

AC784xx_DFP CAN

7.1.0

Generated by Doxygen 1.8.13

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	2
2.1	File List	2
3	Class Documentation	3
3.1	Can_BaudrateConfigType Struct Reference	3
3.1.1	Detailed Description	3
3.1.2	Member Data Documentation	3
3.1.2.1	DataBtrate	3
3.1.2.2	NormalBtrate	4
3.1.2.3	SspOffset	4
3.2	Can_BitrateParamsType Struct Reference	4
3.2.1	Detailed Description	4
3.2.2	Member Data Documentation	4
3.2.2.1	Presc	5
3.2.2.2	Seg1	5
3.2.2.3	Seg2	5
3.2.2.4	Sjw	5
3.3	Can_DeviceType Struct Reference	5
3.3.1	Detailed Description	6
3.3.2	Member Data Documentation	6
3.3.2.1	FdEn	6
3.3.2.2	IrqCallback	6

3.3.2.3	Mode	6
3.3.2.4	RxFifoDmaChannel	7
3.3.2.5	RxFifoDmaChannelAddr	7
3.3.2.6	WakeUpIrqCallback	7
3.4	Can_DmUlqCBParams Struct Reference	7
3.4.1	Detailed Description	7
3.4.2	Member Data Documentation	8
3.4.2.1	DmUlqFlagMasks	8
3.4.2.2	Instance	8
3.5	Can_FilterParamsType Struct Reference	8
3.5.1	Detailed Description	8
3.5.2	Member Data Documentation	8
3.5.2.1	FrameType	9
3.5.2.2	ID1	9
3.5.2.3	ID2	9
3.6	Can_FiltersConfigType Struct Reference	9
3.6.1	Detailed Description	9
3.6.2	Member Data Documentation	9
3.6.2.1	FiltersNum	10
3.6.2.2	FiltersParamsPtr	10
3.7	Can_HalConfigType Struct Reference	10
3.7.1	Detailed Description	10
3.7.2	Member Data Documentation	11
3.7.2.1	BaudrateConfigPtr	11
3.7.2.2	CommonIrqEnMasks	11
3.7.2.3	DmUlqCallback	11
3.7.2.4	DmUlqEnMasks	11
3.7.2.5	EccEn	11
3.7.2.6	FdEn	12
3.7.2.7	FdIsoEn	12
3.7.2.8	Filters	12

3.7.2.9	IrqCallback	12
3.7.2.10	RxFifoBufDmaConfigNum	12
3.7.2.11	RxFifoBufDmaConfigPtr	13
3.7.2.12	RxOverWrite	13
3.7.2.13	WakeupIrqCallback	13
3.8	Can_IrqCBParams Struct Reference	13
3.8.1	Detailed Description	13
3.8.2	Member Data Documentation	14
3.8.2.1	CommonIrqFlagMasks	14
3.8.2.2	Instance	14
3.9	Can_MessageInfoType Struct Reference	14
3.9.1	Detailed Description	14
3.9.2	Member Data Documentation	15
3.9.2.1	MsgBrs	15
3.9.2.2	MsgDataLen	15
3.9.2.3	MsgDataPtr	15
3.9.2.4	MsgFdf	15
3.9.2.5	MsgId	15
3.9.2.6	MsgIdType	16
3.9.2.7	MsgRtr	16
3.9.2.8	MsgRts	16
3.10	Can_MramAddressType Struct Reference	16
3.10.1	Detailed Description	17
3.10.2	Member Data Documentation	17
3.10.2.1	endAddress	17
3.10.2.2	extFilterStartAddress	17
3.10.2.3	rxDbufferStartAddress	17
3.10.2.4	rxFifo0StartAddress	17
3.10.2.5	rxFifo1StartAddress	18
3.10.2.6	stdFilterStartAddress	18
3.10.2.7	txBufferStartAddress	18

3.10.2.8	txEventfifoStartAddress	18
3.11	Can_RxFifoBufDmaConfigType Struct Reference	18
3.11.1	Detailed Description	19
3.11.2	Member Data Documentation	19
3.11.2.1	DmaCb	19
3.11.2.2	DmaCbArgs	19
3.11.2.3	DmaDstAddr	19
3.11.2.4	Fifold	20
3.12	Can_TimeStampType Struct Reference	20
3.12.1	Detailed Description	20
3.12.2	Member Data Documentation	20
3.12.2.1	En	20
3.12.2.2	ExtClkDiv	21
3.12.2.3	ExtClkSrc	21
3.12.2.4	PosEnd	21
4	File Documentation	22
4.1	AC784xx_API_Reference_Manual_CAN.pdf File Reference	22
4.2	AC784xx_Can_Reg.h File Reference	22
4.2.1	Detailed Description	22
4.3	Can_Hal.c File Reference	22
4.3.1	Detailed Description	23
4.3.2	Macro Definition Documentation	23
4.3.2.1	CAN_WAIT_TIMEOUT	23
4.3.3	Function Documentation	23
4.3.3.1	Can_Hal_GetBase()	24
4.3.3.2	Can_Hal_GetBaseLocal()	24
4.3.3.3	Can_Hal_GetDevice()	24
4.3.3.4	Can_Hal_StartNextDma()	25
4.3.3.5	ISR() [1/8]	25
4.3.3.6	ISR() [2/8]	26
4.3.3.7	ISR() [3/8]	26

4.3.3.8	ISR() [4/8]	26
4.3.3.9	ISR() [5/8]	27
4.3.3.10	ISR() [6/8]	27
4.3.3.11	ISR() [7/8]	27
4.3.3.12	ISR() [8/8]	27
4.4	Can_Hal.h File Reference	28
4.4.1	Detailed Description	28
4.4.2	Function Documentation	29
4.4.2.1	Can_Hal_AbortTransmit()	29
4.4.2.2	Can_Hal_ConfigExtendMode()	29
4.4.2.3	Can_Hal_ConfigTimeStamp()	30
4.4.2.4	Can_Hal_Deinit()	30
4.4.2.5	Can_Hal_GetBase()	30
4.4.2.6	Can_Hal_GetControllerState()	31
4.4.2.7	Can_Hal_GetErrorsInfo()	31
4.4.2.8	Can_Hal_GetErrorState()	32
4.4.2.9	Can_Hal_GetMsgInfo()	32
4.4.2.10	Can_Hal_GetRxErrorCount()	33
4.4.2.11	Can_Hal_GetRxStatus()	33
4.4.2.12	Can_Hal_GetTxErrorCount()	34
4.4.2.13	Can_Hal_GetTxStatus()	34
4.4.2.14	Can_Hal_Init()	34
4.4.2.15	Can_Hal_ReadRxBuffer()	35
4.4.2.16	Can_Hal_SetControllerState()	35
4.4.2.17	Can_Hal_SetMsgInfo()	36
4.4.2.18	Can_Hal_StartNextDma()	36
4.4.2.19	Can_Hal_WriteTxBuffer()	37
4.5	Can_Hal_Types.h File Reference	37
4.5.1	Detailed Description	39
4.5.2	Macro Definition Documentation	39
4.5.2.1	BIT	39
4.5.3	Enumeration Type Documentation	39
4.5.3.1	Can_CtrlFrameType	39
4.5.3.2	Can_DevErrorStateType	39
4.5.3.3	Can_ErrorMaskType	40
4.5.3.4	Can_ExtendModeType	40
4.5.3.5	Can_FilterFrameType	40
4.5.3.6	Can_HalStateType	41
4.5.3.7	Can_MessageIdType	41
4.5.3.8	Can_MsgDlcType	41

Chapter 1

Class Index

1.1 Class List

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

Can_BaudrateConfigType	
CAN baudrate config args structure	3
Can_BitrateParamsType	
CAN bitrate config args structure	4
Can_DeviceType	
Can device args struct type	5
Can_DmulIrqCBParams	
CAN DMU IRQ callback args structure	7
Can_FilterParamsType	
CAN filter config args structure	8
Can_FiltersConfigType	
.	9
Can_HalConfigType	
CAN baudrate config args structure	10
Can_IrqCBParams	
CAN IRQ callback args structure	13
Can_MessageInfoType	
CAN tx/rx frame information structure	14
Can_MramAddressType	
Message RAM start address struct	16
Can_RxFifoBufDmaConfigType	
CAN baudrate config args structure	18
Can_TimeStampType	
CAN timestamp config information structure	20

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

AC784xx_API_Reference_Manual_CAN.pdf	22
AC784xx_Can_Reg.h	
This file provides extern Can Reg API	22
Can_Hal.c	
This file provides extern Can Hal API implement	22
Can_Hal.h	
This file provides extern Can Hal API implement	28
Can_Hal_Types.h	
This file provides can hal types header	37

Chapter 3

Class Documentation

3.1 Can_BaudrateConfigType Struct Reference

CAN baudrate config args structure.

```
#include <Can_Hal_Types.h>
```

Public Attributes

- [Can_BitrateParamsType NormalBitrate](#)
- [Can_BitrateParamsType DataBitrate](#)
- uint8 [SspOffset](#)

3.1.1 Detailed Description

CAN baudrate config args structure.

