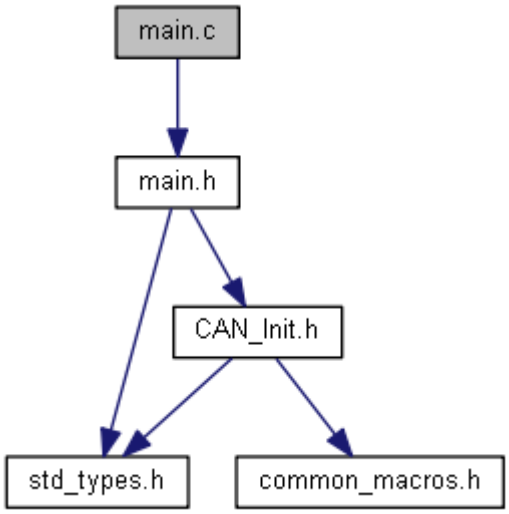
typedef unsigned long long uint64

typedef unsigned char uint8

main.c File Reference

```
#include "main.h"
```

Include dependency graph for main.c:



Functions

static void **Can0_PinsConfigurations** (void)
int **main** (void)

Variables

volatile **uint32 delay**
Can_MessageObjectConfig CanMessageObjectConfigurationSend
uint8 ReceviedData [8] = {0}
Can_MessageObjectConfig CanMessageObjectConfigurationReceive

Function Documentation

static void Can0_PinsConfigurations (void) `[static]`

int main (void)

Variable Documentation

Can_MessageObjectConfig CanMessageObjectConfigurationReceive

Initial value:= {

```
                                .MessageObjectType    = MessageObjectTypeReceive,
                                .MessageObjectId      = 2,
                                .pMsgData             = (uint8*)&ReceviedData,
```

}

Can_MessageObjectConfig CanMessageObjectConfigurationSend

Initial value:= {

```
                                    .MessageObjectType       = MessageObjectTypeTransmit,
                                    .MessageObjectSettings =
(Can_MessageObjectSettings)(CanSettingsUseIdFilter | CanSettingsUseDirectionFilter),
                                    .MessageId               = 0x01,
                                    .MessageIdMask             = 0x01,
                                    .pMsgData                = {"Hello"},
                                    .DataLengthCode           = 6,
                                    .MessageObjectId          = 1,
```

}

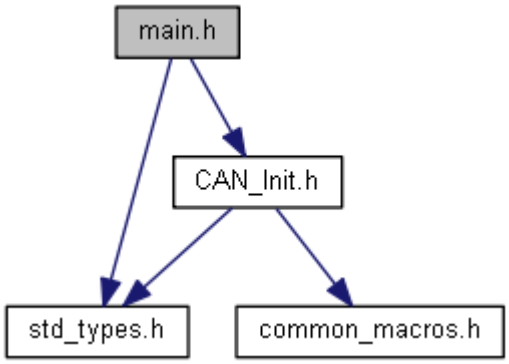
volatile uint32 delay

uint8 ReceviedData[8] = {0}

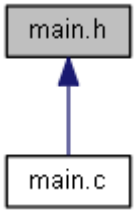
main.h File Reference

```
#include "std_types.h"
#include "CAN_Init.h"
```

Include dependency graph for main.h:



This graph shows which files directly or indirectly include this file:



Macros

#define **GPIO_PORTA_DATA_REG** (*((volatile unsigned long *)0x400043FC))
#define **GPIO_PORTA_DIR_REG** (*((volatile unsigned long *)0x40004400))
#define **GPIO_PORTA_AFSEL_REG** (*((volatile unsigned long *)0x40004420))
#define **GPIO_PORTA_PUR_REG** (*((volatile unsigned long *)0x40004510))
#define **GPIO_PORTA_PDR_REG** (*((volatile unsigned long *)0x40004514))
#define **GPIO_PORTA_DEN_REG** (*((volatile unsigned long *)0x4000451C))
#define **GPIO_PORTA_LOCK_REG** (*((volatile unsigned long *)0x40004520))
#define **GPIO_PORTA_CR_REG** (*((volatile unsigned long *)0x40004524))
#define **GPIO_PORTA_AMSEL_REG** (*((volatile unsigned long *)0x40004528))
#define **GPIO_PORTA_PCTL_REG** (*((volatile unsigned long *)0x4000452C))
#define **GPIO_PORTA_IS_REG** (*((volatile unsigned long *)0x40004404))
#define **GPIO_PORTA_IBE_REG** (*((volatile unsigned long *)0x40004408))
#define **GPIO_PORTA_IEV_REG** (*((volatile unsigned long *)0x4000440C))
#define **GPIO_PORTA_IM_REG** (*((volatile unsigned long *)0x40004410))
#define **GPIO_PORTA_RIS_REG** (*((volatile unsigned long *)0x40004414))
#define **GPIO_PORTA_ICR_REG** (*((volatile unsigned long *)0x4000441C))
#define **GPIO_PORTB_DATA_REG** (*((volatile unsigned long *)0x400053FC))
#define **GPIO_PORTB_DIR_REG** (*((volatile unsigned long *)0x40005400))
#define **GPIO_PORTB_AFSEL_REG** (*((volatile unsigned long *)0x40005420))
#define **GPIO_PORTB_PUR_REG** (*((volatile unsigned long *)0x40005510))
#define **GPIO_PORTB_PDR_REG** (*((volatile unsigned long *)0x40005514))
#define **GPIO_PORTB_DEN_REG** (*((volatile unsigned long *)0x4000551C))
#define **GPIO_PORTB_LOCK_REG** (*((volatile unsigned long *)0x40005520))
#define **GPIO_PORTB_CR_REG** (*((volatile unsigned long *)0x40005524))
#define **GPIO_PORTB_AMSEL_REG** (*((volatile unsigned long *)0x40005528))
#define **GPIO_PORTB_PCTL_REG** (*((volatile unsigned long *)0x4000552C))
#define **GPIO_PORTB_IS_REG** (*((volatile unsigned long *)0x40005404))
#define **GPIO_PORTB_IBE_REG** (*((volatile unsigned long *)0x40005408))

#define **GPIO_PORTB_IEV_REG** (*((volatile unsigned long *)0x4000540C))
#define **GPIO_PORTB_IM_REG** (*((volatile unsigned long *)0x40005410))
#define **GPIO_PORTB_RIS_REG** (*((volatile unsigned long *)0x40005414))
#define **GPIO_PORTB_ICR_REG** (*((volatile unsigned long *)0x4000541C))
#define **GPIO_PORTF_DATA_REG** (*((volatile unsigned long *)0x400253FC))
#define **GPIO_PORTF_DIR_REG** (*((volatile unsigned long *)0x40025400))
#define **GPIO_PORTF_AFSEL_REG** (*((volatile unsigned long *)0x40025420))
#define **GPIO_PORTF_PUR_REG** (*((volatile unsigned long *)0x40025510))
#define **GPIO_PORTF_PDR_REG** (*((volatile unsigned long *)0x40025514))
#define **GPIO_PORTF_DEN_REG** (*((volatile unsigned long *)0x4002551C))
#define **GPIO_PORTF_LOCK_REG** (*((volatile unsigned long *)0x40025520))
#define **GPIO_PORTF_CR_REG** (*((volatile unsigned long *)0x40025524))
#define **GPIO_PORTF_AMSEL_REG** (*((volatile unsigned long *)0x40025528))
#define **GPIO_PORTF_PCTL_REG** (*((volatile unsigned long *)0x4002552C))
#define **GPIO_PORTF_IS_REG** (*((volatile unsigned long *)0x40025404))
#define **GPIO_PORTF_IBE_REG** (*((volatile unsigned long *)0x40025408))
#define **GPIO_PORTF_IEV_REG** (*((volatile unsigned long *)0x4002540C))
#define **GPIO_PORTF_IM_REG** (*((volatile unsigned long *)0x40025410))
#define **GPIO_PORTF_RIS_REG** (*((volatile unsigned long *)0x40025414))
#define **GPIO_PORTF_ICR_REG** (*((volatile unsigned long *)0x4002541C))
#define **SYSCTL_REGCGC2_REG** (*((volatile unsigned long *)0x400FE108))
#define **SYSCTL_REGCGC1_REG** (*((volatile unsigned long *)0x400FE104))
#define **NVIC_EN0_REG** (*((volatile unsigned long *)0xE000E100))
#define **NVIC_EN1_REG** (*((volatile unsigned long *)0xE000E104))
#define **NVIC_PEND0_REG** (*((volatile unsigned long *)0xE000E200))
#define **NVIC_PEND1_REG** (*((volatile unsigned long *)0xE000E204))
#define **Enable_Interrupts**() __asm(" CPSIE I")
#define **Disable_Interrupts**() __asm(" CPSID I")
#define **Wait_For_Interrupt**() __asm(" WFI")
#define **NUMBER_OF_ITERATIONS_PER_ONE_MILI_SECOND** 762
#define **GPIOB_MODULE_CLK_ENABLE** (**uint8**)0x01)
#define **GPIOB_MODULE_CLK_POS** (**uint8**)0x01)
#define **CAN_RX_PB4_ENABLE** (**uint8**)0x01)
#define **CAN_TX_PB5_ENABLE** (**uint8**)0x01)
#define **CAN_PCTL_VAL** (**uint8**)0x08)
#define **CAN_DIGITAL_ENABLE** (**uint8**)0x01)
#define **CAN_RX_PB4_POS** (**uint8**)0x04)
#define **CAN_TX_PB5_POS** (**uint8**)0x05)
#define **CAN_RX_PCTL_POS** (**uint8**)0x10)
#define **CAN_TX_PCTL_POS** (**uint8**)0x14)

