

AC784xx_DFP SENT
4.0.0

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	Sent_ChannelConfigType Struct Reference	3
3.1.1	Detailed Description	3
3.1.2	Member Data Documentation	3
3.1.2.1	EnabledInterrupts	3
3.1.2.2	ErrCallback	4
3.1.2.3	IoControl	4
3.1.2.4	NibblePointer	4
3.1.2.5	RxCallback	4
3.1.2.6	RxControl	4
3.1.2.7	TickUnit	5
3.1.2.8	TxCallback	5
3.1.2.9	WatchdogTimerLimit	5
3.2	Sent_FifoStatusType Struct Reference	5
3.2.1	Detailed Description	5
3.2.2	Member Data Documentation	6
3.2.2.1	FrameDataFifoAvail	6
3.2.2.2	FrameDataFifoEmpty	6
3.2.2.3	FrameDataFifoFull	6

3.2.2.4	FrameDataFifoHalfFull	6
3.3	Sent_FrameDataType Struct Reference	6
3.3.1	Detailed Description	7
3.3.2	Member Data Documentation	7
3.3.2.1	Data	7
3.3.2.2	FrameDataCRC	7
3.3.2.3	StatusNibble	7
3.3.2.4	TimeStamp	8
3.3.2.5	TotalFrameLength	8
3.4	Sent_IoCtrlType Struct Reference	8
3.4.1	Detailed Description	8
3.4.2	Member Data Documentation	8
3.4.2.1	GlitchFilterDepth	9
3.4.2.2	Input	9
3.4.2.3	InputInvertEnable	9
3.4.2.4	OutputInvertEnable	9
3.4.2.5	TriggerSource	9
3.5	Sent_ModuleConfigType Struct Reference	10
3.5.1	Detailed Description	10
3.5.2	Member Data Documentation	10
3.5.2.1	FractionDivider	10
3.5.2.2	TimeStampPreDivider	10
3.6	Sent_MsgType Struct Reference	10
3.6.1	Detailed Description	11
3.6.2	Member Data Documentation	11
3.6.2.1	DataFrame	11
3.6.2.2	FifoStatus	11
3.6.2.3	SerialFrame	11
3.7	Sent_NibblePointerType Struct Reference	12
3.7.1	Detailed Description	12
3.7.2	Member Data Documentation	12

3.7.2.1	Nibble0Position	12
3.7.2.2	Nibble1Position	12
3.7.2.3	Nibble2Position	13
3.7.2.4	Nibble3Position	13
3.7.2.5	Nibble4Position	13
3.7.2.6	Nibble5Position	13
3.7.2.7	Nibble6Position	13
3.7.2.8	Nibble7Position	14
3.8	Sent_ReceiveCtrlType Struct Reference	14
3.8.1	Detailed Description	14
3.8.2	Member Data Documentation	14
3.8.2.1	AlternateCRCEnable	14
3.8.2.2	DriftErrorDisable	15
3.8.2.3	FastCRCEnable	15
3.8.2.4	FDFL	15
3.8.2.5	FrameCheckMode	15
3.8.2.6	FrameLength	15
3.8.2.7	PausePulseEnable	16
3.8.2.8	SerialCRCEnable	16
3.8.2.9	SerialDataEnable	16
3.8.2.10	SerialMsgType	16
3.8.2.11	StatusIncludedInCRC	16
3.8.2.12	ZeroAugumentInCRCDisable	17
3.9	Sent_SerialFrameType Struct Reference	17
3.9.1	Detailed Description	17
3.9.2	Member Data Documentation	17
3.9.2.1	ConfigBit	17
3.9.2.2	SerialMsgCRC	18
3.9.2.3	SerialMsgData	18
3.9.2.4	SerialMsgID	18
3.10	Sent_StateType Struct Reference	18
3.10.1	Detailed Description	18
3.10.2	Member Data Documentation	19
3.10.2.1	EnabledInterrupts	19
3.10.2.2	ErrCallback	19
3.10.2.3	RxCallback	19
3.10.2.4	SentMsg	19
3.10.2.5	TxCallback	19
3.11	Sent_TransmitCtrlType Struct Reference	20
3.11.1	Detailed Description	20
3.11.2	Member Data Documentation	20
3.11.2.1	Mode	20
3.11.2.2	PulseDelayLength	20
3.11.2.3	PulseLength	20
3.11.2.4	TimeBase	20

4	File Documentation	21
4.1	AC784xx_API_Reference_Manual_SENT.pdf File Reference	21
4.2	AC784xx_Sent_Reg.h File Reference	21
4.2.1	Detailed Description	23
4.2.2	Macro Definition Documentation	23
4.2.2.1	IRQ_STATUS_MASK	24
4.2.3	Function Documentation	24
4.2.3.1	Sent_Reg_ClearIrqStatus()	24
4.2.3.2	Sent_Reg_DisableDriftError()	24
4.2.3.3	Sent_Reg_EnableAlternateCRC()	25
4.2.3.4	Sent_Reg_EnableChannel()	25
4.2.3.5	Sent_Reg_EnableCRCAugumentZero()	26
4.2.3.6	Sent_Reg_EnableFastMsgCRC()	26
4.2.3.7	Sent_Reg_EnableFDFLMode()	26
4.2.3.8	Sent_Reg_EnableFlushFifo()	27
4.2.3.9	Sent_Reg_EnableInputInvert()	27
4.2.3.10	Sent_Reg_EnableIrq()	28
4.2.3.11	Sent_Reg_EnableOutputInvert()	28
4.2.3.12	Sent_Reg_EnablePausePulse()	28
4.2.3.13	Sent_Reg_EnableSerialMsgCRC()	29
4.2.3.14	Sent_Reg_EnableSerialMsgProcess()	29
4.2.3.15	Sent_Reg_EnableStatusNibbleInCRC()	30
4.2.3.16	Sent_Reg_GetBase()	30
4.2.3.17	Sent_Reg_GetChannelStatus()	30
4.2.3.18	Sent_Reg_GetCurrentTimeStamp()	31
4.2.3.19	Sent_Reg_GetEnhancedConfigType()	31
4.2.3.20	Sent_Reg_GetFifoAvailable()	32
4.2.3.21	Sent_Reg_GetFifoEmptyStatus()	32
4.2.3.22	Sent_Reg_GetFifoFullStatus()	32
4.2.3.23	Sent_Reg_GetFifoHalfFullStatus()	33
4.2.3.24	Sent_Reg_GetFrameCRCNibble()	33

4.2.3.25	Sent_Reg_GetFrameDataNibbles()	33
4.2.3.26	Sent_Reg_GetFrameLengthCounter()	34
4.2.3.27	Sent_Reg_GetFrameStatusNibble()	34
4.2.3.28	Sent_Reg_GetIrqStatus()	35
4.2.3.29	Sent_Reg_GetModuleClkDivider()	35
4.2.3.30	Sent_Reg_GetSerialMsgCRC()	35
4.2.3.31	Sent_Reg_GetSerialMsgData()	36
4.2.3.32	Sent_Reg_GetSerialMsgID()	36
4.2.3.33	Sent_Reg_GetSpcTransferState()	36
4.2.3.34	Sent_Reg_SelectExternalTriggerSource()	37
4.2.3.35	Sent_Reg_SelectInputDataSource()	37
4.2.3.36	Sent_Reg_SetChannelFractionalDivider()	38
4.2.3.37	Sent_Reg_SetChannelPreDivider()	38
4.2.3.38	Sent_Reg_SetDataNibbleView()	38
4.2.3.39	Sent_Reg_SetFrameCheckMode()	39
4.2.3.40	Sent_Reg_SetFrameLength()	39
4.2.3.41	Sent_Reg_SetGlitchFilterDepth()	40
4.2.3.42	Sent_Reg_SetIrqStatus()	40
4.2.3.43	Sent_Reg_SetModuleClkDivider()	40
4.2.3.44	Sent_Reg_SetModuleTimeStampDivider()	41
4.2.3.45	Sent_Reg_SetSerialMsgType()	41
4.2.3.46	Sent_Reg_SetSPCParams()	42
4.2.3.47	Sent_Reg_SetSPCPulseDelay()	42
4.2.3.48	Sent_Reg_SetSPCPulseLength()	42
4.2.3.49	Sent_Reg_SetSPCTimebase()	43
4.2.3.50	Sent_Reg_SetSPCTriggerMode()	43
4.2.3.51	Sent_Reg_SetWatchdogLimit()	44
4.3	Sent_Hal.c File Reference	44
4.3.1	Detailed Description	45
4.3.2	Macro Definition Documentation	45
4.3.2.1	IRQ_CRCI	45

4.3.2.2	IRQ_ERROR	46
4.3.2.3	IRQ_ESI	46
4.3.2.4	IRQ_FDI	46
4.3.2.5	IRQ_FDONE	46
4.3.2.6	IRQ_FRI	46
4.3.2.7	IRQ_HFLL	46
4.3.2.8	IRQ_NNI	47
4.3.2.9	IRQ_NVI	47
4.3.2.10	IRQ_RDI	47
4.3.2.11	IRQ_RECEIVE	47
4.3.2.12	IRQ_RFI	47
4.3.2.13	IRQ_RSI	47
4.3.2.14	IRQ_SCRI	48
4.3.2.15	IRQ_SDI	48
4.3.2.16	IRQ_TBI	48
4.3.2.17	IRQ_TDI	48
4.3.2.18	IRQ_TRANSMIT	48
4.3.2.19	IRQ_WDI	48
4.3.2.20	IRQ_WSI	49
4.3.3	Function Documentation	49
4.3.3.1	ISR() [1/2]	49
4.3.3.2	ISR() [2/2]	49
4.3.3.3	Sent_Hal_ConfigChannel()	49
4.3.3.4	Sent_Hal_Deinit()	50
4.3.3.5	Sent_Hal_EnableChannel()	50
4.3.3.6	Sent_Hal_GetChannelStatus()	51
4.3.3.7	Sent_Hal_GetDefaultChannelConfig()	51
4.3.3.8	Sent_Hal_GetDefaultModuleConfig()	51
4.3.3.9	Sent_Hal_Init()	52
4.3.3.10	Sent_Hal_InitChannelTxCtrl()	52
4.4	Sent_Hal.h File Reference	52

4.4.1	Detailed Description	53
4.4.2	Function Documentation	53
4.4.2.1	Sent_Hal_ConfigChannel()	53
4.4.2.2	Sent_Hal_Deinit()	54
4.4.2.3	Sent_Hal_EnableChannel()	54
4.4.2.4	Sent_Hal_GetChannelStatus()	54
4.4.2.5	Sent_Hal_GetDefaultChannelConfig()	55
4.4.2.6	Sent_Hal_GetDefaultModuleConfig()	55
4.4.2.7	Sent_Hal_Init()	56
4.4.2.8	Sent_Hal_InitChannelTxCtrl()	56
4.5	Sent_Hal_Types.h File Reference	56
4.5.1	Typedef Documentation	58
4.5.1.1	Sent_CallbackType	58
4.5.1.2	Sent_RxCallbackType	58
4.5.2	Enumeration Type Documentation	58
4.5.2.1	anonymous enum	58
4.5.2.2	Sent_ChannelIdType	59
4.5.2.3	Sent_ChannelStatusType	59
4.5.2.4	Sent_EnhancedConfigType	60
4.5.2.5	Sent_ExternalTriggerType	60
4.5.2.6	Sent_FrameCheckModeType	60
4.5.2.7	Sent_InputDataSelectType	61
4.5.2.8	Sent_NibblePositonType	61
4.5.2.9	Sent_SerialMsgType	61
4.5.2.10	Sent_SpcTimeBaseType	62
4.5.2.11	Sent_SpcTriggerModeType	62
	Index	63

Chapter 1

Class Index

1.1 Class List

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

Sent_ChannelConfigType	
SENT channel configuration structure	3
Sent_FifoStatusType	
Define the sent fifo status	5
Sent_FrameDataType	
Define the sent frame data	6
Sent_IoCtrlType	
Define the sent channel io control config	8
Sent_ModuleConfigType	
SENT module configuration structure	10
Sent_MsgType	
Define the sent message	10
Sent_NibblePointerType	
Define the sent nibble position config	12
Sent_ReceiveCtrlType	
Define the sent channel receive config	14
Sent_SerialFrameType	
Define the sent serial frame	17
Sent_StateType	
ATC SENT runtime status	18
Sent_TransmitCtrlType	
Define the sent channel transmit config	20

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

AC784xx_API_Reference_Manual_SENT.pdf	21
AC784xx_Sent_Reg.h	
This file provides sent hardware access functions	21
Sent_Hal.c	
This file provides sent integration functions	44
Sent_Hal.h	
This file provides sent integration functions interface	52
Sent_Hal_Types.h	56

Chapter 3

Class Documentation

3.1 Sent_ChannelConfigType Struct Reference

SENT channel configuration structure.

```
#include <Sent_Hal_Types.h>
```

Public Attributes

- uint16 [WatchdogTimerLimit](#)
- [Sent_IoCtrlType](#) IoControl
- [Sent_ReceiveCtrlType](#) RxControl
- float32 [TickUnit](#)
- uint32 [EnabledInterrupts](#)
- [Sent_NibblePointerType](#) NibblePointer
- [Sent_RxCallbackType](#) RxCallback
- [Sent_CallbackType](#) TxCallback
- [Sent_CallbackType](#) ErrCallback

3.1.1 Detailed Description

SENT channel configuration structure.

Definition at line 308 of file [Sent_Hal_Types.h](#).

3.1.2 Member Data Documentation

3.1.2.1 EnabledInterrupts

```
uint32 Sent_ChannelConfigType::EnabledInterrupts
```

interrupts control

Definition at line 314 of file [Sent_Hal_Types.h](#).

3.1.2.2 ErrCallback

`Sent_CallbackType` `Sent_ChannelConfigType::ErrCallback`

sent channel error callback function

Definition at line 318 of file `Sent_Hal_Types.h`.