Definition at line 532 of file Can_Hal_Types.h.

3.1.2 Member Data Documentation

3.1.2.1 DataBitrate

```
Can\_BitrateParamsType Can_BaudrateConfigType::DataBitrate
```

Can data bitrate

Definition at line 535 of file Can_Hal_Types.h.

3.1.2.2 NormalBitrate

`Can_BitrateParamsType Can_BaudrateConfigType::NormalBitrate`

Can normal bitrate

Definition at line 534 of file Can_Hal_Types.h.

3.1.2.3 SspOffset

`uint8 Can_BaudrateConfigType::SspOffset`

Can SSPOFF:0 disable others:1~0x7F

Definition at line 536 of file Can_Hal_Types.h.

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

- [Can_Hal_Types.h](#)

3.2 Can_BitrateParamsType Struct Reference

CAN bitrate config args structure.

```
#include <Can_Hal_Types.h>
```

Public Attributes

- uint16 [Presc](#)
- uint8 [Seg1](#)
- uint8 [Seg2](#)
- uint8 [Sjw](#)

3.2.1 Detailed Description

CAN bitrate config args structure.

Definition at line 521 of file Can_Hal_Types.h.

3.2.2 Member Data Documentation

3.2.2.1 Presc

```
uint16 Can_BitrateParamsType::Presc
```

bitrate args: Presc(BRP) 0:clk/1 ... 255:clk/256

Definition at line 523 of file Can_Hal_Types.h.

3.2.2.2 Seg1

```
uint8 Can_BitrateParamsType::Seg1
```

bitrate args: Seg1 0 ~ 0xFF

Definition at line 524 of file Can_Hal_Types.h.

3.2.2.3 Seg2

```
uint8 Can_BitrateParamsType::Seg2
```

bitrate args: Seg2 0 ~ 0x7F

Definition at line 525 of file Can_Hal_Types.h.

3.2.2.4 Sjw

```
uint8 Can_BitrateParamsType::Sjw
```

bitrate args: Sjw 0 ~ 0x7F

Definition at line 526 of file Can_Hal_Types.h.

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

- [Can_Hal_Types.h](#)

3.3 Can_DeviceType Struct Reference

can device args struct type

Public Attributes

- Hal_CallbackType [IrqCallback](#)
- Hal_CallbackType [WakeuptIrqCallback](#)
- boolean [FdEn](#)
- [Can_ExtendModeType](#) [Mode](#)
- Dma_ChannelAddrType [RxFifoDmaChannelAddr](#) [CAN_RX_FIFO_NUM_MAX]
- uint8 [RxFifoDmaChannel](#) [CAN_RX_FIFO_NUM_MAX]

3.3.1 Detailed Description

can device args struct type

Definition at line 85 of file Can_Hal.c.

3.3.2 Member Data Documentation

3.3.2.1 FdEn

`boolean Can_DeviceType::FdEn`

current can cfg args, from Can_HalConfigType->FdEn

Definition at line 89 of file Can_Hal.c.

3.3.2.2 IrqCallback

`Hal_CallbackType Can_DeviceType::IrqCallback`

irq callback, from Can_HalConfigType->IrqCallback

Definition at line 87 of file Can_Hal.c.

3.3.2.3 Mode

`Can_ExtendModeType Can_DeviceType::Mode`

current can cfg args, from Can_Hal_ConfigExtendMode set

Definition at line 90 of file Can_Hal.c.

3.3.2.4 RxFifoDmaChannel

```
uint8 Can_DeviceType::RxFifoDmaChannel [CAN_RX_FIFO_NUM_MAX]
```

can rx fifo dma chanel

Definition at line 92 of file Can_Hal.c.

3.3.2.5 RxFifoDmaChannelAddr

```
Dma_ChannelAddrType Can_DeviceType::RxFifoDmaChannelAddr [CAN_RX_FIFO_NUM_MAX]
```

can rx fifo dma chanel addr

Definition at line 91 of file Can_Hal.c.

3.3.2.6 WakeupIrqCallback

```
Hal_CallbackType Can_DeviceType::WakeupIrqCallback
```

wakeup irq callback, from Can_HalConfigType->WakeupIrqCallback

Definition at line 88 of file Can_Hal.c.

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

- [Can_Hal.c](#)

3.4 Can_DmulrqCBParams Struct Reference

CAN DMU IRQ callback args structure.

```
#include <Can_Hal_Types.h>
```

Public Attributes

- uint8 [Instance](#)
- uint32 [DmulrqFlagMasks](#)

3.4.1 Detailed Description

CAN DMU IRQ callback args structure.

Definition at line 409 of file Can_Hal_Types.h.

3.4.2 Member Data Documentation

3.4.2.1 DmuIrqFlagMasks

```
uint32 Can_DmuIrqCBParams::DmuIrqFlagMasks
```

can dmU irq flag masks

Definition at line 412 of file Can_Hal_Types.h.

3.4.2.2 Instance

```
uint8 Can_DmuIrqCBParams::Instance
```

can hw unit index

Definition at line 411 of file Can_Hal_Types.h.

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

- [Can_Hal_Types.h](#)

3.5 Can_FilterParamsType Struct Reference

CAN filter config args structure.

```
#include <Can_Hal_Types.h>
```

Public Attributes

- uint32 [ID1](#)
- uint32 [ID2](#)
- [Can_FilterFrameType](#) FrameType

3.5.1 Detailed Description

CAN filter config args structure.

Definition at line 497 of file Can_Hal_Types.h.

3.5.2 Member Data Documentation

3.5.2.1 FrameType

`Can_FilterFrameType Can_FilterParamsType::FrameType`

filter accept frame type

Definition at line 501 of file Can_Hal_Types.h.

3.5.2.2 ID1

`uint32 Can_FilterParamsType::ID1`

filter ID1 or Code

Definition at line 499 of file Can_Hal_Types.h.

3.5.2.3 ID2

`uint32 Can_FilterParamsType::ID2`

filter ID2 or Mask

Definition at line 500 of file Can_Hal_Types.h.

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

- [Can_Hal_Types.h](#)

3.6 Can_FiltersConfigType Struct Reference

```
#include <Can_Hal_Types.h>
```

Public Attributes

- `uint8 FiltersNum`
- `Can_FilterParamsType * FiltersParamsPtr`

3.6.1 Detailed Description

Definition at line 508 of file Can_Hal_Types.h.

3.6.2 Member Data Documentation

3.6.2.1 FiltersNum

```
uint8 Can_FiltersConfigType::FiltersNum
```

can standard filters num

Definition at line 510 of file Can_Hal_Types.h.

3.6.2.2 FiltersParamsPtr

```
Can_FilterParamsType* Can_FiltersConfigType::FiltersParamsPtr
```

can extended filters args array ptr

Definition at line 511 of file Can_Hal_Types.h.

