Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_COMPARE			Compare event on CC[n] match
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.18.5.14 PUBLISH_COMPARE[n] (n=0..5)

Address offset: 0x1C0 + (n × 0x4)

Publish configuration for event COMPARE[n]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event COMPARE[n] will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.18.5.15 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A-F	RW	COMPARE[i]_CLEAR (i=0..5)			Shortcut between event COMPARE[i] and task CLEAR
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
G-L	RW	COMPARE[i]_STOP (i=0..5)			Shortcut between event COMPARE[i] and task STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

6.18.5.16 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A-F	RW	COMPARE[i] (i=0..5)			Write '1' to enable interrupt for event COMPARE[i]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.18.5.17 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A-F	RW	COMPARE[i] (i=0..5)			Write '1' to disable interrupt for event COMPARE[i]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.18.5.18 MODE

Address offset: 0x504

Timer mode selection

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	MODE			Timer mode
			Timer	0	Select Timer mode
			Counter	1	Select Counter mode
			LowPowerCounter	2	Select Low Power Counter mode

6.18.5.19 BITMODE

Address offset: 0x508

Configure the number of bits used by the TIMER

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	BITMODE			Timer bit width
			16Bit	0	16 bit timer bit width
			08Bit	1	8 bit timer bit width
			24Bit	2	24 bit timer bit width
			32Bit	3	32 bit timer bit width

6.18.5.20 PRESCALER

Address offset: 0x510

Timer prescaler register

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A
Reset 0x00000004	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	PRESCALER		[0..9]	Prescaler value

6.18.5.21 CC[n] (n=0..5)

Address offset: 0x540 + (n × 0x4)

Capture/Compare register n

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	CC			Capture/Compare value Only the number of bits indicated by BITMODE will be used by the TIMER.

6.18.6 Electrical specification

6.19 TWIM — I²C compatible two-wire interface master with EasyDMA

TWI master with EasyDMA (TWIM) is a two-wire half-duplex master which can communicate with multiple slave devices connected to the same bus

Listed here are the main features for TWIM:

- I²C compatible
- 100 kbps, 250 kbps, or 400 kbps
- Support for clock stretching (non I²C compliant)
- EasyDMA

The two-wire interface can communicate with a bi-directional wired-AND bus with two lines (SCL, SDA). The protocol makes it possible to interconnect up to 127 individually addressable devices. TWIM is not compatible with CBUS.

The GPIOs used for each two-wire interface line can be chosen from any GPIO on the device and are independently configurable. This enables great flexibility in device pinout and efficient use of board space and signal routing.