3.1.2.3 IoControl

`Sent_IoCtrlType` `Sent_ChannelConfigType::IoControl`

input output control property

Definition at line 311 of file `Sent_Hal_Types.h`.

3.1.2.4 NibblePointer

`Sent_NibblePointerType` `Sent_ChannelConfigType::NibblePointer`

nibble position control

Definition at line 315 of file `Sent_Hal_Types.h`.

3.1.2.5 RxCallback

`Sent_RxCallbackType` `Sent_ChannelConfigType::RxCallback`

sent receive callback function

Definition at line 316 of file `Sent_Hal_Types.h`.

3.1.2.6 RxControl

`Sent_ReceiveCtrlType` `Sent_ChannelConfigType::RxControl`

receive control property

Definition at line 312 of file `Sent_Hal_Types.h`.

3.1.2.7 TickUnit

```
float32 Sent_ChannelConfigType::TickUnit
```

desired tick time unit, unit:seconds

Definition at line 313 of file Sent_Hal_Types.h.

3.1.2.8 TxCallback

```
Sent_CallbackType Sent_ChannelConfigType::TxCallback
```

sent transmit callback user data

Definition at line 317 of file Sent_Hal_Types.h.

3.1.2.9 WatchdogTimerLimit

```
uint16 Sent_ChannelConfigType::WatchdogTimerLimit
```

watch dog timeout value

Definition at line 310 of file Sent_Hal_Types.h.

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

- [Sent_Hal_Types.h](#)

3.2 Sent_FifoStatusType Struct Reference

Define the sent fifo status.

```
#include <Sent_Hal_Types.h>
```

Public Attributes

- uint16 [FrameDataFifoAvail](#)
- boolean [FrameDataFifoFull](#)
- boolean [FrameDataFifoEmpty](#)
- boolean [FrameDataFifoHalfFull](#)

3.2.1 Detailed Description

Define the sent fifo status.

Definition at line 188 of file Sent_Hal_Types.h.

3.2.2 Member Data Documentation

3.2.2.1 FrameDataFifoAvail

```
uint16 Sent_FifoStatusType::FrameDataFifoAvail
```

fifo free space

Definition at line 190 of file Sent_Hal_Types.h.

3.2.2.2 FrameDataFifoEmpty

```
boolean Sent_FifoStatusType::FrameDataFifoEmpty
```

indicate fifo empty

Definition at line 192 of file Sent_Hal_Types.h.

3.2.2.3 FrameDataFifoFull

```
boolean Sent_FifoStatusType::FrameDataFifoFull
```

indicate fifo full

Definition at line 191 of file Sent_Hal_Types.h.

3.2.2.4 FrameDataFifoHalfFull

```
boolean Sent_FifoStatusType::FrameDataFifoHalfFull
```

indicate fifo half full

Definition at line 193 of file Sent_Hal_Types.h.

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

- [Sent_Hal_Types.h](#)

3.3 Sent_FrameDataType Struct Reference

Define the sent frame data.

```
#include <Sent_Hal_Types.h>
```

Public Attributes

- uint32 [Data](#)
- uint32 [TimeStamp](#)
- uint32 [TotalFrameLength](#)
- uint8 [StatusNibble](#)
- uint8 [FrameDataCRC](#)

3.3.1 Detailed Description

Define the sent frame data.

Definition at line 210 of file Sent_Hal_Types.h.

3.3.2 Member Data Documentation

3.3.2.1 Data

```
uint32 Sent_FrameDataType::Data
```

frame data nibble value, if frame length is bigger than 8, then data represent data value for every 8 nibbles

Definition at line 212 of file Sent_Hal_Types.h.

3.3.2.2 FrameDataCRC

```
uint8 Sent_FrameDataType::FrameDataCRC
```

frame data CRC

Definition at line 217 of file Sent_Hal_Types.h.

3.3.2.3 StatusNibble

```
uint8 Sent_FrameDataType::StatusNibble
```

frame status nibble value

Definition at line 216 of file Sent_Hal_Types.h.

3.3.2.4 TimeStamp

```
uint32 Sent_FrameDataType::TimeStamp
```

module time stamp

Definition at line 214 of file Sent_Hal_Types.h.

3.3.2.5 TotalFrameLength

```
uint32 Sent_FrameDataType::TotalFrameLength
```

frame time length

Definition at line 215 of file Sent_Hal_Types.h.

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

- [Sent_Hal_Types.h](#)

3.4 Sent_IoCtrlType Struct Reference

Define the sent channel io control config.

```
#include <Sent_Hal_Types.h>
```

Public Attributes

- boolean [InputInvertEnable](#)
- boolean [OutputInvertEnable](#)
- uint8 [GlitchFilterDepth](#)
- [Sent_InputDataSelectType](#) Input
- [Sent_ExternalTriggerType](#) TriggerSource

3.4.1 Detailed Description

Define the sent channel io control config.

Definition at line 241 of file Sent_Hal_Types.h.

3.4.2 Member Data Documentation

3.4.2.1 GlitchFilterDepth

```
uint8 Sent_IoCtrlType::GlitchFilterDepth
```

glitch filter depth

Definition at line 245 of file Sent_Hal_Types.h.

3.4.2.2 Input

```
Sent_InputDataSelectType Sent_IoCtrlType::Input
```

pulse starts after external trigger

Definition at line 246 of file Sent_Hal_Types.h.

3.4.2.3 InputInvertEnable

```
boolean Sent_IoCtrlType::InputInvertEnable
```

channel input invert enable

Definition at line 243 of file Sent_Hal_Types.h.

3.4.2.4 OutputInvertEnable

```
boolean Sent_IoCtrlType::OutputInvertEnable
```

channel output invert enable

Definition at line 244 of file Sent_Hal_Types.h.

3.4.2.5 TriggerSource

```
Sent_ExternalTriggerType Sent_IoCtrlType::TriggerSource
```

external trigger source select

Definition at line 247 of file Sent_Hal_Types.h.

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

- [Sent_Hal_Types.h](#)

3.5 Sent_ModuleConfigType Struct Reference

SENT module configuration structure.

```
#include <Sent_Hal_Types.h>
```

Public Attributes

- uint16 [FractionDivider](#)
- uint32 [TimeStampPreDivider](#)

3.5.1 Detailed Description

SENT module configuration structure.

Definition at line 299 of file Sent_Hal_Types.h.

3.5.2 Member Data Documentation

3.5.2.1 FractionDivider

```
uint16 Sent_ModuleConfigType::FractionDivider
```

module frantional divider

Definition at line 301 of file Sent_Hal_Types.h.

3.5.2.2 TimeStampPreDivider

```
uint32 Sent_ModuleConfigType::TimeStampPreDivider
```

timestamp divider

Definition at line 302 of file Sent_Hal_Types.h.

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

- [Sent_Hal_Types.h](#)

3.6 Sent_MsgType Struct Reference

Define the sent message.

```
#include <Sent_Hal_Types.h>
```

Public Attributes

- [Sent_SerialFrameType](#) SerialFrame
- [Sent_FrameDataType](#) DataFrame
- [Sent_FifoStatusType](#) FifoStatus

3.6.1 Detailed Description

Define the sent message.

Definition at line 224 of file Sent_Hal_Types.h.

3.6.2 Member Data Documentation

3.6.2.1 DataFrame

[Sent_FrameDataType](#) Sent_MsgType::DataFrame

fast message frame

Definition at line 227 of file Sent_Hal_Types.h.

3.6.2.2 FifoStatus

[Sent_FifoStatusType](#) Sent_MsgType::FifoStatus

fifo status

Definition at line 228 of file Sent_Hal_Types.h.

3.6.2.3 SerialFrame

[Sent_SerialFrameType](#) Sent_MsgType::SerialFrame

serial message frame

Definition at line 226 of file Sent_Hal_Types.h.

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

- [Sent_Hal_Types.h](#)

3.7 Sent_NibblePointerType Struct Reference

Define the sent nibble position config.

```
#include <Sent_Hal_Types.h>
```

Public Attributes

- [Sent_NibblePositonType Nibble0Position](#)
- [Sent_NibblePositonType Nibble1Position](#)
- [Sent_NibblePositonType Nibble2Position](#)
- [Sent_NibblePositonType Nibble3Position](#)
- [Sent_NibblePositonType Nibble4Position](#)
- [Sent_NibblePositonType Nibble5Position](#)
- [Sent_NibblePositonType Nibble6Position](#)
- [Sent_NibblePositonType Nibble7Position](#)

3.7.1 Detailed Description

Define the sent nibble position config.

Definition at line 253 of file Sent_Hal_Types.h.

3.7.2 Member Data Documentation

3.7.2.1 Nibble0Position

```
Sent\_NibblePositonType Sent_NibblePointerType::Nibble0Position
```

nibble 0 Position

Definition at line 255 of file Sent_Hal_Types.h.

3.7.2.2 Nibble1Position

```
Sent\_NibblePositonType Sent_NibblePointerType::Nibble1Position
```

nibble 1 Position

Definition at line 256 of file Sent_Hal_Types.h.

3.7.2.3 Nibble2Position

`Sent_NibblePositonType` `Sent_NibblePointerType::Nibble2Position`

nibble 2 Position

Definition at line 257 of file `Sent_Hal_Types.h`.

3.7.2.4 Nibble3Position

`Sent_NibblePositonType` `Sent_NibblePointerType::Nibble3Position`

nibble 3 Position

Definition at line 258 of file `Sent_Hal_Types.h`.

3.7.2.5 Nibble4Position

`Sent_NibblePositonType` `Sent_NibblePointerType::Nibble4Position`

nibble 4 Position

Definition at line 259 of file `Sent_Hal_Types.h`.

3.7.2.6 Nibble5Position

`Sent_NibblePositonType` `Sent_NibblePointerType::Nibble5Position`

nibble 5 Position

Definition at line 260 of file `Sent_Hal_Types.h`.

3.7.2.7 Nibble6Position

`Sent_NibblePositonType` `Sent_NibblePointerType::Nibble6Position`

nibble 6 Position

Definition at line 261 of file `Sent_Hal_Types.h`.

3.7.2.8 Nibble7Position

`Sent_NibblePositonType` `Sent_NibblePointerType::Nibble7Position`

nibble 7 Position

Definition at line 262 of file `Sent_Hal_Types.h`.

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

- [Sent_Hal_Types.h](#)

3.8 Sent_ReceiveCtrlType Struct Reference

Define the sent channel receive config.

```
#include <Sent_Hal_Types.h>
```

Public Attributes

- boolean [PausePulseEnable](#)
- boolean [AlternateCRCEnable](#)
- boolean [StatusIncludedInCRC](#)
- boolean [SerialDataEnable](#)
- boolean [SerialCRCEnable](#)
- boolean [FastCRCEnable](#)
- [Sent_FrameCheckModeType](#) [FrameCheckMode](#)
- uint8 [FrameLength](#)
- boolean [ZeroAugumentInCRCDisable](#)
- [Sent_SerialMsgType](#) [SerialMsgType](#)
- boolean [DriftErrorDisable](#)
- boolean [FDFL](#)

3.8.1 Detailed Description

Define the sent channel receive config.

Definition at line 268 of file `Sent_Hal_Types.h`.

3.8.2 Member Data Documentation

3.8.2.1 AlternateCRCEnable

`boolean` `Sent_ReceiveCtrlType::AlternateCRCEnable`

alternate CRC methethod enable

Definition at line 271 of file `Sent_Hal_Types.h`.

3.8.2.2 DriftErrorDisable

```
boolean Sent_ReceiveCtrlType::DriftErrorDisable
```

ignore drift error mode

Definition at line 280 of file Sent_Hal_Types.h.

3.8.2.3 FastCRCEnable

```
boolean Sent_ReceiveCtrlType::FastCRCEnable
```

fast channel CRC enable

Definition at line 275 of file Sent_Hal_Types.h.

3.8.2.4 FDFL

```
boolean Sent_ReceiveCtrlType::FDFL
```

frequency drift check based on frame length

Definition at line 281 of file Sent_Hal_Types.h.

3.8.2.5 FrameCheckMode

```
Sent\_FrameCheckModeType Sent_ReceiveCtrlType::FrameCheckMode
```

consecutive frame check mode

Definition at line 276 of file Sent_Hal_Types.h.

3.8.2.6 FrameLength

```
uint8 Sent_ReceiveCtrlType::FrameLength
```

data nibbles per frame

Definition at line 277 of file Sent_Hal_Types.h.

3.8.2.7 PausePulseEnable

```
boolean Sent_ReceiveCtrlType::PausePulseEnable
```

pause pulse enable

Definition at line 270 of file Sent_Hal_Types.h.

3.8.2.8 SerialCRCEnable

```
boolean Sent_ReceiveCtrlType::SerialCRCEnable
```

serial message CRC enable

Definition at line 274 of file Sent_Hal_Types.h.

3.8.2.9 SerialDataEnable

```
boolean Sent_ReceiveCtrlType::SerialDataEnable
```

serial message process enable

Definition at line 273 of file Sent_Hal_Types.h.

3.8.2.10 SerialMsgType

```
Sent\_SerialMsgType Sent_ReceiveCtrlType::SerialMsgType
```

serial message type

Definition at line 279 of file Sent_Hal_Types.h.

3.8.2.11 StatusIncludedInCRC

```
boolean Sent_ReceiveCtrlType::StatusIncludedInCRC
```

status included in CRC

Definition at line 272 of file Sent_Hal_Types.h.

3.8.2.12 ZeroAugumentInCRCDisable

```
boolean Sent_ReceiveCtrlType::ZeroAugumentInCRCDisable
```

CRC with zero nibble disable

Definition at line 278 of file Sent_Hal_Types.h.

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

- [Sent_Hal_Types.h](#)

3.9 Sent_SerialFrameType Struct Reference

Define the sent serial frame.

```
#include <Sent_Hal_Types.h>
```

Public Attributes

- uint8 [SerialMsgCRC](#)
- uint8 [SerialMsgID](#)
- uint16 [SerialMsgData](#)
- [Sent_EnhancedConfigType](#) ConfigBit

3.9.1 Detailed Description

Define the sent serial frame.