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

- [Can_Hal_Types.h](#)

3.7 Can_HalConfigType Struct Reference

CAN baudrate config args structure.

```
#include <Can_Hal_Types.h>
```

Public Attributes

- boolean [EccEn](#)
- boolean [FdEn](#)
- boolean [FdIsoEn](#)
- boolean [RxOverWrite](#)
- uint8 [RxFifoBufDmaConfigNum](#)
- [Can_RxFifoBufDmaConfigType](#) * [RxFifoBufDmaConfigPtr](#)
- [Can_BaudrateConfigType](#) * [BaudrateConfigPtr](#)
- [Can_FiltersConfigType](#) [Filters](#)
- Hal_CallbackType [WakeupIrqCallback](#)
- Hal_CallbackType [IrqCallback](#)
- uint32 [CommonIrqEnMasks](#)
- uint32 [DmulIrqEnMasks](#)
- Hal_CallbackType [DmulIrqCallback](#)

3.7.1 Detailed Description

CAN baudrate config args structure.

Definition at line 556 of file Can_Hal_Types.h.

3.7.2 Member Data Documentation

3.7.2.1 BaudrateConfigPtr

`Can_BaudrateConfigType* Can_HalConfigType::BaudrateConfigPtr`

Can Baudrate config args

Definition at line 565 of file Can_Hal_Types.h.

3.7.2.2 CommonIrqEnMasks

`uint32 Can_HalConfigType::CommonIrqEnMasks`

common irq enable masks, refer to CAN_COMMON_IRQ_EN_MASKS

Definition at line 583 of file Can_Hal_Types.h.

3.7.2.3 DmulrqCallback

`Hal_CallbackType Can_HalConfigType::DmuIrqCallback`

DmulrqEnMasks != 0U must dmu irq callback != NULL

Definition at line 590 of file Can_Hal_Types.h.

3.7.2.4 DmulrqEnMasks

`uint32 Can_HalConfigType::DmuIrqEnMasks`

dmu irq enable masks, refer to reg:DMUIE

Definition at line 589 of file Can_Hal_Types.h.

3.7.2.5 EccEn

`boolean Can_HalConfigType::EccEn`

Can ECC enable

Definition at line 558 of file Can_Hal_Types.h.

3.7.2.6 FdEn

```
boolean Can_HalConfigType::FdEn
```

Can FD enable

Definition at line 559 of file Can_Hal_Types.h.

3.7.2.7 FdIsoEn

```
boolean Can_HalConfigType::FdIsoEn
```

Can FD ISO mode enable

Definition at line 560 of file Can_Hal_Types.h.

3.7.2.8 Filters

```
Can_FiltersConfigType Can_HalConfigType::Filters
```

Can Filters config args

Definition at line 566 of file Can_Hal_Types.h.

3.7.2.9 IrqCallback

```
Hal_CallbackType Can_HalConfigType::IrqCallback
```

Can irq callback(include common irq masks && ecc irq masks),

Definition at line 581 of file Can_Hal_Types.h.

3.7.2.10 RxFifoBufDmaConfigNum

```
uint8 Can_HalConfigType::RxFifoBufDmaConfigNum
```

Can rx fifo buf dam config num

Definition at line 563 of file Can_Hal_Types.h.

3.7.2.11 RxFifoBufDmaConfigPtr

[Can_RxFifoBufDmaConfigType](#)* Can_HalConfigType::RxFifoBufDmaConfigPtr

Can rx fifo buf dam config args

Definition at line 564 of file Can_Hal_Types.h.

3.7.2.12 RxOverWrite

boolean Can_HalConfigType::RxOverWrite

Can rx overwrite enable, true: overwrite old frame when rx fifo overflow,

Definition at line 561 of file Can_Hal_Types.h.

3.7.2.13 WakeupIrqCallback

Hal_CallbackType Can_HalConfigType::WakeupIrqCallback

Can wakeup irq callback, NULL not enable irq

Definition at line 580 of file Can_Hal_Types.h.

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

- [Can_Hal_Types.h](#)

3.8 Can_IrqCBParams Struct Reference

CAN IRQ callback args structure.

```
#include <Can_Hal_Types.h>
```

Public Attributes

- uint8 [Instance](#)
- uint32 [CommonIrqFlagMasks](#)

3.8.1 Detailed Description

CAN IRQ callback args structure.

Definition at line 393 of file Can_Hal_Types.h.

3.8.2 Member Data Documentation

3.8.2.1 CommonIrqFlagMasks

```
uint32 Can_IrqCBParams::CommonIrqFlagMasks
```

can common irq flag masks, refer to CAN_COMMON_IRQ_FLAG_MASKS

Definition at line 396 of file Can_Hal_Types.h.

3.8.2.2 Instance

```
uint8 Can_IrqCBParams::Instance
```

can hw unit index

Definition at line 395 of file Can_Hal_Types.h.

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

- [Can_Hal_Types.h](#)

3.9 Can_MessageInfoType Struct Reference

CAN tx/rx frame information structure.

```
#include <Can_Hal_Types.h>
```

Public Attributes

- boolean [MsgFdf](#)
- boolean [MsgRtr](#)
- boolean [MsgBrs](#)
- [Can_MessageIdType](#) [MsgIdType](#)
- uint8 [MsgDataLen](#)
- uint8 * [MsgDataPtr](#)
- uint32 [MsgRts](#)
- uint32 [MsgId](#)

3.9.1 Detailed Description

CAN tx/rx frame information structure.

Definition at line 370 of file Can_Hal_Types.h.

3.9.2 Member Data Documentation

3.9.2.1 MsgBrs

```
boolean Can_MessageInfoType::MsgBrs
```

Bit rate switch(BRS)

Definition at line 374 of file Can_Hal_Types.h.

3.9.2.2 MsgDataLen

```
uint8 Can_MessageInfoType::MsgDataLen
```

frame data len

Definition at line 376 of file Can_Hal_Types.h.

3.9.2.3 MsgDataPtr

```
uint8* Can_MessageInfoType::MsgDataPtr
```

frame data ptr

Definition at line 377 of file Can_Hal_Types.h.

3.9.2.4 MsgFdf

```
boolean Can_MessageInfoType::MsgFdf
```

FD format indicator(FDF)

Definition at line 372 of file Can_Hal_Types.h.

3.9.2.5 MsgId

```
uint32 Can_MessageInfoType::MsgId
```

CAN identifier

Definition at line 379 of file Can_Hal_Types.h.

3.9.2.6 MsgIdType

[Can_MessageIdType](#) Can_MessageInfoType::MsgIdType

frame type, standard frame or extended frame

Definition at line 375 of file Can_Hal_Types.h.

3.9.2.7 MsgRtr

boolean Can_MessageInfoType::MsgRtr

Remote transmission request(RTR)

Definition at line 373 of file Can_Hal_Types.h.

3.9.2.8 MsgRts

uint32 Can_MessageInfoType::MsgRts

Receive time stamp(RTS), rx only

Definition at line 378 of file Can_Hal_Types.h.

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

- [Can_Hal_Types.h](#)

3.10 Can_MramAddressType Struct Reference

Message RAM start address struct.

```
#include <Can_Hal_Types.h>
```

Public Attributes

- uint32 [rxFifo0StartAddress](#)
- uint32 [rxFifo1StartAddress](#)
- uint32 [rxDbufferStartAddress](#)
- uint32 [txBufferStartAddress](#)
- uint32 [stdFilterStartAddress](#)
- uint32 [extFilterStartAddress](#)
- uint32 [txEventfifoStartAddress](#)
- uint32 [endAddress](#)

3.10.1 Detailed Description

Message RAM start address struct.

Definition at line 354 of file Can_Hal_Types.h.

3.10.2 Member Data Documentation

3.10.2.1 endAddress

```
uint32 Can_MramAddressType::endAddress
```

Definition at line 363 of file Can_Hal_Types.h.