Functions

void **Delay_MS** (unsigned long n)

Macro Definition Documentation

#define Disable_Interrupts() __asm(" CPSID I")

#define Enable_Interrupts() __asm(" CPSIE I")

#define GPIO_PORTA_AFSEL_REG (*((volatile unsigned long *)0x40004420))

#define GPIO_PORTA_AMSEL_REG (*((volatile unsigned long *)0x40004528))

#define GPIO_PORTA_CR_REG (*((volatile unsigned long *)0x40004524))

#define GPIO_PORTA_DATA_REG (*((volatile unsigned long *)0x400043FC))

#define GPIO_PORTA_DEN_REG (*((volatile unsigned long *)0x4000451C))

#define GPIO_PORTA_DIR_REG (*((volatile unsigned long *)0x40004400))

#define GPIO_PORTA_IBE_REG (*((volatile unsigned long *)0x40004408))

#define GPIO_PORTA_ICR_REG (*((volatile unsigned long *)0x4000441C))

#define GPIO_PORTA_IEV_REG (*((volatile unsigned long *)0x4000440C))

#define GPIO_PORTA_IM_REG (*((volatile unsigned long *)0x40004410))

#define GPIO_PORTA_IS_REG (*((volatile unsigned long *)0x40004404))

#define GPIO_PORTA_LOCK_REG (*((volatile unsigned long *)0x40004520))

#define GPIO_PORTA_PCTL_REG (*((volatile unsigned long *)0x4000452C))

#define GPIO_PORTA_PDR_REG (*((volatile unsigned long *)0x40004514))

#define GPIO_PORTA_PUR_REG (*((volatile unsigned long *)0x40004510))

#define GPIO_PORTA_RIS_REG (*((volatile unsigned long *)0x40004414))

#define GPIO_PORTB_AFSEL_REG (*((volatile unsigned long *)0x40005420))

#define GPIO_PORTB_AMSEL_REG (*((volatile unsigned long *)0x40005528))

#define GPIO_PORTB_CR_REG (*((volatile unsigned long *)0x40005524))

#define GPIO_PORTB_DATA_REG (*((volatile unsigned long *)0x400053FC))

#define GPIO_PORTB_DEN_REG (*((volatile unsigned long *)0x4000551C))

#define GPIO_PORTB_DIR_REG (*((volatile unsigned long *)0x40005400))