Definition at line 199 of file Sent_Hal_Types.h.

3.9.2 Member Data Documentation

3.9.2.1 ConfigBit

```
Sent\_EnhancedConfigType Sent_SerialFrameType::ConfigBit
```

Configuration bit

Definition at line 204 of file Sent_Hal_Types.h.

3.9.2.2 SerialMsgCRC

```
uint8 Sent_SerialFrameType::SerialMsgCRC
```

Serial message CRC

Definition at line 201 of file Sent_Hal_Types.h.

3.9.2.3 SerialMsgData

```
uint16 Sent_SerialFrameType::SerialMsgData
```

Serial message data

Definition at line 203 of file Sent_Hal_Types.h.

3.9.2.4 SerialMsgID

```
uint8 Sent_SerialFrameType::SerialMsgID
```

Serial message ID

Definition at line 202 of file Sent_Hal_Types.h.

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

- [Sent_Hal_Types.h](#)

3.10 Sent_StateType Struct Reference

ATC SENT runtime status.

Public Attributes

- [Sent_MsgType](#) SentMsg
- [uint32](#) EnabledInterrupts
- [Sent_RxCallbackType](#) RxCallback
- [Sent_CallbackType](#) TxCallback
- [Sent_CallbackType](#) ErrCallback

3.10.1 Detailed Description

ATC SENT runtime status.

Definition at line 82 of file Sent_Hal.c.

3.10.2 Member Data Documentation

3.10.2.1 EnabledInterrupts

`uint32 Sent_StateType::EnabledInterrupts`

interrupts control

Definition at line 85 of file `Sent_Hal.c`.

3.10.2.2 ErrCallback

`Sent_CallbackType Sent_StateType::ErrCallback`

sent channel error callback function

Definition at line 88 of file `Sent_Hal.c`.

3.10.2.3 RxCallback

`Sent_RxCallbackType Sent_StateType::RxCallback`

sent receive callback function

Definition at line 86 of file `Sent_Hal.c`.

3.10.2.4 SentMsg

`Sent_MsgType Sent_StateType::SentMsg`

sent message frame

Definition at line 84 of file `Sent_Hal.c`.

3.10.2.5 TxCallback

`Sent_CallbackType Sent_StateType::TxCallback`

sent transmit callback user data

Definition at line 87 of file `Sent_Hal.c`.

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

- [Sent_Hal.c](#)

3.11 Sent_TransmitCtrlType Struct Reference

Define the sent channel transmit config.

```
#include <Sent_Hal_Types.h>
```

Public Attributes

- uint8 [PulseLength](#)
- uint8 [PulseDelayLength](#)
- [Sent_SpcTriggerModeType](#) Mode
- [Sent_SpcTimeBaseType](#) TimeBase

3.11.1 Detailed Description

Define the sent channel transmit config.

Definition at line 288 of file `Sent_Hal_Types.h`.

3.11.2 Member Data Documentation

3.11.2.1 Mode

[Sent_SpcTriggerModeType](#) `Sent_TransmitCtrlType::Mode`

spc trigger mode

Definition at line 292 of file `Sent_Hal_Types.h`.

3.11.2.2 PulseDelayLength

`uint8 Sent_TransmitCtrlType::PulseDelayLength`

pause delay length

Definition at line 291 of file `Sent_Hal_Types.h`.

3.11.2.3 PulseLength

`uint8 Sent_TransmitCtrlType::PulseLength`

pulse length

Definition at line 290 of file `Sent_Hal_Types.h`.

3.11.2.4 TimeBase

[Sent_SpcTimeBaseType](#) `Sent_TransmitCtrlType::TimeBase`

time base

Definition at line 293 of file `Sent_Hal_Types.h`.

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

- [Sent_Hal_Types.h](#)

Chapter 4

File Documentation

4.1 AC784xx_API_Reference_Manual_SENT.pdf File Reference

4.2 AC784xx_Sent_Reg.h File Reference

This file provides sent hardware access functions.

```
#include "Device_Register.h"
#include "Sent_Hal_Types.h"
```

Macros

- `#define IRQ_STATUS_MASK`

Functions

- LOCAL_INLINE SENT_CHANNEL_Type * [Sent_Reg_GetBase](#) (uint8 Instance)
Get SENT base.
- LOCAL_INLINE void [Sent_Reg_EnableChannel](#) (SENT_CHANNEL_Type *Base, boolean Enable)
SENT set channel enable bit.
- LOCAL_INLINE void [Sent_Reg_EnablePausePulse](#) (SENT_CHANNEL_Type *Base, boolean Enable)
SENT set channel pause pulse receive enable bit.
- LOCAL_INLINE void [Sent_Reg_EnableAlternateCRC](#) (SENT_CHANNEL_Type *Base, boolean Enable)
SENT set channel alternate CRC enable bit.
- LOCAL_INLINE void [Sent_Reg_EnableStatusNibbleInCRC](#) (SENT_CHANNEL_Type *Base, boolean Enable)
SENT set channel status nibble in CRC enable bit.
- LOCAL_INLINE void [Sent_Reg_EnableSerialMsgProcess](#) (SENT_CHANNEL_Type *Base, boolean Enable)
SENT set serial message process enable bit.
- LOCAL_INLINE void [Sent_Reg_EnableSerialMsgCRC](#) (SENT_CHANNEL_Type *Base, boolean Enable)
SENT set serial message CRC enable bit.
- LOCAL_INLINE void [Sent_Reg_EnableFastMsgCRC](#) (SENT_CHANNEL_Type *Base, boolean Enable)
SENT set fast message CRC enable bit.
- LOCAL_INLINE void [Sent_Reg_DisableDriftError](#) (SENT_CHANNEL_Type *Base, boolean Enable)
SENT set drift error enable bit.
- LOCAL_INLINE void [Sent_Reg_EnableCRCAugumentZero](#) (SENT_CHANNEL_Type *Base, boolean Enable)

- SENT set fast CRC augument zero enable bit.*
- LOCAL_INLINE void [Sent_Reg_EnableFDFLMode](#) (SENT_CHANNEL_Type *Base, boolean Enable)
- SENT set FDFL mode enable bit.*
- LOCAL_INLINE [Sent_ChannelStatusType](#) [Sent_Reg_GetChannelStatus](#) (const SENT_CHANNEL_Type *Base)
- SENT get channel state.*
- LOCAL_INLINE [Sent_EnhancedConfigType](#) [Sent_Reg_GetEnhancedConfigType](#) (const SENT_CHANNEL_Type *Base)
- SENT get enhanced serial msg config type.*
- LOCAL_INLINE void [Sent_Reg_EnableOutputInvert](#) (SENT_CHANNEL_Type *Base, boolean Enable)
- SENT set output invert enable bit.*
- LOCAL_INLINE void [Sent_Reg_EnableInputInvert](#) (SENT_CHANNEL_Type *Base, boolean Enable)
- SENT set input invert enable bit.*
- LOCAL_INLINE uint32 [Sent_Reg_GetIrqStatus](#) (const SENT_CHANNEL_Type *Base)
- SENT get irq status.*
- LOCAL_INLINE void [Sent_Reg_SetIrqStatus](#) (SENT_CHANNEL_Type *Base, uint32 Irq)
- SENT set irq status.*
- LOCAL_INLINE void [Sent_Reg_ClearIrqStatus](#) (SENT_CHANNEL_Type *Base, uint32 Irq)
- SENT clear irq status.*
- LOCAL_INLINE void [Sent_Reg_EnableIrq](#) (SENT_CHANNEL_Type *Base, uint32 Irq)
- SENT set irq enable bit.*
- LOCAL_INLINE void [Sent_Reg_EnableFlushFifo](#) (SENT_CHANNEL_Type *Base, boolean Enable)
- SENT set fifo flush status.*
- LOCAL_INLINE void [Sent_Reg_SetModuleClkDivider](#) (SENT_CTRL_Type *Base, uint16 Step)
- SENT set moulde clock divider.*
- LOCAL_INLINE uint16 [Sent_Reg_GetModuleClkDivider](#) (const SENT_CTRL_Type *Base)
- SENT get moulde clock divider.*
- LOCAL_INLINE uint32 [Sent_Reg_GetCurrentTimeStamp](#) (const SENT_CTRL_Type *Base)
- SENT get module timestamp vaule.*
- LOCAL_INLINE void [Sent_Reg_SetModuleTimeStampDivider](#) (SENT_CTRL_Type *Base, uint32 Step)
- SENT set moulde timestamp divider.*
- LOCAL_INLINE void [Sent_Reg_SetChannelPreDivider](#) (SENT_CHANNEL_Type *Base, uint16 Step)
- SENT set channel pre divider.*
- LOCAL_INLINE void [Sent_Reg_SetChannelFractionalDivider](#) (SENT_CHANNEL_Type *Base, uint16 Step)
- SENT set channel fractional divider divider.*
- LOCAL_INLINE void [Sent_Reg_SetFrameCheckMode](#) (SENT_CHANNEL_Type *Base, [Sent_FrameCheckModeType](#) Mode)
- SENT set channel frame check mode.*
- LOCAL_INLINE void [Sent_Reg_SetFrameLength](#) (SENT_CHANNEL_Type *Base, uint8 Length)
- SENT set fast channel frame length.*
- LOCAL_INLINE void [Sent_Reg_SetSerialMsgType](#) (SENT_CHANNEL_Type *Base, [Sent_SerialMsgType](#) Type)
- SENT set serial message type.*
- LOCAL_INLINE uint8 [Sent_Reg_GetFrameStatusNibble](#) (const SENT_CHANNEL_Type *Base)
- SENT get frame status nibble.*
- LOCAL_INLINE uint8 [Sent_Reg_GetFrameCRCNibble](#) (const SENT_CHANNEL_Type *Base)
- SENT get frame CRC nibble.*
- LOCAL_INLINE uint32 [Sent_Reg_GetFrameLengthCounter](#) (const SENT_CHANNEL_Type *Base)
- SENT get frame length counter.*
- LOCAL_INLINE uint8 [Sent_Reg_GetSerialMsgCRC](#) (const SENT_CHANNEL_Type *Base)
- SENT get serial message CRC value.*
- LOCAL_INLINE uint8 [Sent_Reg_GetSerialMsgID](#) (const SENT_CHANNEL_Type *Base)
- SENT get serial message ID.*
- LOCAL_INLINE uint16 [Sent_Reg_GetSerialMsgData](#) (const SENT_CHANNEL_Type *Base)
- SENT get serial message data.*

- LOCAL_INLINE void [Sent_Reg_SelectInputDataSource](#) (SENT_CHANNEL_Type *Base, [Sent_InputDataSourceType](#) Input)
SENT set input data source.
- LOCAL_INLINE void [Sent_Reg_SetGlitchFilterDepth](#) (SENT_CHANNEL_Type *Base, uint8 Depth)
SENT set glitch filter depth.
- LOCAL_INLINE void [Sent_Reg_SelectExternalTriggerSource](#) (SENT_CHANNEL_Type *Base, [Sent_ExternalTriggerType](#) Trigger)
SENT set external trigger source.
- LOCAL_INLINE uint8 [Sent_Reg_GetSpcTransferState](#) (const SENT_CHANNEL_Type *Base)
SENT get SPC transfer state.
- LOCAL_INLINE void [Sent_Reg_SetSPCTimebase](#) (SENT_CHANNEL_Type *Base, [Sent_SpcTimeBaseType](#) Timebase)
SENT set SPC pulse timebase type.
- LOCAL_INLINE void [Sent_Reg_SetSPCPulseLength](#) (SENT_CHANNEL_Type *Base, uint8 Length)
SENT set SPC pulse length.
- LOCAL_INLINE void [Sent_Reg_SetSPCPulseDelay](#) (SENT_CHANNEL_Type *Base, uint8 Delay)
SENT set SPC pulse delay.
- LOCAL_INLINE void [Sent_Reg_SetSPCTriggerMode](#) (SENT_CHANNEL_Type *Base, [Sent_SpcTriggerModeType](#) Mode)
SENT set SPC pulse trigger mode.
- LOCAL_INLINE void [Sent_Reg_SetSPCParams](#) (SENT_CHANNEL_Type *Base, uint32 Param)
SENT set spc parameters.
- LOCAL_INLINE void [Sent_Reg_SetDataNibbleView](#) (SENT_CHANNEL_Type *Base, uint32 View)
SENT set data nibble view.
- LOCAL_INLINE uint8 [Sent_Reg_GetFifoEmptyStatus](#) (const SENT_CHANNEL_Type *Base)
SENT get fifo empty status.
- LOCAL_INLINE uint8 [Sent_Reg_GetFifoHalfFullStatus](#) (const SENT_CHANNEL_Type *Base)
SENT get fifo half full status.
- LOCAL_INLINE uint8 [Sent_Reg_GetFifoFullStatus](#) (const SENT_CHANNEL_Type *Base)
SENT get fifo full status.
- LOCAL_INLINE uint8 [Sent_Reg_GetFifoAvailable](#) (const SENT_CHANNEL_Type *Base)
SENT get fifo available bytes.
- LOCAL_INLINE void [Sent_Reg_SetWatchdogLimit](#) (SENT_CHANNEL_Type *Base, uint16 Limit)
SENT set watchdog limit.
- LOCAL_INLINE uint32 [Sent_Reg_GetFrameDataNibbles](#) (const SENT_CHANNEL_Type *Base)
SENT get frame data nibbles.

4.2.1 Detailed Description

This file provides sent hardware access functions.