3.10.2.2 extFilterStartAddress

```
uint32 Can_MramAddressType::extFilterStartAddress
```

Ext Filter Start Address

Definition at line 361 of file Can_Hal_Types.h.

3.10.2.3 rxDbufferStartAddress

```
uint32 Can_MramAddressType::rxDbufferStartAddress
```

Dedicate rx buffer start address

Definition at line 358 of file Can_Hal_Types.h.

3.10.2.4 rxFifo0StartAddress

```
uint32 Can_MramAddressType::rxFifo0StartAddress
```

Rx Fifo 0 start address

Definition at line 356 of file Can_Hal_Types.h.

3.10.2.5 rxFifo1StartAddress

```
uint32 Can_MramAddressType::rxFifo1StartAddress
```

Rx Fifo 1 start Aaddress

Definition at line 357 of file Can_Hal_Types.h.

3.10.2.6 stdFilterStartAddress

```
uint32 Can_MramAddressType::stdFilterStartAddress
```

Std Filter Start Address

Definition at line 360 of file Can_Hal_Types.h.

3.10.2.7 txBufferStartAddress

```
uint32 Can_MramAddressType::txBufferStartAddress
```

Tx Buffer Start Address

Definition at line 359 of file Can_Hal_Types.h.

3.10.2.8 txEventfifoStartAddress

```
uint32 Can_MramAddressType::txEventfifoStartAddress
```

Ex Event fifo Start Address

Definition at line 362 of file Can_Hal_Types.h.

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

- [Can_Hal_Types.h](#)

3.11 Can_RxFifoBufDmaConfigType Struct Reference

CAN baudrate config args structure.

```
#include <Can_Hal_Types.h>
```


Public Attributes

- uint8 [Fifold](#)
- uint32 [DmaDstAddr](#)
- Hal_CallbackType [DmaCb](#)
- void * [DmaCbArgs](#)

3.11.1 Detailed Description

CAN baudrate config args structure.

Definition at line 545 of file Can_Hal_Types.h.

3.11.2 Member Data Documentation

3.11.2.1 DmaCb

```
Hal_CallbackType Can_RxFifoBufDmaConfigType::DmaCb
```

Can dma callback, user can swap DmaDstAddr and start dma again

Definition at line 549 of file Can_Hal_Types.h.

3.11.2.2 DmaCbArgs

```
void* Can_RxFifoBufDmaConfigType::DmaCbArgs
```

Can dma callback args, if needed

Definition at line 550 of file Can_Hal_Types.h.

3.11.2.3 DmaDstAddr

```
uint32 Can_RxFifoBufDmaConfigType::DmaDstAddr
```

Can dma dest address, it's length must one frame info size

Definition at line 548 of file Can_Hal_Types.h.

3.11.2.4 Fifold

```
uint8 Can_RxFifoBufDmaConfigType::FifoId
```

Can rx fifo id

Definition at line 547 of file Can_Hal_Types.h.

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

- [Can_Hal_Types.h](#)

3.12 Can_TimeStampType Struct Reference

CAN timestamp config information structure.

```
#include <Can_Hal_Types.h>
```

Public Attributes

- boolean [En](#)
- boolean [ExtClkSrc](#)
- boolean [PosEnd](#)
- uint8 [ExtClkDiv](#)

3.12.1 Detailed Description

CAN timestamp config information structure.

Definition at line 418 of file Can_Hal_Types.h.

3.12.2 Member Data Documentation

3.12.2.1 En

```
boolean Can_TimeStampType::En
```

can timestamp enable, affect if TTS/RTS is valid

Definition at line 420 of file Can_Hal_Types.h.

3.12.2.2 ExtClkDiv

```
uint8 Can_TimeStampType::ExtClkDiv
```

can timestamp external clk source division, 0:clk/1 ... 7:clk/8

Definition at line 423 of file Can_Hal_Types.h.

3.12.2.3 ExtClkSrc

```
boolean Can_TimeStampType::ExtClkSrc
```

can timestamp use external clk source or not

Definition at line 421 of file Can_Hal_Types.h.

3.12.2.4 PosEnd

```
boolean Can_TimeStampType::PosEnd
```

can timestamp mark position select, false:SOF true:EOF

Definition at line 422 of file Can_Hal_Types.h.

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

- [Can_Hal_Types.h](#)

Chapter 4

File Documentation

4.1 AC784xx_API_Reference_Manual_CAN.pdf File Reference

4.2 AC784xx_Can_Reg.h File Reference

This file provides extern Can Reg API.

```
#include "Can_Hal_Types.h"
```

4.2.1 Detailed Description

This file provides extern Can Reg API.

4.3 Can_Hal.c File Reference

This file provides extern Can Hal API implement.

```
#include "OsIf.h"  
#include "OsIf_Time.h"  
#include "Ckgen_Hal.h"  
#include "Rcm_Hal.h"  
#include "Dma_Hal.h"  
#include "AC784xx_Can_Reg.h"  
#include "Can_Hal.h"  
#include "Core_Hal.h"
```

Classes

- struct [Can_DeviceType](#)
can device args struct type

Macros

- #define `CAN_WAIT_TIMEOUT` (50U) /*MS*/

Functions

- LOCAL_INLINE CAN_Type * `Can_Hal_GetBaseLocal` (uint8 Instance)
Get the can base address.
- LOCAL_INLINE Can_DeviceType * `Can_Hal_GetDevice` (uint8 Instance)
Get the can device ptr.
- void `Can_Hal_StartNextDma` (uint8 Instance, uint8 Fifold, uint32 DmaDstAddr)
Can start next dma transfer.
- CAN_Type * `Can_Hal_GetBase` (uint8 Instance)
Get the can base address.
- `ISR` (CAN0_IRQHandler)
CAN0 IRQ Handler function.
- `ISR` (CAN0_Wakeup_IRQHandler)
CAN0 wakeup IRQ Handler function.
- `ISR` (CAN1_IRQHandler)
CAN1 IRQ Handler function.
- `ISR` (CAN1_Wakeup_IRQHandler)
CAN1 wakeup IRQ Handler function.
- `ISR` (CAN2_IRQHandler)
CAN2 IRQ Handler function.
- `ISR` (CAN2_Wakeup_IRQHandler)
CAN2 wakeup IRQ Handler function.
- `ISR` (CAN3_IRQHandler)
CAN3 IRQ Handler function.
- `ISR` (CAN3_Wakeup_IRQHandler)
CAN3 wakeup IRQ Handler function.

4.3.1 Detailed Description

This file provides extern Can Hal API implement.

4.3.2 Macro Definition Documentation

4.3.2.1 CAN_WAIT_TIMEOUT

```
#define CAN_WAIT_TIMEOUT (50U) /*MS*/
```

Definition at line 54 of file Can_Hal.c.

4.3.3 Function Documentation

4.3.3.1 Can_Hal_GetBase()

```
CAN_Type* Can_Hal_GetBase (
    uint8 Instance )
```

Get the can base address.

Note

Function ID: DES_CAN_API_016
Service ID: none

Parameters

in	Instance	Specify CAN HW Unit
----	----------	---------------------

Returns

can base address ptr

Definition at line 2952 of file Can_Hal.c.

4.3.3.2 Can_Hal_GetBaseLocal()

```
LOCAL_INLINE CAN_Type * Can_Hal_GetBaseLocal (
    uint8 Instance )
```

Get the can base address.

Note

Function ID: DES_CAN_API_029
Service ID: none

Parameters

in	Instance	Specify CAN HW Unit
----	----------	---------------------

Returns

can base address ptr

Definition at line 2958 of file Can_Hal.c.

4.3.3.3 Can_Hal_GetDevice()

```
LOCAL_INLINE Can_DeviceType * Can_Hal_GetDevice (
    uint8 Instance )
```

Get the can device ptr.

Note

Function ID: DES_CAN_API_030
Service ID: none

Parameters

in	<i>Instance</i>	Specify CAN HW Unit
----	-----------------	---------------------

Returns

can device ptr

Definition at line 2971 of file Can_Hal.c.