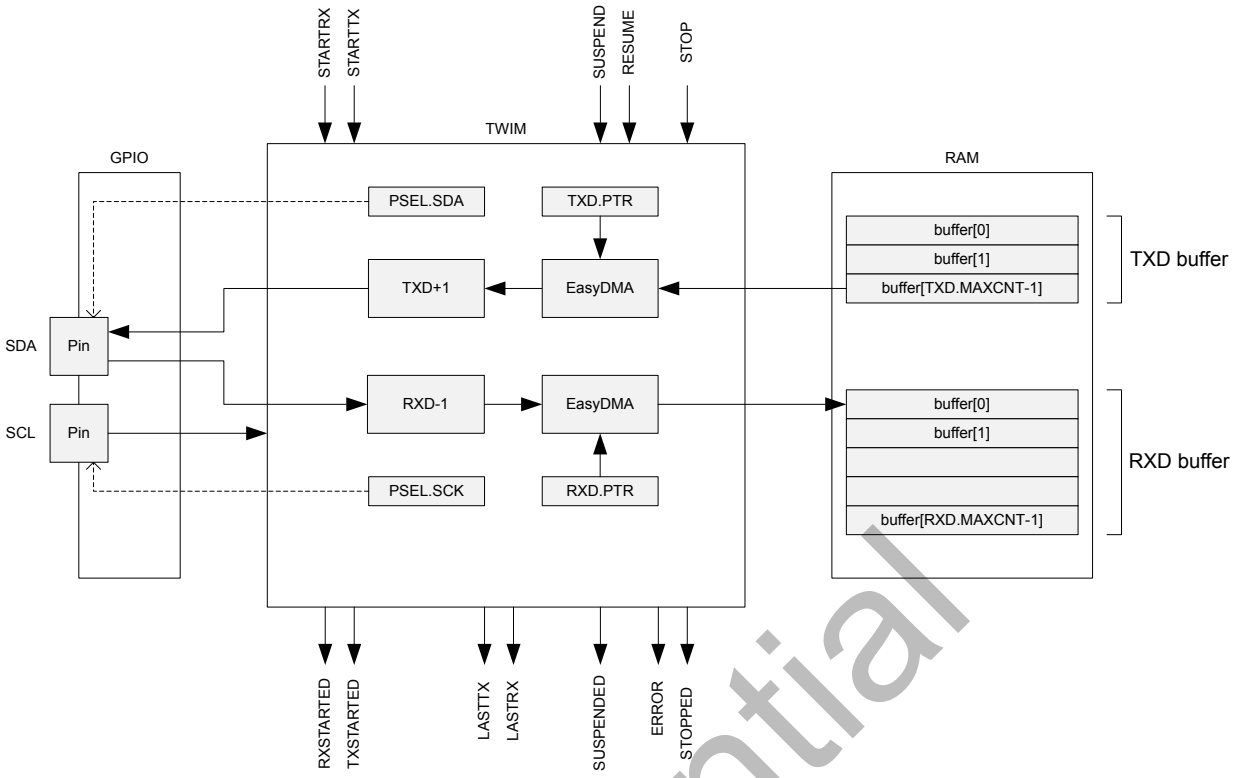

Figure 87: TWI master with EasyDMA

A typical TWI setup consists of one master and one or more slaves. For an example, see A typical TWI setup comprising one master and three slaves on page 292. This TWIM is only able to operate as a single master on the TWI bus. Multi-master bus configuration is not supported.

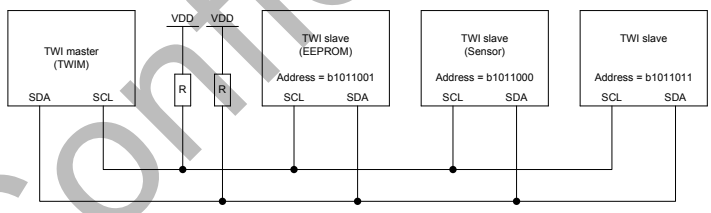

Figure 88: A typical TWI setup comprising one master and three slaves

This TWI master supports clock stretching performed by the slaves. Note that the SCK pulse following a stretched clock cycle may be shorter than specified by the I2C specification.

The TWI master is started by triggering the STARTTX or STARTRX tasks, and stopped by triggering the STOP task. The TWI master will generate a STOPPED event when it has stopped following a STOP task. The TWI master cannot get stopped while it is suspended, so the STOP task has to be issued after the TWI master has been resumed.

After the TWI master is started, the STARTTX task or the STARTRX task should not be triggered again before the TWI master has stopped, i.e. following a LASTRX, LASTTX or STOPPED event.

If a NACK is clocked in from the slave, the TWI master will generate an ERROR event.

6.19.1 EasyDMA

The TWI master implements EasyDMA for reading and writing to and from the RAM.

If the TXD.PTR and the RXD.PTR are not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 21 for more information about the different memory regions.

The .PTR and .MAXCNT registers are double-buffered. They can be updated and prepared for the next RX/TX transmission immediately after having received the RXSTARTED/TXSTARTED event.

The STOPPED event indicates that EasyDMA has finished accessing the buffer in RAM.

6.19.2 Master write sequence

A TWI master write sequence is started by triggering the STARTTX task. After the STARTTX task has been triggered, the TWI master will generate a start condition on the TWI bus, followed by clocking out the address and the READ/WRITE bit set to 0 (WRITE=0, READ=1).