4.2.2 Macro Definition Documentation

4.2.2.1 IRQ_STATUS_MASK

```
#define IRQ_STATUS_MASK
```

Value:

```
(SENT_CHANNEL_INTSTAT_RSI_Msk | SENT_CHANNEL_INTSTAT_RDI_Msk \
 | SENT_CHANNEL_INTSTAT_RFI_Msk | SENT_CHANNEL_INTSTAT_TDI_Msk \
 | SENT_CHANNEL_INTSTAT_TBI_Msk | SENT_CHANNEL_INTSTAT_FRI_Msk \
 | SENT_CHANNEL_INTSTAT_FDI_Msk | SENT_CHANNEL_INTSTAT_NNI_Msk \
 | SENT_CHANNEL_INTSTAT_NVI_Msk | SENT_CHANNEL_INTSTAT_CRCI_Msk \
 | SENT_CHANNEL_INTSTAT_WSI_Msk | SENT_CHANNEL_INTSTAT_SDI_Msk \
 | SENT_CHANNEL_INTSTAT_SCRI_Msk | SENT_CHANNEL_INTSTAT_WDI_Msk \
 | SENT_CHANNEL_INTSTAT_ESI_Msk | SENT_CHANNEL_INTSTAT_HFLL_Msk \
 | SENT_CHANNEL_INTSTAT_FDONE_Msk)
```

Definition at line 59 of file AC784xx_Sent_Reg.h.

4.2.3 Function Documentation

4.2.3.1 Sent_Reg_ClearIrqStatus()

```
LOCAL_INLINE void Sent_Reg_ClearIrqStatus (
    SENT_CHANNEL_Type * Base,
    uint32 Irq )
```

SENT clear irq status.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Irq</i>	irq status

Returns

void

Definition at line 386 of file AC784xx_Sent_Reg.h.

4.2.3.2 Sent_Reg_DisableDriftError()

```
LOCAL_INLINE void Sent_Reg_DisableDriftError (
    SENT_CHANNEL_Type * Base,
    boolean Enable )
```

SENT set drift error enable bit.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Enable</i>	drift error enable or disable

Returns

void

Definition at line 236 of file AC784xx_Sent_Reg.h.

4.2.3.3 Sent_Reg_EnableAlternateCRC()

```
LOCAL_INLINE void Sent_Reg_EnableAlternateCRC (
    SENT_CHANNEL_Type * Base,
    boolean Enable )
```

SENT set channel alternate CRC enable bit.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Enable</i>	Alternate CRC enable or disable

Returns

void

Definition at line 141 of file AC784xx_Sent_Reg.h.

4.2.3.4 Sent_Reg_EnableChannel()

```
LOCAL_INLINE void Sent_Reg_EnableChannel (
    SENT_CHANNEL_Type * Base,
    boolean Enable )
```

SENT set channel enable bit.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Enable</i>	channel enable or disable

Returns

void

Definition at line 103 of file AC784xx_Sent_Reg.h.

4.2.3.5 Sent_Reg_EnableCRCAugumentZero()

```
LOCAL_INLINE void Sent_Reg_EnableCRCAugumentZero (
    SENT_CHANNEL_Type * Base,
    boolean Enable )
```

SENT set fast CRC augument zero enable bit.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Enable</i>	CRC augument zero enable or disable

Returns

void

Definition at line 255 of file AC784xx_Sent_Reg.h.

4.2.3.6 Sent_Reg_EnableFastMsgCRC()

```
LOCAL_INLINE void Sent_Reg_EnableFastMsgCRC (
    SENT_CHANNEL_Type * Base,
    boolean Enable )
```

SENT set fast message CRC enable bit.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Enable</i>	fast message CRC enable or disable

Returns

void

Definition at line 217 of file AC784xx_Sent_Reg.h.

4.2.3.7 Sent_Reg_EnableFDFLMode()

```
LOCAL_INLINE void Sent_Reg_EnableFDFLMode (
    SENT_CHANNEL_Type * Base,
    boolean Enable )
```

SENT set FDFL mode enable bit.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Enable</i>	FDFL mode enable or disable

Returns

void

Definition at line 274 of file AC784xx_Sent_Reg.h.

4.2.3.8 Sent_Reg_EnableFlushFifo()

```
LOCAL_INLINE void Sent_Reg_EnableFlushFifo (
    SENT_CHANNEL_Type * Base,
    boolean Enable )
```

SENT set fifo flush status.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Enable</i>	fifo flush enable or disable

Returns

void

Definition at line 410 of file AC784xx_Sent_Reg.h.

4.2.3.9 Sent_Reg_EnableInputInvert()

```
LOCAL_INLINE void Sent_Reg_EnableInputInvert (
    SENT_CHANNEL_Type * Base,
    boolean Enable )
```

SENT set input invert enable bit.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Enable</i>	input invert enable or disable

Returns

void

Definition at line 344 of file AC784xx_Sent_Reg.h.

4.2.3.10 Sent_Reg_EnableIrq()

```
LOCAL_INLINE void Sent_Reg_EnableIrq (
    SENT_CHANNEL_Type * Base,
    uint32 Irq )
```

SENT set irq enable bit.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Irq</i>	Irq enable bit

Returns

void

Definition at line 398 of file AC784xx_Sent_Reg.h.

4.2.3.11 Sent_Reg_EnableOutputInvert()

```
LOCAL_INLINE void Sent_Reg_EnableOutputInvert (
    SENT_CHANNEL_Type * Base,
    boolean Enable )
```

SENT set output invert enable bit.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Enable</i>	output invert enable or disable

Returns

void

Definition at line 325 of file AC784xx_Sent_Reg.h.

4.2.3.12 Sent_Reg_EnablePausePulse()

```
LOCAL_INLINE void Sent_Reg_EnablePausePulse (
    SENT_CHANNEL_Type * Base,
    boolean Enable )
```

SENT set channel pause pulse receive enable bit.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Enable</i>	pause pulse enable or disable

Returns

void

Definition at line 122 of file AC784xx_Sent_Reg.h.

4.2.3.13 Sent_Reg_EnableSerialMsgCRC()

```
LOCAL_INLINE void Sent_Reg_EnableSerialMsgCRC (  
    SENT_CHANNEL_Type * Base,  
    boolean Enable )
```

SENT set serial message CRC enable bit.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Enable</i>	serial message CRC enable or disable

Returns

void

Definition at line 198 of file AC784xx_Sent_Reg.h.

4.2.3.14 Sent_Reg_EnableSerialMsgProcess()

```
LOCAL_INLINE void Sent_Reg_EnableSerialMsgProcess (  
    SENT_CHANNEL_Type * Base,  
    boolean Enable )
```

SENT set serial message process enable bit.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Enable</i>	serial message process enable or disable

Returns

void

Definition at line 179 of file AC784xx_Sent_Reg.h.

4.2.3.15 Sent_Reg_EnableStatusNibbleInCRC()

```
LOCAL_INLINE void Sent_Reg_EnableStatusNibbleInCRC (
    SENT_CHANNEL_Type * Base,
    boolean Enable )
```

SENT set channel status nibble in CRC enable bit.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Enable</i>	status nibble in CRC enable or disable

Returns

void

Definition at line 160 of file AC784xx_Sent_Reg.h.

4.2.3.16 Sent_Reg_GetBase()

```
LOCAL_INLINE SENT_CHANNEL_Type* Sent_Reg_GetBase (
    uint8 Instance )
```

Get SENT base.

Parameters

in	<i>Instance</i>	: SENT hardware channel ID.
----	-----------------	-----------------------------

Returns

SENT_Type*: the SENT base addr.

Definition at line 82 of file AC784xx_Sent_Reg.h.

4.2.3.17 Sent_Reg_GetChannelStatus()

```
LOCAL_INLINE Sent_ChannelStatusType Sent_Reg_GetChannelStatus (
    const SENT_CHANNEL_Type * Base )
```

SENT get channel state.

Parameters

in	<i>Base</i>	SENT base pointer
----	-------------	-------------------

Returns

channel status

Definition at line 292 of file AC784xx_Sent_Reg.h.

4.2.3.18 Sent_Reg_GetCurrentTimeStamp()

```
LOCAL_INLINE uint32 Sent_Reg_GetCurrentTimeStamp (  
    const SENT_CTRL_Type * Base )
```

SENT get module timestamp vaule.

Parameters

in	<i>Base</i>	SENT module base pointer
----	-------------	--------------------------

Returns

module timestamp vaule

Definition at line 453 of file AC784xx_Sent_Reg.h.

4.2.3.19 Sent_Reg_GetEnhancedConfigType()

```
LOCAL_INLINE Sent\_EnhancedConfigType Sent_Reg_GetEnhancedConfigType (  
    const SENT_CHANNEL_Type * Base )
```

SENT get enhanced serial msg config type.

Parameters

in	<i>Base</i>	SENT channel base pointer
----	-------------	---------------------------

Returns

enhanced serial msg config type

Definition at line 308 of file AC784xx_Sent_Reg.h.

4.2.3.20 Sent_Reg_GetFifoAvailable()

```
LOCAL_INLINE uint8 Sent_Reg_GetFifoAvailable (
    const SENT_CHANNEL_Type * Base )
```

SENT get fifo available bytes.

Parameters

in	<i>Base</i>	SENT channel base pointer
----	-------------	---------------------------

Returns

fifo available bytes

Definition at line 794 of file AC784xx_Sent_Reg.h.

4.2.3.21 Sent_Reg_GetFifoEmptyStatus()

```
LOCAL_INLINE uint8 Sent_Reg_GetFifoEmptyStatus (
    const SENT_CHANNEL_Type * Base )
```

SENT get fifo empty status.

Parameters

in	<i>Base</i>	SENT channel base pointer
----	-------------	---------------------------

Returns

fifo empty status

Definition at line 761 of file AC784xx_Sent_Reg.h.

4.2.3.22 Sent_Reg_GetFifoFullStatus()

```
LOCAL_INLINE uint8 Sent_Reg_GetFifoFullStatus (
    const SENT_CHANNEL_Type * Base )
```

SENT get fifo full status.

Parameters

in	<i>Base</i>	SENT channel base pointer
----	-------------	---------------------------

Returns

fifo full status

Definition at line 783 of file AC784xx_Sent_Reg.h.

4.2.3.23 Sent_Reg_GetFifoHalfFullStatus()

```
LOCAL_INLINE uint8 Sent_Reg_GetFifoHalfFullStatus (  
    const SENT_CHANNEL_Type * Base )
```

SENT get fifo half full status.

Parameters

in	<i>Base</i>	SENT channel base pointer
----	-------------	---------------------------

Returns

fifo half full status

Definition at line 772 of file AC784xx_Sent_Reg.h.

4.2.3.24 Sent_Reg_GetFrameCRCNibble()

```
LOCAL_INLINE uint8 Sent_Reg_GetFrameCRCNibble (  
    const SENT_CHANNEL_Type * Base )
```

SENT get frame CRC nibble.

Parameters

in	<i>Base</i>	SENT channel base pointer
----	-------------	---------------------------

Returns

frame CRC nibble

Definition at line 565 of file AC784xx_Sent_Reg.h.

4.2.3.25 Sent_Reg_GetFrameDataNibbles()

```
LOCAL_INLINE uint32 Sent_Reg_GetFrameDataNibbles (  
    const SENT_CHANNEL_Type * Base )
```

SENT get frame data nibbles.

Parameters

in	<i>Base</i>	SENT channel base pointer
----	-------------	---------------------------

Returns

frame data nibbles

Definition at line 820 of file AC784xx_Sent_Reg.h.

4.2.3.26 Sent_Reg_GetFrameLengthCounter()

```
LOCAL_INLINE uint32 Sent_Reg_GetFrameLengthCounter (  
    const SENT_CHANNEL_Type * Base )
```

SENT get frame length counter.

Parameters

in	<i>Base</i>	SENT channel base pointer
----	-------------	---------------------------

Returns

frame length counter, including pause nibble

Definition at line 576 of file AC784xx_Sent_Reg.h.

4.2.3.27 Sent_Reg_GetFrameStatusNibble()

```
LOCAL_INLINE uint8 Sent_Reg_GetFrameStatusNibble (  
    const SENT_CHANNEL_Type * Base )
```

SENT get frame status nibbble.

Parameters

in	<i>Base</i>	SENT channel base pointer
----	-------------	---------------------------

Returns

frame status nibbble

Definition at line 554 of file AC784xx_Sent_Reg.h.

4.2.3.28 Sent_Reg_GetIrqStatus()

```
LOCAL_INLINE uint32 Sent_Reg_GetIrqStatus (
    const SENT_CHANNEL_Type * Base )
```

SENT get irq status.

Parameters

in	<i>Base</i>	SENT channel base pointer
----	-------------	---------------------------

Returns

irq status

Definition at line 362 of file AC784xx_Sent_Reg.h.

4.2.3.29 Sent_Reg_GetModuleClkDivider()

```
LOCAL_INLINE uint16 Sent_Reg_GetModuleClkDivider (
    const SENT_CTRL_Type * Base )
```

SENT get module clock divider.

Parameters

in	<i>Base</i>	SENT module base pointer
----	-------------	--------------------------

Returns

module clock divider value

Definition at line 442 of file AC784xx_Sent_Reg.h.

4.2.3.30 Sent_Reg_GetSerialMsgCRC()

```
LOCAL_INLINE uint8 Sent_Reg_GetSerialMsgCRC (
    const SENT_CHANNEL_Type * Base )
```

SENT get serial message CRC value.

Parameters

in	<i>Base</i>	SENT channel base pointer
----	-------------	---------------------------

Returns

serial message CRC value

Definition at line 587 of file AC784xx_Sent_Reg.h.

4.2.3.31 Sent_Reg_GetSerialMsgData()

```
LOCAL_INLINE uint16 Sent_Reg_GetSerialMsgData (  
    const SENT_CHANNEL_Type * Base )
```

SENT get serial message data.

Parameters

in	<i>Base</i>	SENT channel base pointer
----	-------------	---------------------------

Returns

serial message message data

Definition at line 609 of file AC784xx_Sent_Reg.h.

4.2.3.32 Sent_Reg_GetSerialMsgID()

```
LOCAL_INLINE uint8 Sent_Reg_GetSerialMsgID (  
    const SENT_CHANNEL_Type * Base )
```

SENT get serial message ID.

Parameters

in	<i>Base</i>	SENT channel base pointer
----	-------------	---------------------------

Returns

serial message ID

Definition at line 598 of file AC784xx_Sent_Reg.h.