4.3.3.4 Can_Hal_StartNextDma()

```
void Can_Hal_StartNextDma (
    uint8 Instance,
    uint8 FifoId,
    uint32 DmaDstAddr )
```

Can start next dma transfer.

Note

Function ID: DES_CAN_API_022
Service ID: none

Parameters

in	<i>Instance</i>	Specify CAN HW Unit
in	<i>Fifold</i>	Can HW Unit rx fifoid
in	<i>DmaDstAddr</i>	The dma dest addresss

Returns

void

Definition at line 2930 of file Can_Hal.c.

4.3.3.5 ISR() [1/8]

```
ISR (
    CAN0_IRQHandler )
```

CAN0 IRQ Handler function.

Returns

None

Definition at line 3126 of file Can_Hal.c.

4.3.3.6 ISR() [2/8]

```
ISR (
    CAN0_Wakeup_IRQHandler )
```

CAN0 wakeup IRQ Handler function.

Returns

None

Definition at line 3135 of file Can_Hal.c.

4.3.3.7 ISR() [3/8]

```
ISR (
    CAN1_IRQHandler )
```

CAN1 IRQ Handler function.

Returns

None

Definition at line 3144 of file Can_Hal.c.

4.3.3.8 ISR() [4/8]

```
ISR (
    CAN1_Wakeup_IRQHandler )
```

CAN1 wakeup IRQ Handler function.

Returns

None

Definition at line 3153 of file Can_Hal.c.

4.3.3.9 ISR() [5/8]

```
ISR (
    CAN2_IRQHandler )
```

CAN2 IRQ Handler function.

Returns

None

Definition at line 3162 of file Can_Hal.c.

4.3.3.10 ISR() [6/8]

```
ISR (
    CAN2_Wakeup_IRQHandler )
```

CAN2 wakeup IRQ Handler function.

Returns

None

Definition at line 3171 of file Can_Hal.c.

4.3.3.11 ISR() [7/8]

```
ISR (
    CAN3_IRQHandler )
```

CAN3 IRQ Handler function.

Returns

None

Definition at line 3180 of file Can_Hal.c.

4.3.3.12 ISR() [8/8]

```
ISR (
    CAN3_Wakeup_IRQHandler )
```

CAN3 wakeup IRQ Handler function.

Returns

None

Definition at line 3189 of file Can_Hal.c.

4.4 Can_Hal.h File Reference

This file provides extern Can Hal API implement.

```
#include "Can_Hal_Types.h"
```

Functions

- void [Can_Hal_Init](#) (uint8 Instance, const [Can_HalConfigType](#) *ConfigPtr)
Initialize a can hw unit.
- void [Can_Hal_Deinit](#) (uint8 Instance)
Deinitializes a can hw unit.
- Hal_StatusType [Can_Hal_SetControllerState](#) (uint8 Instance, [Can_HalStateType](#) State)
Set work state for a can hw unit.
- [Can_HalStateType](#) [Can_Hal_GetControllerState](#) (uint8 Instance)
Get work state for a can hw unit.
- Hal_StatusType [Can_Hal_WriteTxBuffer](#) (uint8 Instance, uint8 BufferId, const [Can_MessageInfoType](#) *MessagePtr)
Write send frame information to hw and send it.
- Hal_StatusType [Can_Hal_ReadRxBuffer](#) (uint8 Instance, uint8 BufferId, [Can_MessageInfoType](#) *MessagePtr)
Read receive frame information from hw.
- Hal_StatusType [Can_Hal_GetTxStatus](#) (uint8 Instance, uint8 BufferId)
Get sent status.
- Hal_StatusType [Can_Hal_GetRxStatus](#) (uint8 Instance, uint8 BufferId)
Get received status.
- uint8 [Can_Hal_GetTxErrorCount](#) (uint8 Instance)
Get the number of frames the can sent errors.
- uint8 [Can_Hal_GetRxErrorCount](#) (uint8 Instance)
Get the number of frames the can received errors.
- [Can_DevErrorStateType](#) [Can_Hal_GetErrorState](#) (uint8 Instance)
Get can error state.
- uint32 [Can_Hal_GetErrorsInfo](#) (uint8 Instance)
Get can errors information.
- void [Can_Hal_ConfigExtendMode](#) (uint8 Instance, [Can_ExtendModeType](#) Mode)
Set can timestamp.
- void [Can_Hal_ConfigTimeStamp](#) (uint8 Instance, const [Can_TimeStampType](#) *tsConfigPtr)
Config can timestamp function.
- CAN_Type * [Can_Hal_GetBase](#) (uint8 Instance)
Get the can base address.
- Hal_StatusType [Can_Hal_AbortTransmit](#) (uint8 Instance, uint8 BufferId)
abort tx transmit
- void [Can_Hal_SetMsgInfo](#) (uint8 Instance, uint8 BufferIndex, const [Can_MessageInfoType](#) *InfoPtr)
config send frame info to hw.
- void [Can_Hal_GetMsgInfo](#) ([Can_MessageInfoType](#) *InfoPtr, const uint32 *Buf)
Get the CAN message information from reveive buffer.
- void [Can_Hal_StartNextDma](#) (uint8 Instance, uint8 Fifold, uint32 DmaDstAddr)
Can start next dma transfer.

4.4.1 Detailed Description

This file provides extern Can Hal API implement.

4.4.2 Function Documentation

4.4.2.1 Can_Hal_AbortTransmit()

```
Hal_StatusType Can_Hal_AbortTransmit (
    uint8 Instance,
    uint8 BufferId )
```

abort tx transmit

Note

Function ID: DES_CAN_API_017
Service ID: none

Parameters

in	<i>Instance</i>	Specify CAN HW Unit
in	<i>BufferId</i>	Read buffer id

Returns

Can_HalStateType. STATUS_SUCCESS: abort success, STATUS_ERROR: can not abort(transmit completed)

4.4.2.2 Can_Hal_ConfigExtendMode()

```
void Can_Hal_ConfigExtendMode (
    uint8 Instance,
    Can_ExtendModeType Mode )
```

Set can timestamp.

Note

Function ID: DES_CAN_API_013
Service ID: none

Parameters

in	<i>Instance</i>	Specify CAN HW Unit
in	<i>Mode</i>	Can_ExtendModeType

Returns

void

4.4.2.3 Can_Hal_ConfigTimeStamp()

```
void Can_Hal_ConfigTimeStamp (
    uint8 Instance,
    const Can_TimeStampType * tsConfigPtr )
```

Config can timestamp function.

Note

Function ID: DES_CAN_API_014
Service ID: none

Parameters

in	<i>Instance</i>	Specify CAN HW Unit
in	<i>tsConfigPtr</i>	timestamp config args

Returns

void

4.4.2.4 Can_Hal_Deinit()

```
void Can_Hal_Deinit (
    uint8 Instance )
```

Deinitializes a can hw unit.

Note

Function ID: DES_CAN_API_002
Service ID: none

Parameters

in	<i>Instance</i>	Specify CAN HW Unit
----	-----------------	---------------------

Returns

void

4.4.2.5 Can_Hal_GetBase()

```
CAN_Type* Can_Hal_GetBase (
    uint8 Instance )
```

Get the can base address.

Note

Function ID: DES_CAN_API_016
Service ID: none

Parameters

in	<i>Instance</i>	Specify CAN HW Unit
----	-----------------	---------------------

Returns

can base address ptr

Definition at line 2952 of file Can_Hal.c.

4.4.2.6 Can_Hal_GetControllerState()

```
Can_HalStateType Can_Hal_GetControllerState (  
    uint8 Instance )
```

Get work state for a can hw unit.

Note

Function ID: DES_CAN_API_004
Service ID: none

Parameters

in	<i>Instance</i>	Specify CAN HW Unit
----	-----------------	---------------------

Returns

Can_HalStateType. return The can currenrt work state

4.4.2.7 Can_Hal_GetErrorsInfo()

```
uint32 Can_Hal_GetErrorsInfo (  
    uint8 Instance )
```

Get can errors information.