#define GPIO_PORTB_IBE_REG (*((volatile unsigned long *)0x40005408))

```
#define GPIO_PORTB_ICR_REG   (*((volatile unsigned long *)0x4000541C))
#define GPIO_PORTB_IEV_REG   (*((volatile unsigned long *)0x4000540C))
#define GPIO_PORTB_IM_REG   (*((volatile unsigned long *)0x40005410))
#define GPIO_PORTB_IS_REG   (*((volatile unsigned long *)0x40005404))
#define GPIO_PORTB_LOCK_REG   (*((volatile unsigned long *)0x40005520))
#define GPIO_PORTB_PCTL_REG   (*((volatile unsigned long *)0x4000552C))
#define GPIO_PORTB_PDR_REG   (*((volatile unsigned long *)0x40005514))
#define GPIO_PORTB_PUR_REG   (*((volatile unsigned long *)0x40005510))
#define GPIO_PORTB_RIS_REG   (*((volatile unsigned long *)0x40005414))
#define GPIO_PORTF_AFSEL_REG   (*((volatile unsigned long *)0x40025420))
#define GPIO_PORTF_AMSEL_REG   (*((volatile unsigned long *)0x40025528))
#define GPIO_PORTF_CR_REG   (*((volatile unsigned long *)0x40025524))
#define GPIO_PORTF_DATA_REG   (*((volatile unsigned long *)0x400253FC))
#define GPIO_PORTF_DEN_REG   (*((volatile unsigned long *)0x4002551C))
#define GPIO_PORTF_DIR_REG   (*((volatile unsigned long *)0x40025400))
#define GPIO_PORTF_IBE_REG   (*((volatile unsigned long *)0x40025408))
#define GPIO_PORTF_ICR_REG   (*((volatile unsigned long *)0x4002541C))
#define GPIO_PORTF_IEV_REG   (*((volatile unsigned long *)0x4002540C))
#define GPIO_PORTF_IM_REG   (*((volatile unsigned long *)0x40025410))
#define GPIO_PORTF_IS_REG   (*((volatile unsigned long *)0x40025404))
#define GPIO_PORTF_LOCK_REG   (*((volatile unsigned long *)0x40025520))
#define GPIO_PORTF_PCTL_REG   (*((volatile unsigned long *)0x4002552C))
#define GPIO_PORTF_PDR_REG   (*((volatile unsigned long *)0x40025514))
#define GPIO_PORTF_PUR_REG   (*((volatile unsigned long *)0x40025510))
#define GPIO_PORTF_RIS_REG   (*((volatile unsigned long *)0x40025414))
#define NUMBER_OF_ITERATIONS_PER_ONE_MILI_SECOND   762
```

#define NVIC_EN0_REG (*((volatile unsigned long *)0xE000E100))

#define NVIC_EN1_REG (*((volatile unsigned long *)0xE000E104))

#define NVIC_PEND0_REG (*((volatile unsigned long *)0xE000E200))

#define NVIC_PEND1_REG (*((volatile unsigned long *)0xE000E204))

#define SYSCTL_REGCGC1_REG (*((volatile unsigned long *)0x400FE104))

#define SYSCTL_REGCGC2_REG (*((volatile unsigned long *)0x400FE108))

#define Wait_For_Interrupt() __asm(" WFI")

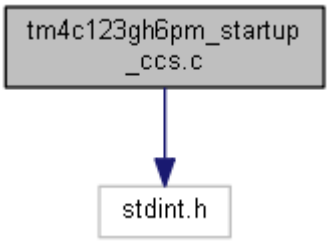
Function Documentation

void Delay_MS (unsigned long *n*)

tm4c123gh6pm_startup_ccs.c File Reference

```
#include <stdint.h>
```

Include dependency graph for tm4c123gh6pm_startup_ccs.c:



Functions

void **ResetISR** (void)
static void **NmiSR** (void)
static void **FaultISR** (void)
static void **IntDefaultHandler** (void)
void **_c_int00** (void)

Variables

uint32_t **__STACK_TOP**
void(*const **g_pfnVectors** [])(void)

Function Documentation

void _c_int00 (void)

static void FaultISR (void) [static]

static void IntDefaultHandler (void) [static]

static void NmiSR (void) [static]

void ResetISR (void)

Variable Documentation

uint32_t __STACK_TOP

void(* const g_pfnVectors[])(void)

Index

INDEX