4.2.3.33 Sent_Reg_GetSpcTransferState()

```
LOCAL_INLINE uint8 Sent_Reg_GetSpcTransferState (  
    const SENT_CHANNEL_Type * Base )
```

SENT get SPC transfer state.

Parameters

in	<i>Base</i>	SENT channel base pointer
----	-------------	---------------------------

Returns

whether SPC pulse is being sent

Definition at line 666 of file AC784xx_Sent_Reg.h.

4.2.3.34 Sent_Reg_SelectExternalTriggerSource()

```
LOCAL_INLINE void Sent_Reg_SelectExternalTriggerSource (
    SENT_CHANNEL_Type * Base,
    Sent_ExternalTriggerType Trigger )
```

SENT set external trigger source.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Trigger</i>	external trigger source 0~3

Returns

void

Definition at line 652 of file AC784xx_Sent_Reg.h.

4.2.3.35 Sent_Reg_SelectInputDataSource()

```
LOCAL_INLINE void Sent_Reg_SelectInputDataSource (
    SENT_CHANNEL_Type * Base,
    Sent_InputDataSelectType Input )
```

SENT set input data source.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Input</i>	sent input data source 0~1

Returns

void

Definition at line 621 of file AC784xx_Sent_Reg.h.

4.2.3.36 Sent_Reg_SetChannelFractionalDivider()

```
LOCAL_INLINE void Sent_Reg_SetChannelFractionalDivider (
    SENT_CHANNEL_Type * Base,
    uint16 Step )
```

SENT set channel fractional divider divider.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Step</i>	divider value

Returns

void

Definition at line 495 of file AC784xx_Sent_Reg.h.

4.2.3.37 Sent_Reg_SetChannelPreDivider()

```
LOCAL_INLINE void Sent_Reg_SetChannelPreDivider (
    SENT_CHANNEL_Type * Base,
    uint16 Step )
```

SENT set channel pre divider.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Step</i>	divider value

Returns

void

Definition at line 480 of file AC784xx_Sent_Reg.h.

4.2.3.38 Sent_Reg_SetDataNibbleView()

```
LOCAL_INLINE void Sent_Reg_SetDataNibbleView (
    SENT_CHANNEL_Type * Base,
    uint32 View )
```

SENT set data nibble view.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>View</i>	data nibble view value, data nibble order can be adjusted

Returns

void

Definition at line 750 of file AC784xx_Sent_Reg.h.

4.2.3.39 Sent_Reg_SetFrameCheckMode()

```
LOCAL_INLINE void Sent_Reg_SetFrameCheckMode (
    SENT_CHANNEL_Type * Base,
    Sent_FrameCheckModeType Mode )
```

SENT set channel frame check mode.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Mode</i>	frame check mode

Returns

void

Definition at line 510 of file AC784xx_Sent_Reg.h.

4.2.3.40 Sent_Reg_SetFrameLength()

```
LOCAL_INLINE void Sent_Reg_SetFrameLength (
    SENT_CHANNEL_Type * Base,
    uint8 Length )
```

SENT set fast channel frame length.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Length</i>	frame length

Returns

void

Definition at line 525 of file AC784xx_Sent_Reg.h.

4.2.3.41 Sent_Reg_SetGlitchFilterDepth()

```
LOCAL_INLINE void Sent_Reg_SetGlitchFilterDepth (
    SENT_CHANNEL_Type * Base,
    uint8 Depth )
```

SENT set glitch filter depth.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Depth</i>	glitch filter depth, received data is valid if it crosses the filter depth

Returns

void

Definition at line 637 of file AC784xx_Sent_Reg.h.

4.2.3.42 Sent_Reg_SetIrqStatus()

```
LOCAL_INLINE void Sent_Reg_SetIrqStatus (
    SENT_CHANNEL_Type * Base,
    uint32 Irq )
```

SENT set irq status.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Irq</i>	irq status

Returns

void

Definition at line 374 of file AC784xx_Sent_Reg.h.

4.2.3.43 Sent_Reg_SetModuleClkDivider()

```
LOCAL_INLINE void Sent_Reg_SetModuleClkDivider (
    SENT_CTRL_Type * Base,
    uint16 Step )
```

SENT set module clock divider.

Parameters

in	<i>Base</i>	SENT module base pointer
in	<i>Step</i>	divider value

Definition at line 428 of file AC784xx_Sent_Reg.h.

4.2.3.44 Sent_Reg_SetModuleTimeStampDivider()

```
LOCAL_INLINE void Sent_Reg_SetModuleTimeStampDivider (
    SENT_CTRL_Type * Base,
    uint32 Step )
```

SENT set module timestamp divider.

Parameters

in	<i>Base</i>	SENT module base pointer
in	<i>Step</i>	divider value

Returns

void

Definition at line 465 of file AC784xx_Sent_Reg.h.

4.2.3.45 Sent_Reg_SetSerialMsgType()

```
LOCAL_INLINE void Sent_Reg_SetSerialMsgType (
    SENT_CHANNEL_Type * Base,
    Sent_SerialMsgType Type )
```

SENT set serial message type.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Type</i>	serial message type

Returns

void

Definition at line 540 of file AC784xx_Sent_Reg.h.

4.2.3.46 Sent_Reg_SetSPCParams()

```
LOCAL_INLINE void Sent_Reg_SetSPCParams (
    SENT_CHANNEL_Type * Base,
    uint32 Param )
```

SENT set spc parameters.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Param</i>	SPC control params

Returns

void

Definition at line 738 of file AC784xx_Sent_Reg.h.

4.2.3.47 Sent_Reg_SetSPCPulseDelay()

```
LOCAL_INLINE void Sent_Reg_SetSPCPulseDelay (
    SENT_CHANNEL_Type * Base,
    uint8 Delay )
```

SENT set SPC pulse delay.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Delay</i>	sent SPC gap between messages, in clock cycle unit time

Returns

void

Definition at line 708 of file AC784xx_Sent_Reg.h.

4.2.3.48 Sent_Reg_SetSPCPulseLength()

```
LOCAL_INLINE void Sent_Reg_SetSPCPulseLength (
    SENT_CHANNEL_Type * Base,
    uint8 Length )
```

SENT set SPC pulse length.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Length</i>	pulse length, in tick times

Returns

none

Definition at line 693 of file AC784xx_Sent_Reg.h.

4.2.3.49 Sent_Reg_SetSPCTimebase()

```
LOCAL_INLINE void Sent_Reg_SetSPCTimebase (
    SENT_CHANNEL_Type * Base,
    Sent_SpcTimeBaseType Timebase )
```

SENT set SPC pulse timebase type.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Timebase</i>	timebase type

Returns

void

Definition at line 678 of file AC784xx_Sent_Reg.h.

4.2.3.50 Sent_Reg_SetSPCTriggerMode()

```
LOCAL_INLINE void Sent_Reg_SetSPCTriggerMode (
    SENT_CHANNEL_Type * Base,
    Sent_SpcTriggerModeType Mode )
```

SENT set SPC pulse trigger mode.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Mode</i>	trigger mode

Returns

void

Definition at line 723 of file AC784xx_Sent_Reg.h.

4.2.3.51 Sent_Reg_SetWatchdogLimit()

```
LOCAL_INLINE void Sent_Reg_SetWatchdogLimit (
    SENT_CHANNEL_Type * Base,
    uint16 Limit )
```

SENT set watchdog limit.

Parameters

in	<i>Base</i>	SENT channel base pointer
in	<i>Limit</i>	watchdog limit value

Returns

void

Definition at line 806 of file AC784xx_Sent_Reg.h.

4.3 Sent_Hal.c File Reference

This file provides sent integration functions.

```
#include "AC784xx_Sent_Reg.h"
#include "Sent_Hal.h"
#include "Ckgen_Hal.h"
#include "Rcm_Hal.h"
#include "Core_Hal.h"
```

Classes

- struct [Sent_StateType](#)
ATC SENT runtime status.

Macros

- #define [IRQ_RSI](#) (0x1UL)
- #define [IRQ_RDI](#) (0x2UL)
- #define [IRQ_RFI](#) (0x4UL)
- #define [IRQ_TDI](#) (0x8UL)
- #define [IRQ_TBI](#) (0x10UL)
- #define [IRQ_FRI](#) (0x20UL)
- #define [IRQ_FDI](#) (0x40UL)
- #define [IRQ_NNI](#) (0x80UL)
- #define [IRQ_NVI](#) (0x100UL)

- `#define IRQ_CRCI` (0x200UL)
- `#define IRQ_WSI` (0x400UL)
- `#define IRQ_SDI` (0x800UL)
- `#define IRQ_SCRI` (0x1000UL)
- `#define IRQ_WDI` (0x2000UL)
- `#define IRQ_ESI` (0x4000UL)
- `#define IRQ_HFLL` (0x8000UL)
- `#define IRQ_FDONE` (0x10000UL)
- `#define IRQ_RECEIVE` (IRQ_RSI | IRQ_RDI | IRQ_SDI | IRQ_HFLL | IRQ_FDONE)
- `#define IRQ_TRANSMIT` (IRQ_TDI)
- `#define IRQ_ERROR`

Functions

- void `Sent_Hal_GetDefaultChannelConfig` (`Sent_ChannelConfigType` *SentChannelConfig)
Initializes the SENT channel configuration structure with default values.
- void `Sent_Hal_GetDefaultModuleConfig` (`Sent_ModuleConfigType` *SentModuleConfig)
Initializes the SENT module configuration structure with default values.
- void `Sent_Hal_Init` (const `Sent_ModuleConfigType` *SentModuleConfig)
initializes the SENT module
- void `Sent_Hal_Deinit` (void)
deinitializes the SENT module
- void `Sent_Hal_ConfigChannel` (`Sent_ChannelIdType` ChannelId, const `Sent_ChannelConfigType` *SentChannelConfig)
Set the SENT channel configuration.
- void `Sent_Hal_EnableChannel` (`Sent_ChannelIdType` ChannelId, boolean Enable)
enable/disable SENT channel
- void `Sent_Hal_InitChannelTxCtrl` (`Sent_ChannelIdType` ChannelId, const `Sent_TransmitCtrlType` *SentChannelTxCtrl)
Initializes the SENT channel Tx configuration.
- `Sent_ChannelStatusType` `Sent_Hal_GetChannelStatus` (`Sent_ChannelIdType` ChannelId)
SENT get channel state.
- `ISR` (SENT_Channel0_IRQHandler)
SENT channel 0 interrupts handler.
- `ISR` (SENT_Channel1_IRQHandler)
SENT channel 1 interrupts handler.

4.3.1 Detailed Description

This file provides sent integration functions.

4.3.2 Macro Definition Documentation

4.3.2.1 IRQ_CRCI

```
#define IRQ_CRCI (0x200UL)
```

Definition at line 64 of file Sent_Hal.c.

4.3.2.2 IRQ_ERROR

```
#define IRQ_ERROR
```

Value:

```
(IRQ_RFI | IRQ_TBI | IRQ_FRI | IRQ_FDI | IRQ_NNI |  
      | IRQ_NVI | IRQ_CRCI | IRQ_WSI |  
      IRQ_SCRI | IRQ_WDI | IRQ_ESI)
```

Definition at line 75 of file Sent_Hal.c.

4.3.2.3 IRQ_ESI

```
#define IRQ_ESI (0x4000UL)
```

Definition at line 69 of file Sent_Hal.c.

4.3.2.4 IRQ_FDI

```
#define IRQ_FDI (0x40UL)
```

Definition at line 61 of file Sent_Hal.c.

4.3.2.5 IRQ_FDONE

```
#define IRQ_FDONE (0x10000UL)
```

Definition at line 71 of file Sent_Hal.c.

4.3.2.6 IRQ_FRI

```
#define IRQ_FRI (0x20UL)
```

Definition at line 60 of file Sent_Hal.c.

4.3.2.7 IRQ_HFLL

```
#define IRQ_HFLL (0x8000UL)
```

Definition at line 70 of file Sent_Hal.c.

4.3.2.8 IRQ_NNI

```
#define IRQ_NNI (0x80UL)
```

Definition at line 62 of file Sent_Hal.c.

4.3.2.9 IRQ_NVI

```
#define IRQ_NVI (0x100UL)
```

Definition at line 63 of file Sent_Hal.c.

4.3.2.10 IRQ_RDI

```
#define IRQ_RDI (0x2UL)
```

Definition at line 56 of file Sent_Hal.c.

4.3.2.11 IRQ_RECEIVE

```
#define IRQ_RECEIVE (IRQ_RSI | IRQ_RDI | IRQ_SDI | IRQ_HFLL | IRQ_FDONE)
```

Definition at line 73 of file Sent_Hal.c.

4.3.2.12 IRQ_RFI

```
#define IRQ_RFI (0x4UL)
```

Definition at line 57 of file Sent_Hal.c.

4.3.2.13 IRQ_RSI

```
#define IRQ_RSI (0x1UL)
```

Definition at line 55 of file Sent_Hal.c.

4.3.2.14 IRQ_SCRI

```
#define IRQ_SCRI (0x1000UL)
```

Definition at line 67 of file Sent_Hal.c.

4.3.2.15 IRQ_SDI

```
#define IRQ_SDI (0x800UL)
```

Definition at line 66 of file Sent_Hal.c.

4.3.2.16 IRQ_TBI

```
#define IRQ_TBI (0x10UL)
```

Definition at line 59 of file Sent_Hal.c.

4.3.2.17 IRQ_TDI

```
#define IRQ_TDI (0x8UL)
```

Definition at line 58 of file Sent_Hal.c.

4.3.2.18 IRQ_TRANSMIT

```
#define IRQ_TRANSMIT (IRQ\_TDI)
```

Definition at line 74 of file Sent_Hal.c.

4.3.2.19 IRQ_WDI

```
#define IRQ_WDI (0x2000UL)
```

Definition at line 68 of file Sent_Hal.c.

4.3.2.20 IRQ_WSI

```
#define IRQ_WSI (0x400UL)
```

Definition at line 65 of file Sent_Hal.c.

4.3.3 Function Documentation

4.3.3.1 ISR() [1/2]

```
ISR (
    SENT_Channel0_IRQHandler )
```

SENT channel 0 interrupts handler.