Note

Function ID: DES_CAN_API_012
Service ID: none

Parameters

in	<i>Instance</i>	Specify CAN HW Unit
----	-----------------	---------------------

Returns

can errors mask information

4.4.2.8 Can_Hal_GetErrorState()

```
Can_DevErrorStateType Can_Hal_GetErrorState (
    uint8 Instance )
```

Get can error state.

Note

Function ID: DES_CAN_API_011

Service ID: none

Parameters

in	<i>Instance</i>	Specify CAN HW Unit
----	-----------------	---------------------

Returns

Can_DevErrorStateType. can error state

4.4.2.9 Can_Hal_GetMsgInfo()

```
void Can_Hal_GetMsgInfo (
    Can_MessageInfoType * InfoPtr,
    const uint32 * Buf )
```

Get the CAN message information from reveive buffer.

Note

Function ID: DES_CAN_API_021

Service ID: none

Parameters

out	<i>InfoPtr</i>	pointer to message information
in	<i>Buf</i>	pointer to receive buffer

Returns

void

4.4.2.10 Can_Hal_GetRxErrorCount()

```
uint8 Can_Hal_GetRxErrorCount (
    uint8 Instance )
```

Get the number of frames the can received errors.

Note

Function ID: DES_CAN_API_010
Service ID: none

Parameters

in	<i>Instance</i>	Specify CAN HW Unit
----	-----------------	---------------------

Returns

the number of frames the can received errors

4.4.2.11 Can_Hal_GetRxStatus()

```
Hal_StatusType Can_Hal_GetRxStatus (
    uint8 Instance,
    uint8 BufferId )
```

Get received status.

Note

Function ID: DES_CAN_API_008
Service ID: none

Parameters

in	<i>Instance</i>	Specify CAN HW Unit
in	<i>BufferId</i>	Read buffer id

Returns

Can_HalStateType. if received successful return STATUS_SUCCESS or return STATUS_ERROR

4.4.2.12 Can_Hal_GetTxErrorCount()

```
uint8 Can_Hal_GetTxErrorCount (
    uint8 Instance )
```

Get the number of frames the can sent errors.

Note

Function ID: DES_CAN_API_009
Service ID: none

Parameters

in	<i>Instance</i>	Specify CAN HW Unit
----	-----------------	---------------------

Returns

the number of frames the can sent errors

4.4.2.13 Can_Hal_GetTxStatus()

```
Hal_StatusType Can_Hal_GetTxStatus (
    uint8 Instance,
    uint8 BufferId )
```

Get sent status.

Note

Function ID: DES_CAN_API_007
Service ID: none

Parameters

in	<i>Instance</i>	Specify CAN HW Unit
in	<i>BufferId</i>	Read buffer id

Returns

Can_HalStateType. if sent successful return STATUS_SUCCESS or return STATUS_ERROR

4.4.2.14 Can_Hal_Init()

```
void Can_Hal_Init (
    uint8 Instance,
    const Can_HalConfigType * ConfigPtr )
```

Initialize a can hw unit.

Note

Function ID: DES_CAN_API_001
Service ID: none

Parameters

in	<i>Instance</i>	Specify CAN HW Unit
in	<i>ConfigPtr</i>	Config args

Returns

void

4.4.2.15 Can_Hal_ReadRxBuffer()

```
Hal_StatusType Can_Hal_ReadRxBuffer (
    uint8 Instance,
    uint8 BufferId,
    Can_MessageInfoType * MessagePtr )
```

Read receive frame information from hw.

Note

Function ID: DES_CAN_API_006
Service ID: none

Parameters

in	<i>Instance</i>	Specify CAN HW Unit
in	<i>BufferId</i>	Read buffer id
out	<i>MessagePtr</i>	Read can frame information

Returns

Can_HalStateType. if successful return STATUS_SUCCESS or return STATUS_ERROR

4.4.2.16 Can_Hal_SetControllerState()

```
Hal_StatusType Can_Hal_SetControllerState (
    uint8 Instance,
    Can_HalStateType State )
```

Set work state for a can hw unit.

Note

Function ID: DES_CAN_API_003
Service ID: none

Parameters

in	<i>Instance</i>	Specify CAN HW Unit
in	<i>State</i>	The can work state to be set

Returns

Hal_StatusType. if successful return STATUS_SUCCESS or return STATUS_ERROR

4.4.2.17 Can_Hal_SetMsgInfo()

```
void Can_Hal_SetMsgInfo (
    uint8 Instance,
    uint8 BufferIndex,
    const Can_MessageInfoType * InfoPtr )
```

config send frame info to hw.

Note

Function ID: DES_CAN_API_019

Service ID: none

Parameters

in	<i>Instance</i>	CAN module Instance
in	<i>BufferIndex</i>	rx buffer index
in	<i>InfoPtr</i>	- a non-null pointer pointing to the frame information to be transmitted

Returns

void

4.4.2.18 Can_Hal_StartNextDma()

```
void Can_Hal_StartNextDma (
    uint8 Instance,
    uint8 FifoId,
    uint32 DmaDstAddr )
```

Can start next dma transfer.

Note

Function ID: DES_CAN_API_022

Service ID: none

Parameters

in	<i>Instance</i>	Specify CAN HW Unit
in	<i>Fifold</i>	Can HW Unit rx fifoid
in	<i>DmaDstAddr</i>	The dma dest addresss

Returns

void

Definition at line 2930 of file Can_Hal.c.

4.4.2.19 Can_Hal_WriteTxBuffer()

```
Hal_StatusType Can_Hal_WriteTxBuffer (
    uint8 Instance,
    uint8 BufferId,
    const Can_MessageInfoType * MessagePtr )
```

Write send frame information to hw and send it.

Note

Function ID: DES_CAN_API_005

Service ID: none

Parameters

in	<i>Instance</i>	Specify CAN HW Unit
in	<i>BufferId</i>	Send buffer id
in	<i>MessagePtr</i>	Send can frame information

Returns

Can_HalStateType. if successful return STATUS_SUCCESS or return STATUS_BUSY

4.5 Can_Hal_Types.h File Reference

This file provides can hal types header.

#include "Device_Register.h"

Classes

- struct [Can_MramAddressType](#)
Message RAM start address struct.

- struct [Can_MessageInfoType](#)
CAN tx/rx frame information structure.
- struct [Can_IrqCBParams](#)
CAN IRQ callback args structure.
- struct [Can_DmulrqCBParams](#)
CAN DMU IRQ callback args structure.
- struct [Can_TimeStampType](#)
CAN timestamp config information structure.
- struct [Can_FilterParamsType](#)
CAN filter config args structure.
- struct [Can_FiltersConfigType](#)
- struct [Can_BitrateParamsType](#)
CAN bitrate config args structure.
- struct [Can_BaudrateConfigType](#)
CAN baudrate config args structure.
- struct [Can_RxFifoBufDmaConfigType](#)
CAN baudrate config args structure.
- struct [Can_HalConfigType](#)
CAN baudrate config args structure.

Macros

- #define [BIT\(x\)](#) (1UL << (uint32)(x))

Enumerations

- enum [Can_ErrorMaskType](#) {
[CAN_BUSERR_BIT_MASK](#) = 0x1, [CAN_BUSERR_FORM_MASK](#) = 0x2, [CAN_BUSERR_STUFF_MASK](#) = 0x4,
[CAN_BUSERR_ACK_MASK](#) = 0x8,
[CAN_BUSERR_CRC_MASK](#) = 0x10, [CAN_BUSERR_OTHER_MASK](#) = 0x20, [CAN_ARBIT_ERR_MASK](#) = 0x40,
[CAN_OVERFLOW_ERR_MASK](#) = 0x80,
[CAN_ECC_WARN_MASK](#) = 0x100, [CAN_ECC_ERR_MASK](#) = 0x200 }
CAN error mask type enumeration.
- enum [Can_CtrlFrameType](#) { [CAN_CTRL_DATA_FRAME](#) = 0x0U, [CAN_CTRL_REMOTE_FRAME](#) }
CAN frame RTR type type enumeration.
- enum [Can_MsgDlcType](#) {
[CAN_MSG_DLC_12_BYTES](#) = 0x09U, [CAN_MSG_DLC_16_BYTES](#), [CAN_MSG_DLC_20_BYTES](#), [CAN_MSG_DLC_24_BYTES](#),
[CAN_MSG_DLC_32_BYTES](#), [CAN_MSG_DLC_48_BYTES](#), [CAN_MSG_DLC_64_BYTES](#) }
CAN frame DLC value type enumeration.
- enum [Can_ExtendModeType](#) { [CAN_EXTMODE_OFF](#) = 0x0U, [CAN_EXTMODE_LISTENING](#), [CAN_EXTMODE_LOOPBACK_INTERNAL](#), [CAN_EXTMODE_LOOPBACK_EXTERNAL](#) }
CAN extend mode type enumeration.
- enum [Can_DevErrorStateType](#) { [CAN_ERR_STATE_ACTIVE](#) = 0x0U, [CAN_ERR_STATE_PASSIVE](#), [CAN_ERR_STATE_BUSOFF](#) }
CAN error state type enumeration.
- enum [Can_HalStateType](#) { [CAN_STATE_STOP](#) = 0x0U, [CAN_STATE_RUNNING](#), [CAN_STATE_STANDBY](#) }
CAN work state type enumeration.
- enum [Can_MessageIdType](#) { [CAN_STANDARD_MSG](#) = 0x0U, [CAN_EXTENDED_MSG](#) }
CAN frame type enumeration.
- enum [Can_FilterFrameType](#) { [STD_FRAME](#) = 0U, [EXT_FRAME](#) }
CAN filter type enumeration.

4.5.1 Detailed Description

This file provides can hal types header.

4.5.2 Macro Definition Documentation

4.5.2.1 BIT

```
#define BIT(  
    x ) (1UL << (uint32)(x))
```

Definition at line 51 of file Can_Hal_Types.h.

4.5.3 Enumeration Type Documentation

4.5.3.1 Can_CtrlFrameType

```
enum Can_CtrlFrameType
```

CAN frame RTR type type enumeration.

Enumerator

CAN_CTRL_DATA_FRAME	can frame RTR: data frame
CAN_CTRL_REMOTE_FRAME	can frame RTR: remote frame

Definition at line 133 of file Can_Hal_Types.h.

4.5.3.2 Can_DevErrorStateType

```
enum Can_DevErrorStateType
```

CAN error state type enumeration.

Enumerator

CAN_ERR_STATE_ACTIVE	error state active
CAN_ERR_STATE_PASSIVE	error state passive
CAN_ERR_STATE_BUSOFF	error state busoff

Definition at line 167 of file Can_Hal_Types.h.

4.5.3.3 Can_ErrorMaskType

enum [Can_ErrorMaskType](#)

CAN error mask type enumeration.

Enumerator

CAN_BUSERR_BIT_MASK	can bus error: bit error
CAN_BUSERR_FORM_MASK	can bus error: form error
CAN_BUSERR_STUFF_MASK	can bus error: stuff error
CAN_BUSERR_ACK_MASK	can bus error: ack error
CAN_BUSERR_CRC_MASK	can bus error: crc error
CAN_BUSERR_OTHER_MASK	can bus error: others error
CAN_ARBIT_ERR_MASK	can arbitrate error
CAN_OVERFLOW_ERR_MASK	can receive fifo overflow error
CAN_ECC_WARN_MASK	can ecc warning, refer to CAN_VERMEM_MDWIF_Msk
CAN_ECC_ERR_MASK	can ecc error, refer to CAN_VERMEM_MDEIF_Msk

Definition at line 116 of file Can_Hal_Types.h.

4.5.3.4 Can_ExtendModeType

enum [Can_ExtendModeType](#)

CAN extend mode type enumeration.

Enumerator

CAN_EXTMODE_OFF	can extend mode off
CAN_EXTMODE_LISTENING	can extend mode listening
CAN_EXTMODE_LOOPBACK_INTERNAL	can extend mode internal
CAN_EXTMODE_LOOPBACK_EXTERNAL	can extend mode external

Definition at line 156 of file Can_Hal_Types.h.

4.5.3.5 Can_FilterFrameType

enum [Can_FilterFrameType](#)

CAN filter type enumeration.

Enumerator

STD_FRAME	Only receive standard frame id
EXT_FRAME	Only receive extend frame id

Definition at line 196 of file Can_Hal_Types.h.

4.5.3.6 Can_HalStateType

```
enum Can_HalStateType
```

CAN work state type enumeration.

Enumerator

CAN_STATE_STOP	state stop
CAN_STATE_RUNNING	state running
CAN_STATE_STANDBY	state standby

Definition at line 177 of file Can_Hal_Types.h.

4.5.3.7 Can_MessageIdType

```
enum Can_MessageIdType
```

CAN frame type enumeration.

Enumerator

CAN_STANDARD_MSG	standard frame
CAN_EXTENDED_MSG	extend frame

Definition at line 187 of file Can_Hal_Types.h.

4.5.3.8 Can_MsgDlcType

```
enum Can_MsgDlcType
```

CAN frame DLC value type enumeration.

Enumerator

CAN_MSG_DLC_12_BYTES	can frame DLC value, valid data len 12 bytes
CAN_MSG_DLC_16_BYTES	can frame DLC value, valid data len 16 bytes

Enumerator

CAN_MSG_DLC_20_BYTES	can frame DLC value, valid data len 20 bytes
CAN_MSG_DLC_24_BYTES	can frame DLC value, valid data len 24 bytes
CAN_MSG_DLC_32_BYTES	can frame DLC value, valid data len 32 bytes
CAN_MSG_DLC_48_BYTES	can frame DLC value, valid data len 48 bytes
CAN_MSG_DLC_64_BYTES	can frame DLC value, valid data len 64 bytes

Definition at line 142 of file Can_Hal_Types.h.

Index

AC784xx_API_Reference_Manual_CAN.pdf, [22](#)

AC784xx_Can_Reg.h, [22](#)

BIT

Can_Hal_Types.h, [39](#)

BaudrateConfigPtr

Can_HalConfigType, [11](#)

CAN_WAIT_TIMEOUT

Can_Hal.c, [23](#)

Can_BaudrateConfigType, [3](#)

DataBitrate, [3](#)

NormalBitrate, [3](#)

SspOffset, [4](#)

Can_BitrateParamsType, [4](#)

Presc, [4](#)

Seg1, [5](#)

Seg2, [5](#)

Sjw, [5](#)

Can_CtrlFrameType

Can_Hal_Types.h, [39](#)

Can_DevErrorStateType

Can_Hal_Types.h, [39](#)

Can_DeviceType, [5](#)

FdEn, [6](#)

IrqCallback, [6](#)

Mode, [6](#)

RxFifoDmaChannel, [6](#)

RxFifoDmaChannelAddr, [7](#)

WakeUpIrqCallback, [7](#)

Can_DmulIrqCBParams, [7](#)

DmulIrqFlagMasks, [8](#)

Instance, [8](#)

Can_ErrorMaskType

Can_Hal_Types.h, [40](#)

Can_ExtendModeType

Can_Hal_Types.h, [40](#)

Can_FilterFrameType

Can_Hal_Types.h, [40](#)

Can_FilterParamsType, [8](#)

FrameType, [8](#)

ID1, [9](#)

ID2, [9](#)

Can_FiltersConfigType, [9](#)

FiltersNum, [9](#)

FiltersParamsPtr, [10](#)

Can_Hal.c, [22](#)

CAN_WAIT_TIMEOUT, [23](#)

Can_Hal_GetBase, [23](#)

Can_Hal_GetBaseLocal, [24](#)

Can_Hal_GetDevice, [24](#)

Can_Hal_StartNextDma, [25](#)