Returns

void

Definition at line 436 of file Sent_Hal.c.

4.3.3.2 ISR() [2/2]

```
ISR (
    SENT_Channel1_IRQHandler )
```

SENT channel 1 interrupts handler.

Returns

void

Definition at line 446 of file Sent_Hal.c.

4.3.3.3 Sent_Hal_ConfigChannel()

```
void Sent_Hal_ConfigChannel (
    Sent_ChannelIdType ChannelId,
    const Sent_ChannelConfigType * SentChannelConfig )
```

Set the SENT channel configuration.

Parameters

in	<i>ChannelId</i>	sent channel ID
in	<i>SentChannelConfig</i>	sent channel config pointer

Returns

void

Definition at line 354 of file Sent_Hal.c.

4.3.3.4 Sent_Hal_Deinit()

```
void Sent_Hal_Deinit (  
    void )
```

deinitializes the SENT module

Returns

void

Definition at line 330 of file Sent_Hal.c.

4.3.3.5 Sent_Hal_EnableChannel()

```
void Sent_Hal_EnableChannel (  
    Sent_ChannelIdType ChannelId,  
    boolean Enable )
```

enable/disable SENT channel

Parameters

in	<i>ChannelId</i>	sent channel ID
in	<i>Enable</i>	enable or disable

Returns

void

Definition at line 388 of file Sent_Hal.c.

4.3.3.6 Sent_Hal_GetChannelStatus()

```
Sent_ChannelStatusType Sent_Hal_GetChannelStatus (
    Sent_ChannelIdType ChannelId )
```

SENT get channel state.

Parameters

in	<i>ChannelId</i>	sent channel Id
----	------------------	-----------------

Returns

channel status

Definition at line 423 of file Sent_Hal.c.

4.3.3.7 Sent_Hal_GetDefaultChannelConfig()

```
void Sent_Hal_GetDefaultChannelConfig (
    Sent_ChannelConfigType * SentChannelConfig )
```

Initializes the SENT channel configuration structure with default values.

Parameters

out	<i>SentChannelConfig</i>	Initializes a configuration structure received from the application with default values
-----	--------------------------	---

Returns

void

Definition at line 281 of file Sent_Hal.c.

4.3.3.8 Sent_Hal_GetDefaultModuleConfig()

```
void Sent_Hal_GetDefaultModuleConfig (
    Sent_ModuleConfigType * SentModuleConfig )
```

Initializes the SENT module configuration structure with default values.

Parameters

out	<i>SentModuleConfig</i>	Initializes a configuration structure received from the application with default values
-----	-------------------------	---

Returns

void

Definition at line 293 of file Sent_Hal.c.

4.3.3.9 Sent_Hal_Init()

```
void Sent_Hal_Init (
    const Sent\_ModuleConfigType * SentModuleConfig )
```

initializes the SENT module

Parameters

in	<i>SentModuleConfig</i>	sent module config pointer
----	-------------------------	----------------------------

Returns

void

Definition at line 305 of file Sent_Hal.c.

4.3.3.10 Sent_Hal_InitChannelTxCtrl()

```
void Sent_Hal_InitChannelTxCtrl (
    Sent\_ChannelIdType ChannelId,
    const Sent\_TransmitCtrlType * SentChannelTxConfig )
```

Initializes the SENT channel Tx configuration.

Parameters

in	<i>ChannelId</i>	sent channel Id
in	<i>SentChannelTxConfig</i>	sent channel Tx configuration

Returns

void

Definition at line 402 of file Sent_Hal.c.

4.4 Sent_Hal.h File Reference

This file provides sent integration functions interface.

```
#include "Sent_Hal_Types.h"
```

Functions

- void [Sent_Hal_GetDefaultChannelConfig](#) ([Sent_ChannelConfigType](#) *SentChannelConfig)
Initializes the SENT channel configuration structure with default values.
- void [Sent_Hal_GetDefaultModuleConfig](#) ([Sent_ModuleConfigType](#) *SentModuleConfig)
Initializes the SENT module configuration structure with default values.
- void [Sent_Hal_Init](#) (const [Sent_ModuleConfigType](#) *SentModuleConfig)
initializes the SENT module
- void [Sent_Hal_Deinit](#) (void)
deinitializes the SENT module
- void [Sent_Hal_ConfigChannel](#) ([Sent_ChannelIdType](#) ChannelId, const [Sent_ChannelConfigType](#) *SentChannel↵
Config)
Set the SENT channel configuration.
- void [Sent_Hal_EnableChannel](#) ([Sent_ChannelIdType](#) ChannelId, boolean Enable)
enable/disable SENT channel
- [Sent_ChannelStatusType](#) [Sent_Hal_GetChannelStatus](#) ([Sent_ChannelIdType](#) ChannelId)
SENT get channel state.
- void [Sent_Hal_InitChannelTxCtrl](#) ([Sent_ChannelIdType](#) ChannelId, const [Sent_TransmitCtrlType](#) *SentChannelTx↵
Config)
Initializes the SENT channel Tx configuration.

4.4.1 Detailed Description

This file provides sent integration functions interface.

4.4.2 Function Documentation

4.4.2.1 Sent_Hal_ConfigChannel()

```
void Sent_Hal_ConfigChannel (
    Sent\_ChannelIdType ChannelId,
    const Sent\_ChannelConfigType * SentChannelConfig )
```

Set the SENT channel configuration.

Parameters

in	<i>ChannelId</i>	sent channel ID
in	<i>SentChannelConfig</i>	sent channel config pointer

Returns

void

Definition at line 354 of file Sent_Hal.c.

4.4.2.2 Sent_Hal_Deinit()

```
void Sent_Hal_Deinit (
    void )
```

deinitializes the SENT module

Returns

void

Definition at line 330 of file Sent_Hal.c.

4.4.2.3 Sent_Hal_EnableChannel()

```
void Sent_Hal_EnableChannel (
    Sent_ChannelIdType ChannelId,
    boolean Enable )
```

enable/disable SENT channel

Parameters

in	<i>ChannelId</i>	sent channel ID
in	<i>Enable</i>	enable or disable

Returns

void

Definition at line 388 of file Sent_Hal.c.

4.4.2.4 Sent_Hal_GetChannelStatus()

```
Sent_ChannelStatusType Sent_Hal_GetChannelStatus (
    Sent_ChannelIdType ChannelId )
```

SENT get channel state.

Parameters

in	<i>ChannelId</i>	sent channel ID
----	------------------	-----------------

Returns

channel status

Parameters

in	<i>Channel↔ Id</i>	sent channel Id
----	------------------------	-----------------

Returns

channel status

Definition at line 423 of file Sent_Hal.c.

4.4.2.5 Sent_Hal_GetDefaultChannelConfig()

```
void Sent_Hal_GetDefaultChannelConfig (
    Sent_ChannelConfigType * SentChannelConfig )
```

Initializes the SENT channel configuration structure with default values.

Parameters

out	<i>SentChannelConfig</i>	Initializes a configuration structure received from the application with default values
-----	--------------------------	---

Returns

void

Definition at line 281 of file Sent_Hal.c.

4.4.2.6 Sent_Hal_GetDefaultModuleConfig()

```
void Sent_Hal_GetDefaultModuleConfig (
    Sent_ModuleConfigType * SentModuleConfig )
```

Initializes the SENT module configuration structure with default values.

Parameters

out	<i>SentModuleConfig</i>	Initializes a configuration structure received from the application with default values
-----	-------------------------	---

Returns

void

Definition at line 293 of file Sent_Hal.c.

4.4.2.7 Sent_Hal_Init()

```
void Sent_Hal_Init (
    const Sent_ModuleConfigType * SentModuleConfig )
```

initializes the SENT module

Parameters

in	<i>SentModuleConfig</i>	sent module config pointer
----	-------------------------	----------------------------

Returns

void

Definition at line 305 of file Sent_Hal.c.

4.4.2.8 Sent_Hal_InitChannelTxCtrl()

```
void Sent_Hal_InitChannelTxCtrl (
    Sent_ChannelIdType ChannelId,
    const Sent_TransmitCtrlType * SentChannelTxConfig )
```

Initializes the SENT channel Tx configuration.

Parameters

in	<i>ChannelId</i>	sent channel ID
in	<i>SentChannelTxConfig</i>	sent channel Tx configuration

Returns

void

Parameters

in	<i>ChannelId</i>	sent channel Id
in	<i>SentChannelTxConfig</i>	sent channel Tx configuration

Returns

void

Definition at line 402 of file Sent_Hal.c.

4.5 Sent_Hal_Types.h File Reference

```
#include "Device_Register.h"
```


Classes

- struct [Sent_FifoStatusType](#)
Define the sent fifo status.
- struct [Sent_SerialFrameType](#)
Define the sent serial frame.
- struct [Sent_FrameDataType](#)
Define the sent frame data.
- struct [Sent_MsgType](#)
Define the sent message.
- struct [Sent_IoCtrlType](#)
Define the sent channel io control config.
- struct [Sent_NibblePointerType](#)
Define the sent nibble position config.
- struct [Sent_ReceiveCtrlType](#)
Define the sent channel receive config.
- struct [Sent_TransmitCtrlType](#)
Define the sent channel transmit config.
- struct [Sent_ModuleConfigType](#)
SENT module configuration structure.
- struct [Sent_ChannelConfigType](#)
SENT channel configuration structure.

Typedefs

- typedef void(* [Sent_CallbackType](#)) ([Sent_ChannelIdType](#) ChannelId, uint32 Event)
- typedef void(* [Sent_RxCallbackType](#)) ([Sent_ChannelIdType](#) ChannelId, uint32 Event, [Sent_MsgType](#) *Msg)

Enumerations

- enum [Sent_ChannelIdType](#) { [SENT_CHANNEL_0](#) = 0x00U, [SENT_CHANNEL_1](#) = 0x01U }
Define the sent channel ID.
- enum [Sent_FrameCheckModeType](#) { [SENT_FRAME_CHECK_PAST_SYNC_PULSE](#) = 0x00U, [SENT_FRAME_CHECK_FUTURE_SYNC_PULSE](#) = 0x01U }
Define the sent frame check type.
- enum [Sent_SerialMsgType](#) { [SENT_SHORT_SERIAL_MSG](#) = 0x00U, [SENT_ENHANCED_SERIAL_MSG](#) = 0x01U }
Define the sent serial msg type.
- enum [Sent_ChannelStatusType](#) { [SENT_CHANNEL_STOP](#) = 0x00U, [SENT_CHANNEL_INITIALIZED](#) = 0x01U, [SENT_CHANNEL_RUNNING](#) = 0x02U, [SENT_CHANNEL_SYNCHRONIZED](#) = 0x03U }
Define the sent channel state.
- enum [Sent_EnhancedConfigType](#) { [SENT_ENHANCED_CONFIG_0](#) = 0x00U, [SENT_ENHANCED_CONFIG_1](#) = 0x01U }
Define the configuration bit type of enhanced serial msg.
- enum [Sent_NibblePositonType](#) { [SENT_NIBBLE_POSITION_0](#) = 0x00U, [SENT_NIBBLE_POSITION_1](#) = 0x01U, [SENT_NIBBLE_POSITION_2](#) = 0x02U, [SENT_NIBBLE_POSITION_3](#) = 0x03U, [SENT_NIBBLE_POSITION_4](#) = 0x04U, [SENT_NIBBLE_POSITION_5](#) = 0x05U, [SENT_NIBBLE_POSITION_6](#) = 0x06U, [SENT_NIBBLE_POSITION_7](#) = 0x07U }
Define the configuration bit type of enhanced serial msg.
- enum [Sent_InputDataSelectType](#) { [SENT_INPUT_DATA_SELECT_0](#) = 0x00U, [SENT_INPUT_DATA_SELECT_1](#) = 0x01U }
Define the input source select.

- enum `Sent_ExternalTriggerType` { `SENT_EXTERNAL_TRIGGER_0` = 0x00U, `SENT_EXTERNAL_TRIGGER_1` = 0x01U, `SENT_EXTERNAL_TRIGGER_2` = 0x02U, `SENT_EXTERNAL_TRIGGER_3` = 0x03U }
Define the trigger source select.
- enum `Sent_SpcTimeBaseType` { `SENT_LAST_SYNC` = 0x00U, `SENT_NOMINAL` = 0x01U }
Define the spc time base.
- enum `Sent_SpcTriggerModeType` { `SENT_SPC_TRIGGER_OFF` = 0x00U, `SENT_SPC_TRIGGER_IMMEDIATE` = 0x01U, `SENT_SPC_TRIGGER_FALLING_EDGE` = 0x02U, `SENT_SPC_TRIGGER_EXTERNAL` = 0x03U }
Define the spc trigger mode.
- enum {
 `SENT_EVENT_RECEIVE_SUCCESS` = 0x1UL, `SENT_EVENT_RECEIVE_DATA` = 0x2UL, `SENT_EVENT_RECEIVE_SERIAL_MSG` = 0x4UL, `SENT_EVENT_RECEIVE_FRAME_DONE` = 0x8UL,
 `SENT_EVENT_RECEIVE_FIFO_HALF_FULL` = 0x10UL, `SENT_EVENT_TRANSMIT_DATA` = 0x20UL, `SENT_EVENT_ERROR_RECEIVE_FIFO_OVERFLOW` = 0x40UL, `SENT_EVENT_ERROR_TRANSMIT_BUFFER_UNDERFLOW` = 0x80UL,
 `SENT_EVENT_ERROR_FREQUENCY_RANGE` = 0x100UL, `SENT_EVENT_ERROR_FREQUENCY_DRIFT` = 0x200UL, `SENT_EVENT_ERROR_NIBBLE_NUMBER` = 0x400UL, `SENT_EVENT_ERROR_NIBBLE_DATA` = 0x800UL,
 `SENT_EVENT_ERROR_NIBBLE_CRC` = 0x1000UL, `SENT_EVENT_ERROR_SERIAL_CRC` = 0x2000UL, `SENT_EVENT_ERROR_SHORT_START_BIT` = 0x4000UL, `SENT_EVENT_ERROR_ENHANCED_START_BIT` = 0x8000UL,
 `SENT_EVENT_ERROR_WATCH_DOG` = 0x10000UL }
SENT channel interrupt event.