ISR, [25–27](#)

Can_Hal.h, [28](#)

Can_Hal_AbortTransmit, [29](#)

Can_Hal_ConfigExtendMode, [29](#)

Can_Hal_ConfigTimeStamp, [29](#)

Can_Hal_Deinit, [30](#)

Can_Hal_GetBase, [30](#)

Can_Hal_GetControllerState, [31](#)

Can_Hal_GetErrorState, [32](#)

Can_Hal_GetErrorsInfo, [31](#)

Can_Hal_GetMsgInfo, [32](#)

Can_Hal_GetRxErrorCount, [33](#)

Can_Hal_GetRxStatus, [33](#)

Can_Hal_GetTxErrorCount, [33](#)

Can_Hal_GetTxStatus, [34](#)

Can_Hal_Init, [34](#)

Can_Hal_ReadRxBuffer, [35](#)

Can_Hal_SetControllerState, [35](#)

Can_Hal_SetMsgInfo, [36](#)

Can_Hal_StartNextDma, [36](#)

Can_Hal_WriteTxBuffer, [37](#)

Can_Hal_AbortTransmit

Can_Hal.h, [29](#)

Can_Hal_ConfigExtendMode

Can_Hal.h, [29](#)

Can_Hal_ConfigTimeStamp

Can_Hal.h, [29](#)

Can_Hal_Deinit

Can_Hal.h, [30](#)

Can_Hal_GetBase

Can_Hal.c, [23](#)

Can_Hal.h, [30](#)

Can_Hal_GetBaseLocal

Can_Hal.c, [24](#)

Can_Hal_GetControllerState

Can_Hal.h, [31](#)

Can_Hal_GetDevice

Can_Hal.c, [24](#)

Can_Hal_GetErrorState

Can_Hal.h, [32](#)

Can_Hal_GetErrorsInfo

Can_Hal.h, [31](#)

Can_Hal_GetMsgInfo

Can_Hal.h, [32](#)

Can_Hal_GetRxErrorCount

Can_Hal.h, [33](#)

Can_Hal_GetRxStatus

Can_Hal.h, [33](#)

Can_Hal_GetTxErrorCount

Can_Hal.h, [33](#)

Can_Hal_GetTxStatus

Can_Hal.h, [34](#)

Can_Hal_Init	txEventfifoStartAddress, 18
Can_Hal.h, 34	Can_MsgDlcType
Can_Hal_ReadRxBuffer	Can_Hal_Types.h, 41
Can_Hal.h, 35	Can_RxFifoBufDmaConfigType, 18
Can_Hal_SetControllerState	DmaCb, 19
Can_Hal.h, 35	DmaCbArgs, 19
Can_Hal_SetMsgInfo	DmaDstAddr, 19
Can_Hal.h, 36	Fifold, 19
Can_Hal_StartNextDma	Can_TimeStampType, 20
Can_Hal.c, 25	En, 20
Can_Hal.h, 36	ExtClkDiv, 20
Can_Hal_Types.h, 37	ExtClkSrc, 21
BIT, 39	PosEnd, 21
Can_CtrlFrameType, 39	CommonIrqEnMasks
Can_DevErrorStateType, 39	Can_HalConfigType, 11
Can_ErrorMaskType, 40	CommonIrqFlagMasks
Can_ExtendModeType, 40	Can_IrqCBParams, 14
Can_FilterFrameType, 40	
Can_HalStateType, 41	DataBtrate
Can_MessageIdType, 41	Can_BaudrateConfigType, 3
Can_MsgDlcType, 41	DmaCb
Can_Hal_WriteTxBuffer	Can_RxFifoBufDmaConfigType, 19
Can_Hal.h, 37	DmaCbArgs
Can_HalConfigType, 10	Can_RxFifoBufDmaConfigType, 19
BaudrateConfigPtr, 11	DmaDstAddr
CommonIrqEnMasks, 11	Can_RxFifoBufDmaConfigType, 19
DmulrqCallback, 11	DmulrqCallback
DmulrqEnMasks, 11	Can_HalConfigType, 11
EccEn, 11	DmulrqEnMasks
FdEn, 11	Can_HalConfigType, 11
FdIsoEn, 12	DmulrqFlagMasks
Filters, 12	Can_DmulrqCBParams, 8
IrqCallback, 12	
RxFifoBufDmaConfigNum, 12	EccEn
RxFifoBufDmaConfigPtr, 12	Can_HalConfigType, 11
RxOverWrite, 13	En
WakeupIrqCallback, 13	Can_TimeStampType, 20
Can_HalStateType	endAddress
Can_Hal_Types.h, 41	Can_MramAddressType, 17
Can_IrqCBParams, 13	ExtClkDiv
CommonIrqFlagMasks, 14	Can_TimeStampType, 20
Instance, 14	ExtClkSrc
Can_MessageIdType	Can_TimeStampType, 21
Can_Hal_Types.h, 41	extFilterStartAddress
Can_MessageInfoType, 14	Can_MramAddressType, 17
MsgBrs, 15	
MsgDataLen, 15	FdEn
MsgDataPtr, 15	Can_DeviceType, 6
MsgFdf, 15	Can_HalConfigType, 11
MsgId, 15	
MsgIdType, 15	FdIsoEn
MsgRtr, 16	Can_HalConfigType, 12
MsgRts, 16	Fifold
Can_MramAddressType, 16	Can_RxFifoBufDmaConfigType, 19
endAddress, 17	Filters
extFilterStartAddress, 17	Can_HalConfigType, 12
rxDbufferStartAddress, 17	FiltersNum
rxFifo0StartAddress, 17	Can_FiltersConfigType, 9
rxFifo1StartAddress, 17	FiltersParamsPtr
stdFilterStartAddress, 18	Can_FiltersConfigType, 10
txBufferStartAddress, 18	FrameType
	Can_FilterParamsType, 8

- ID1
 - Can_FilterParamsType, 9
- ID2
 - Can_FilterParamsType, 9
- ISR
 - Can_Hal.c, 25–27
- Instance
 - Can_DmulIrqCBParams, 8
 - Can_IrqCBParams, 14
- IrqCallback
 - Can_DeviceType, 6
 - Can_HalConfigType, 12
- Mode
 - Can_DeviceType, 6
- MsgBrs
 - Can_MessageInfoType, 15
- MsgDataLen
 - Can_MessageInfoType, 15
- MsgDataPtr
 - Can_MessageInfoType, 15
- MsgFdf
 - Can_MessageInfoType, 15
- MsgId
 - Can_MessageInfoType, 15
- MsgIdType
 - Can_MessageInfoType, 15
- MsgRtr
 - Can_MessageInfoType, 16
- MsgRts
 - Can_MessageInfoType, 16
- NormalBitrate
 - Can_BaudrateConfigType, 3
- PosEnd
 - Can_TimeStampType, 21
- Presc
 - Can_BitrateParamsType, 4
- rxDbufferStartAddress
 - Can_MramAddressType, 17
- rxFifo0StartAddress
 - Can_MramAddressType, 17
- rxFifo1StartAddress
 - Can_MramAddressType, 17
- RxFifoBufDmaConfigNum
 - Can_HalConfigType, 12
- RxFifoBufDmaConfigPtr
 - Can_HalConfigType, 12
- RxFifoDmaChannel
 - Can_DeviceType, 6
- RxFifoDmaChannelAddr
 - Can_DeviceType, 7
- RxOverWrite
 - Can_HalConfigType, 13
- Seg1
 - Can_BitrateParamsType, 5
- Seg2
 - Can_BitrateParamsType, 5
- Sjw
 - Can_BitrateParamsType, 5
- SspOffset
 - Can_BaudrateConfigType, 4
- stdFilterStartAddress
 - Can_MramAddressType, 18
- txBufferStartAddress
 - Can_MramAddressType, 18
- txEventfifoStartAddress
 - Can_MramAddressType, 18
- WakeupIrqCallback
 - Can_DeviceType, 7
 - Can_HalConfigType, 13