4.5.1 Typedef Documentation

4.5.1.1 Sent_CallbackType

```
typedef void(* Sent_CallbackType) (Sent_ChannelIdType ChannelId, uint32 Event)
```

< Callback for SENT module Callback for SENT module fast data

Definition at line 233 of file Sent_Hal_Types.h.

4.5.1.2 Sent_RxCallbackType

```
typedef void(* Sent_RxCallbackType) (Sent_ChannelIdType ChannelId, uint32 Event, Sent_MsgType *Msg)
```

Definition at line 236 of file Sent_Hal_Types.h.

4.5.2 Enumeration Type Documentation

4.5.2.1 anonymous enum

anonymous enum

SENT channel interrupt event.

Enumerator

SENT_EVENT_RECEIVE_SUCCESS	Rx CRC checked correct
SENT_EVENT_RECEIVE_DATA	Rx data nibble done
SENT_EVENT_RECEIVE_SERIAL_MSG	Rx serial message done
SENT_EVENT_RECEIVE_FRAME_DONE	Rx fast channel frame done including pause pulse
SENT_EVENT_RECEIVE_FIFO_HALF_FULL	Rx fifo is half full
SENT_EVENT_TRANSMIT_DATA	Tx buffer done
SENT_EVENT_ERROR_RECEIVE_FIFO_OVERFLOW	Rx fifo overflow
SENT_EVENT_ERROR_TRANSMIT_BUFFER_UNDERFLOW	Tx buffer underflow
SENT_EVENT_ERROR_FREQUENCY_RANGE	frequency range error
SENT_EVENT_ERROR_FREQUENCY_DRIFT	frequency drift error
SENT_EVENT_ERROR_NIBBLE_NUMBER	nibble number error
SENT_EVENT_ERROR_NIBBLE_DATA	nibble value error
SENT_EVENT_ERROR_NIBBLE_CRC	crc nibble error
SENT_EVENT_ERROR_SERIAL_CRC	serial crc error
SENT_EVENT_ERROR_SHORT_START_BIT	short serial start bit error
SENT_EVENT_ERROR_ENHANCED_START_BIT	enhanced serial start bit error
SENT_EVENT_ERROR_WATCH_DOG	watchdog timeout error

Definition at line 163 of file Sent_Hal_Types.h.

4.5.2.2 Sent_ChannelIdType

```
enum Sent_ChannelIdType
```

Define the sent channel ID.

Enumerator

SENT_CHANNEL↔ _0	channel 0
SENT_CHANNEL↔ _1	channel 1

Definition at line 56 of file Sent_Hal_Types.h.

4.5.2.3 Sent_ChannelStatusType

```
enum Sent_ChannelStatusType
```

Define the sent channel state.

Enumerator

SENT_CHANNEL_STOP	channel stopped
SENT_CHANNEL_INITIALIZED	channel initialized, means configured and enabled
SENT_CHANNEL_RUNNING	channel running, fallback status from synchronized
SENT_CHANNEL_SYNCHRONIZED	channel synchronized, frequency in range

Definition at line 87 of file Sent_Hal_Types.h.

4.5.2.4 Sent_EnhancedConfigType

```
enum Sent_EnhancedConfigType
```

Define the configuration bit type of enhanced serial msg.

Enumerator

SENT_ENHANCED_CONFIG↔ _0	enhanced serial message config bit equals to 0
SENT_ENHANCED_CONFIG↔ _1	enhanced serial message config bit equals to 1

Definition at line 98 of file Sent_Hal_Types.h.

4.5.2.5 Sent_ExternalTriggerType

```
enum Sent_ExternalTriggerType
```

Define the trigger source select.

Enumerator

SENT_EXTERNAL_TRIGGER↔ _0	external trigger 0
SENT_EXTERNAL_TRIGGER↔ _1	external trigger 1
SENT_EXTERNAL_TRIGGER↔ _2	external trigger 2
SENT_EXTERNAL_TRIGGER↔ _3	external trigger 3

Definition at line 131 of file Sent_Hal_Types.h.

4.5.2.6 Sent_FrameCheckModeType

```
enum Sent_FrameCheckModeType
```

Define the sent frame check type.

Enumerator

SENT_FRAME_CHECK_PAST_SYNC_PULSE	frame check against past sync pulse
SENT_FRAME_CHECK_FUTURE_SYNC_PULSE	frame check against future sync pulse

Definition at line 69 of file Sent_Hal_Types.h.

4.5.2.7 Sent_InputDataSelectType

```
enum Sent_InputDataSelectType
```

Define the input source select.

Enumerator

SENT_INPUT_DATA_SELECT↔ _0	input data from source 0
SENT_INPUT_DATA_SELECT↔ _1	input data from source 1

Definition at line 122 of file Sent_Hal_Types.h.

4.5.2.8 Sent_NibblePositonType

```
enum Sent_NibblePositonType
```

Define the configuration bit type of enhanced serial msg.

Enumerator

SENT_NIBBLE_POSITION↔ _0	store data nibble in position 0
SENT_NIBBLE_POSITION↔ _1	store data nibble in position 1
SENT_NIBBLE_POSITION↔ _2	store data nibble in position 2
SENT_NIBBLE_POSITION↔ _3	store data nibble in position 3
SENT_NIBBLE_POSITION↔ _4	store data nibble in position 4
SENT_NIBBLE_POSITION↔ _5	store data nibble in position 5
SENT_NIBBLE_POSITION↔ _6	store data nibble in position 6
SENT_NIBBLE_POSITION↔ _7	store data nibble in position 7

Definition at line 107 of file Sent_Hal_Types.h.

4.5.2.9 Sent_SerialMsgType

```
enum Sent_SerialMsgType
```

Define the sent serial msg type.

Enumerator

SENT_SHORT_SERIAL_MSG	short serial message type
SENT_ENHANCED_SERIAL_MSG	enhanced serial message type

Definition at line 78 of file Sent_Hal_Types.h.

4.5.2.10 Sent_SpcTimeBaseType

enum [Sent_SpcTimeBaseType](#)

Define the spc time base.

Enumerator

SENT_LAST_SYNC	based on last received sync frequency
SENT_NOMINAL	based on nominal frequency

Definition at line 142 of file Sent_Hal_Types.h.

4.5.2.11 Sent_SpcTriggerModeType

enum [Sent_SpcTriggerModeType](#)

Define the spc trigger mode.

Enumerator

SENT_SPC_TRIGGER_OFF	no pulse generated
SENT_SPC_TRIGGER_IMMEDIATE	pulse starts immediately
SENT_SPC_TRIGGER_FALLING_EDGE	pulse starts each time the first falling edge of calibration pulse
SENT_SPC_TRIGGER_EXTERNAL	pulse starts after external trigger

Definition at line 151 of file Sent_Hal_Types.h.

Index

AC784xx_API_Reference_Manual_SENT.pdf, [21](#)

AC784xx_Sent_Reg.h, [21](#)

IRQ_STATUS_MASK, [23](#)

Sent_Reg_ClearIrqStatus, [24](#)

Sent_Reg_DisableDriftError, [24](#)

Sent_Reg_EnableAlternateCRC, [25](#)

Sent_Reg_EnableCRCAugumentZero, [26](#)

Sent_Reg_EnableChannel, [25](#)

Sent_Reg_EnableFDFLMode, [26](#)

Sent_Reg_EnableFastMsgCRC, [26](#)

Sent_Reg_EnableFlushFifo, [27](#)

Sent_Reg_EnableInputInvert, [27](#)

Sent_Reg_EnableIrq, [28](#)

Sent_Reg_EnableOutputInvert, [28](#)

Sent_Reg_EnablePausePulse, [28](#)

Sent_Reg_EnableSerialMsgCRC, [29](#)

Sent_Reg_EnableSerialMsgProcess, [29](#)

Sent_Reg_EnableStatusNibbleInCRC, [30](#)

Sent_Reg_GetBase, [30](#)

Sent_Reg_GetChannelStatus, [30](#)

Sent_Reg_GetCurrentTimeStamp, [31](#)

Sent_Reg_GetEnhancedConfigType, [31](#)

Sent_Reg_GetFifoAvailable, [31](#)

Sent_Reg_GetFifoEmptyStatus, [32](#)

Sent_Reg_GetFifoFullStatus, [32](#)

Sent_Reg_GetFifoHalfFullStatus, [33](#)

Sent_Reg_GetFrameCRCNibble, [33](#)

Sent_Reg_GetFrameDataNibbles, [33](#)

Sent_Reg_GetFrameLengthCounter, [34](#)

Sent_Reg_GetFrameStatusNibble, [34](#)

Sent_Reg_GetIrqStatus, [34](#)

Sent_Reg_GetModuleClkDivider, [35](#)

Sent_Reg_GetSerialMsgCRC, [35](#)

Sent_Reg_GetSerialMsgData, [36](#)

Sent_Reg_GetSerialMsgID, [36](#)

Sent_Reg_GetSpcTransferState, [36](#)

Sent_Reg_SelectExternalTriggerSource, [37](#)

Sent_Reg_SelectInputDataSource, [37](#)

Sent_Reg_SetChannelFractionalDivider, [38](#)

Sent_Reg_SetChannelPreDivider, [38](#)

Sent_Reg_SetDataNibbleView, [38](#)

Sent_Reg_SetFrameCheckMode, [39](#)

Sent_Reg_SetFrameLength, [39](#)

Sent_Reg_SetGlitchFilterDepth, [40](#)

Sent_Reg_SetIrqStatus, [40](#)

Sent_Reg_SetModuleClkDivider, [40](#)

Sent_Reg_SetModuleTimeStampDivider, [41](#)

Sent_Reg_SetSPCParams, [41](#)

Sent_Reg_SetSPCPulseDelay, [42](#)

Sent_Reg_SetSPCPulseLength, [42](#)

Sent_Reg_SetSPCTimebase, [43](#)

Sent_Reg_SetSPCTriggerMode, [43](#)

Sent_Reg_SetSerialMsgType, [41](#)

Sent_Reg_SetWatchdogLimit, [44](#)

AlternateCRCEnable

Sent_ReceiveCtrlType, [14](#)

ConfigBit

Sent_SerialFrameType, [17](#)

Data

Sent_FrameDataType, [7](#)

DataFrame

Sent_MsgType, [11](#)

DriftErrorDisable

Sent_ReceiveCtrlType, [14](#)

EnabledInterrupts

Sent_ChannelConfigType, [3](#)

Sent_StateType, [19](#)

ErrCallback

Sent_ChannelConfigType, [3](#)

Sent_StateType, [19](#)

FDFL

Sent_ReceiveCtrlType, [15](#)

FastCRCEnable

Sent_ReceiveCtrlType, [15](#)

FifoStatus

Sent_MsgType, [11](#)

FractionDivider

Sent_ModuleConfigType, [10](#)

FrameCheckMode

Sent_ReceiveCtrlType, [15](#)

FrameDataCRC

Sent_FrameDataType, [7](#)

FrameDataFifoAvail

Sent_FifoStatusType, [6](#)

FrameDataFifoEmpty

Sent_FifoStatusType, [6](#)

FrameDataFifoFull

Sent_FifoStatusType, [6](#)

FrameDataFifoHalfFull

Sent_FifoStatusType, [6](#)

FrameLength

Sent_ReceiveCtrlType, [15](#)

GlitchFilterDepth

Sent_IoCtrlType, [8](#)

IRQ_CRCl

Sent_Hal.c, [45](#)

IRQ_ERROR

Sent_Hal.c, [45](#)

IRQ_ESI

- Sent_Hal.c, 46
- IRQ_FDONE
 - Sent_Hal.c, 46
- IRQ_FDI
 - Sent_Hal.c, 46
- IRQ_FRI
 - Sent_Hal.c, 46
- IRQ_HFLL
 - Sent_Hal.c, 46
- IRQ_NNI
 - Sent_Hal.c, 46
- IRQ_NVI
 - Sent_Hal.c, 47
- IRQ_RDI
 - Sent_Hal.c, 47
- IRQ_RECEIVE
 - Sent_Hal.c, 47
- IRQ_RFI
 - Sent_Hal.c, 47
- IRQ_RSI
 - Sent_Hal.c, 47
- IRQ_SCRI
 - Sent_Hal.c, 47
- IRQ_SDI
 - Sent_Hal.c, 48
- IRQ_STATUS_MASK
 - AC784xx_Sent_Reg.h, 23
- IRQ_TBI
 - Sent_Hal.c, 48
- IRQ_TDI
 - Sent_Hal.c, 48
- IRQ_TRANSMIT
 - Sent_Hal.c, 48
- IRQ_WDI
 - Sent_Hal.c, 48
- IRQ_WSI
 - Sent_Hal.c, 48
- ISR
 - Sent_Hal.c, 49
- Input
 - Sent_IoCtrlType, 9
- InputInvertEnable
 - Sent_IoCtrlType, 9
- IoControl
 - Sent_ChannelConfigType, 4
- Mode
 - Sent_TransmitCtrlType, 20
- Nibble0Position
 - Sent_NibblePointerType, 12
- Nibble1Position
 - Sent_NibblePointerType, 12
- Nibble2Position
 - Sent_NibblePointerType, 12
- Nibble3Position
 - Sent_NibblePointerType, 13
- Nibble4Position
 - Sent_NibblePointerType, 13
- Nibble5Position
 - Sent_NibblePointerType, 13
- Nibble6Position
 - Sent_NibblePointerType, 13
- Nibble7Position
 - Sent_NibblePointerType, 13
- NibblePointer
 - Sent_ChannelConfigType, 4
- OutputInvertEnable
 - Sent_IoCtrlType, 9
- PausePulseEnable
 - Sent_ReceiveCtrlType, 15
- PulseDelayLength
 - Sent_TransmitCtrlType, 20
- PulseLength
 - Sent_TransmitCtrlType, 20
- RxCallback
 - Sent_ChannelConfigType, 4
 - Sent_StateType, 19
- RxControl
 - Sent_ChannelConfigType, 4
- Sent_CallbackType
 - Sent_Hal_Types.h, 58
- Sent_ChannelConfigType, 3
 - EnabledInterrupts, 3
 - ErrCallback, 3
 - IoControl, 4
 - NibblePointer, 4
 - RxCallback, 4
 - RxControl, 4
 - TickUnit, 4
 - TxCallback, 5
 - WatchdogTimerLimit, 5
- Sent_ChannelIdType
 - Sent_Hal_Types.h, 59
- Sent_ChannelStatusType
 - Sent_Hal_Types.h, 59
- Sent_EnhancedConfigType
 - Sent_Hal_Types.h, 60
- Sent_ExternalTriggerType
 - Sent_Hal_Types.h, 60
- Sent_FifoStatusType, 5
 - FrameDataFifoAvail, 6
 - FrameDataFifoEmpty, 6
 - FrameDataFifoFull, 6
 - FrameDataFifoHalfFull, 6
- Sent_FrameCheckModeType
 - Sent_Hal_Types.h, 60
- Sent_FrameDataType, 6
 - Data, 7
 - FrameDataCRC, 7
 - StatusNibble, 7
 - TimeStamp, 7
 - TotalFrameLength, 8
- Sent_Hal.c, 44
 - IRQ_CRCI, 45
 - IRQ_ERROR, 45
 - IRQ_ESI, 46
 - IRQ_FDONE, 46
 - IRQ_FDI, 46

- IRQ_FRI, [46](#)
- IRQ_HFLL, [46](#)
- IRQ_NNI, [46](#)
- IRQ_NVI, [47](#)
- IRQ_RDI, [47](#)
- IRQ_RECEIVE, [47](#)
- IRQ_RFI, [47](#)
- IRQ_RSI, [47](#)
- IRQ_SCRI, [47](#)
- IRQ_SDI, [48](#)
- IRQ_TBI, [48](#)
- IRQ_TDI, [48](#)
- IRQ_TRANSMIT, [48](#)
- IRQ_WDI, [48](#)
- IRQ_WSI, [48](#)
- ISR, [49](#)
- Sent_Hal_ConfigChannel, [49](#)
- Sent_Hal_Deinit, [50](#)
- Sent_Hal_EnableChannel, [50](#)
- Sent_Hal_GetChannelStatus, [50](#)
- Sent_Hal_GetDefaultChannelConfig, [51](#)
- Sent_Hal_GetDefaultModuleConfig, [51](#)
- Sent_Hal_Init, [52](#)
- Sent_Hal_InitChannelTxCtrl, [52](#)
- Sent_Hal.h, [52](#)
 - Sent_Hal_ConfigChannel, [53](#)
 - Sent_Hal_Deinit, [53](#)
 - Sent_Hal_EnableChannel, [54](#)
 - Sent_Hal_GetChannelStatus, [54](#)
 - Sent_Hal_GetDefaultChannelConfig, [55](#)
 - Sent_Hal_GetDefaultModuleConfig, [55](#)
 - Sent_Hal_Init, [55](#)
 - Sent_Hal_InitChannelTxCtrl, [56](#)
- Sent_Hal_ConfigChannel
 - Sent_Hal.c, [49](#)
 - Sent_Hal.h, [53](#)
- Sent_Hal_Deinit
 - Sent_Hal.c, [50](#)
 - Sent_Hal.h, [53](#)
- Sent_Hal_EnableChannel
 - Sent_Hal.c, [50](#)
 - Sent_Hal.h, [54](#)
- Sent_Hal_GetChannelStatus
 - Sent_Hal.c, [50](#)
 - Sent_Hal.h, [54](#)
- Sent_Hal_GetDefaultChannelConfig
 - Sent_Hal.c, [51](#)
 - Sent_Hal.h, [55](#)
- Sent_Hal_GetDefaultModuleConfig
 - Sent_Hal.c, [51](#)
 - Sent_Hal.h, [55](#)
- Sent_Hal_Init
 - Sent_Hal.c, [52](#)
 - Sent_Hal.h, [55](#)
- Sent_Hal_InitChannelTxCtrl
 - Sent_Hal.c, [52](#)
 - Sent_Hal.h, [56](#)
- Sent_Hal_Types.h, [56](#)
 - Sent_CallbackType, [58](#)
 - Sent_ChannelIdType, [59](#)
 - Sent_ChannelStatusType, [59](#)
 - Sent_EnhancedConfigType, [60](#)
 - Sent_ExternalTriggerType, [60](#)
 - Sent_FrameCheckModeType, [60](#)
 - Sent_InputDataSelectType, [61](#)
 - Sent_NibblePositonType, [61](#)
 - Sent_RxCallbackType, [58](#)
 - Sent_SerialMsgType, [61](#)
 - Sent_SpcTimeBaseType, [62](#)
 - Sent_SpcTriggerModeType, [62](#)
- Sent_InputDataSelectType
 - Sent_Hal_Types.h, [61](#)
- Sent_IoCtrlType, [8](#)
 - GlitchFilterDepth, [8](#)
 - Input, [9](#)
 - InputInvertEnable, [9](#)
 - OutputInvertEnable, [9](#)
 - TriggerSource, [9](#)
- Sent_ModuleConfigType, [10](#)
 - FractionDivider, [10](#)
 - TimeStampPreDivider, [10](#)
- Sent_MsgType, [10](#)
 - DataFrame, [11](#)
 - FifoStatus, [11](#)
 - SerialFrame, [11](#)
- Sent_NibblePointerType, [12](#)
 - Nibble0Position, [12](#)
 - Nibble1Position, [12](#)
 - Nibble2Position, [12](#)
 - Nibble3Position, [13](#)
 - Nibble4Position, [13](#)
 - Nibble5Position, [13](#)
 - Nibble6Position, [13](#)
 - Nibble7Position, [13](#)
- Sent_NibblePositonType
 - Sent_Hal_Types.h, [61](#)
- Sent_ReceiveCtrlType, [14](#)
 - AlternateCRCEnable, [14](#)
 - DriftErrorDisable, [14](#)
 - FDFL, [15](#)
 - FastCRCEnable, [15](#)
 - FrameCheckMode, [15](#)
 - FrameLength, [15](#)
 - PausePulseEnable, [15](#)
 - SerialCRCEnable, [16](#)
 - SerialDataEnable, [16](#)
 - SerialMsgType, [16](#)
 - StatusIncludedInCRC, [16](#)
 - ZeroAugumentInCRCDisable, [16](#)
- Sent_Reg_ClearIrqStatus
 - AC784xx_Sent_Reg.h, [24](#)
- Sent_Reg_DisableDriftError
 - AC784xx_Sent_Reg.h, [24](#)
- Sent_Reg_EnableAlternateCRC
 - AC784xx_Sent_Reg.h, [25](#)
- Sent_Reg_EnableCRCAugumentZero
 - AC784xx_Sent_Reg.h, [26](#)
- Sent_Reg_EnableChannel
 - AC784xx_Sent_Reg.h, [25](#)
- Sent_Reg_EnableFDFLMode
 - AC784xx_Sent_Reg.h, [26](#)
- Sent_Reg_EnableFastMsgCRC

AC784xx_Sent_Reg.h, [26](#)
Sent_Reg_EnableFlushFifo
AC784xx_Sent_Reg.h, [27](#)
Sent_Reg_EnableInputInvert
AC784xx_Sent_Reg.h, [27](#)
Sent_Reg_EnableIrq
AC784xx_Sent_Reg.h, [28](#)
Sent_Reg_EnableOutputInvert
AC784xx_Sent_Reg.h, [28](#)
Sent_Reg_EnablePausePulse
AC784xx_Sent_Reg.h, [28](#)
Sent_Reg_EnableSerialMsgCRC
AC784xx_Sent_Reg.h, [29](#)
Sent_Reg_EnableSerialMsgProcess
AC784xx_Sent_Reg.h, [29](#)
Sent_Reg_EnableStatusNibbleInCRC
AC784xx_Sent_Reg.h, [30](#)
Sent_Reg_GetBase
AC784xx_Sent_Reg.h, [30](#)
Sent_Reg_GetChannelStatus
AC784xx_Sent_Reg.h, [30](#)
Sent_Reg_GetCurrentTimeStamp
AC784xx_Sent_Reg.h, [31](#)
Sent_Reg_GetEnhancedConfigType
AC784xx_Sent_Reg.h, [31](#)
Sent_Reg_GetFifoAvailable
AC784xx_Sent_Reg.h, [31](#)
Sent_Reg_GetFifoEmptyStatus
AC784xx_Sent_Reg.h, [32](#)
Sent_Reg_GetFifoFullStatus
AC784xx_Sent_Reg.h, [32](#)
Sent_Reg_GetFifoHalfFullStatus
AC784xx_Sent_Reg.h, [33](#)
Sent_Reg_GetFrameCRCNibble
AC784xx_Sent_Reg.h, [33](#)
Sent_Reg_GetFrameDataNibbles
AC784xx_Sent_Reg.h, [33](#)
Sent_Reg_GetFrameLengthCounter
AC784xx_Sent_Reg.h, [34](#)
Sent_Reg_GetFrameStatusNibble
AC784xx_Sent_Reg.h, [34](#)
Sent_Reg_GetIrqStatus
AC784xx_Sent_Reg.h, [34](#)
Sent_Reg_GetModuleClkDivider
AC784xx_Sent_Reg.h, [35](#)
Sent_Reg_GetSerialMsgCRC
AC784xx_Sent_Reg.h, [35](#)
Sent_Reg_GetSerialMsgData
AC784xx_Sent_Reg.h, [36](#)
Sent_Reg_GetSerialMsgID
AC784xx_Sent_Reg.h, [36](#)
Sent_Reg_GetSpcTransferState
AC784xx_Sent_Reg.h, [36](#)
Sent_Reg_SelectExternalTriggerSource
AC784xx_Sent_Reg.h, [37](#)
Sent_Reg_SelectInputDataSource
AC784xx_Sent_Reg.h, [37](#)
Sent_Reg_SetChannelFractionalDivider
AC784xx_Sent_Reg.h, [38](#)
Sent_Reg_SetChannelPreDivider
AC784xx_Sent_Reg.h, [38](#)
Sent_Reg_SetDataNibbleView
AC784xx_Sent_Reg.h, [38](#)
Sent_Reg_SetFrameCheckMode
AC784xx_Sent_Reg.h, [39](#)
Sent_Reg_SetFrameLength
AC784xx_Sent_Reg.h, [39](#)
Sent_Reg_SetGlitchFilterDepth
AC784xx_Sent_Reg.h, [40](#)
Sent_Reg_SetIrqStatus
AC784xx_Sent_Reg.h, [40](#)
Sent_Reg_SetModuleClkDivider
AC784xx_Sent_Reg.h, [40](#)
Sent_Reg_SetModuleTimeStampDivider
AC784xx_Sent_Reg.h, [41](#)
Sent_Reg_SetSPCParams
AC784xx_Sent_Reg.h, [41](#)
Sent_Reg_SetSPCPulseDelay
AC784xx_Sent_Reg.h, [42](#)
Sent_Reg_SetSPCPulseLength
AC784xx_Sent_Reg.h, [42](#)
Sent_Reg_SetSPCTimebase
AC784xx_Sent_Reg.h, [43](#)
Sent_Reg_SetSPCTriggerMode
AC784xx_Sent_Reg.h, [43](#)
Sent_Reg_SetSerialMsgType
AC784xx_Sent_Reg.h, [41](#)
Sent_Reg_SetWatchdogLimit
AC784xx_Sent_Reg.h, [44](#)
Sent_RxCallbackType
Sent_Hal_Types.h, [58](#)
Sent_SerialFrameType, [17](#)
ConfigBit, [17](#)
SerialMsgCRC, [17](#)
SerialMsgData, [18](#)
SerialMsgID, [18](#)
Sent_SerialMsgType
Sent_Hal_Types.h, [61](#)
Sent_SpcTimeBaseType
Sent_Hal_Types.h, [62](#)
Sent_SpcTriggerModeType
Sent_Hal_Types.h, [62](#)
Sent_StateType, [18](#)
EnabledInterrupts, [19](#)
ErrCallback, [19](#)
RxCallback, [19](#)
SentMsg, [19](#)
TxCallback, [19](#)
Sent_TransmitCtrlType, [20](#)
Mode, [20](#)
PulseDelayLength, [20](#)
PulseLength, [20](#)
TimeBase, [20](#)
SentMsg
Sent_StateType, [19](#)
SerialCRCEnable
Sent_ReceiveCtrlType, [16](#)
SerialDataEnable
Sent_ReceiveCtrlType, [16](#)
SerialFrame
Sent_MsgType, [11](#)
SerialMsgCRC

- Sent_SerialFrameType, [17](#)
- SerialMsgData
 - Sent_SerialFrameType, [18](#)
- SerialMsgID
 - Sent_SerialFrameType, [18](#)
- SerialMsgType
 - Sent_ReceiveCtrlType, [16](#)
- StatusIncludedInCRC
 - Sent_ReceiveCtrlType, [16](#)
- StatusNibble
 - Sent_FrameDataType, [7](#)
- TickUnit
 - Sent_ChannelConfigType, [4](#)
- TimeBase
 - Sent_TransmitCtrlType, [20](#)
- TimeStamp
 - Sent_FrameDataType, [7](#)
- TimeStampPreDivider
 - Sent_ModuleConfigType, [10](#)
- TotalFrameLength
 - Sent_FrameDataType, [8](#)
- TriggerSource
 - Sent_IoCtrlType, [9](#)
- TxCallback
 - Sent_ChannelConfigType, [5](#)
 - Sent_StateType, [19](#)
- WatchdogTimerLimit
 - Sent_ChannelConfigType, [5](#)
- ZeroAugumentInCRCDisable
 - Sent_ReceiveCtrlType, [16](#)