The address must match the address of the slave device that the master wants to write to. The READ/WRITE bit is followed by an ACK/NACK bit (ACK=0 or NACK=1) generated by the slave.

After receiving the ACK bit, the TWI master will clock out the data bytes found in the transmit buffer located in RAM at the address specified in the TXD.PTR register. Each byte clocked out from the master will be followed by an ACK/NACK bit clocked in from the slave.

A typical TWI master write sequence is illustrated in TWI master writing data to a slave on page 293. Occurrence 2 in the figure illustrates clock stretching performed by the TWI master following a SUSPEND task.

A SUSPENDED event indicates that the SUSPEND task has taken effect; this event can be used to synchronize the software.

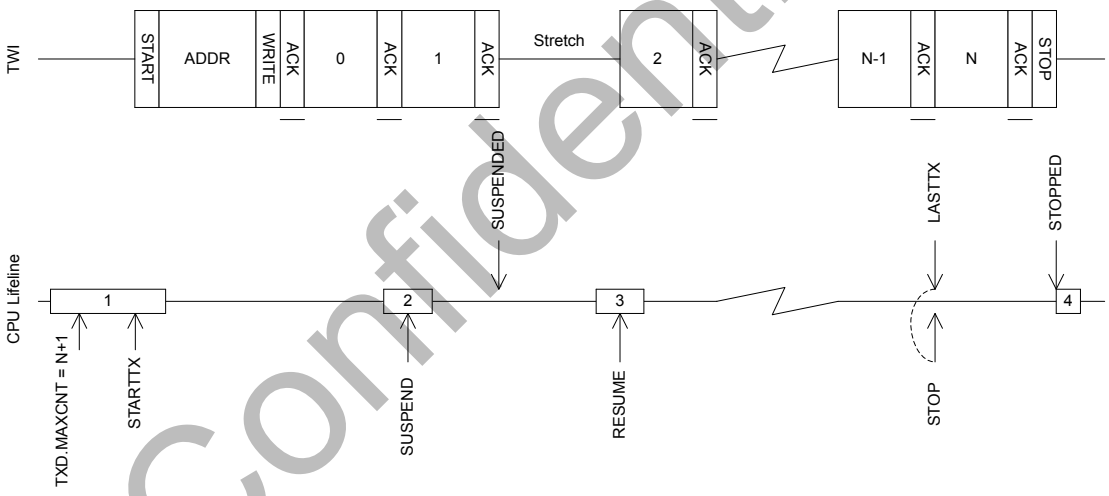

Figure 89: TWI master writing data to a slave

The TWI master will generate a LASTTX event when it starts to transmit the last byte, this is illustrated in TWI master writing data to a slave on page 293

The TWI master is stopped by triggering the STOP task, this task should be triggered during the transmission of the last byte to secure that the TWI will stop as fast as possible after sending the last byte. It is safe to use the shortcut between LASTTX and STOP to accomplish this.

Note that the TWI master does not stop by itself when the whole RAM buffer has been sent, or when an error occurs. The STOP task must be issued, through the use of a local or PPI shortcut, or in software as part of the error handler.

The TWI master cannot get stopped while it is suspended, so the STOP task has to be issued after the TWI master has been resumed.

6.19.3 Master read sequence

A TWI master read sequence is started by triggering the STARTRX task. After the STARTRX task has been triggered the TWI master will generate a start condition on the TWI bus, followed by clocking out the

address and the READ/WRITE bit set to 1 (WRITE = 0, READ = 1). The address must match the address of the slave device that the master wants to read from. The READ/WRITE bit is followed by an ACK/NACK bit (ACK=0 or NACK = 1) generated by the slave.

After having sent the ACK bit the TWI slave will send data to the master using the clock generated by the master.

Data received will be stored in RAM at the address specified in the RXD.PTR register. The TWI master will generate an ACK after all but the last byte received from the slave. The TWI master will generate a NACK after the last byte received to indicate that the read sequence shall stop.

A typical TWI master read sequence is illustrated in The TWI master reading data from a slave on page 294. Occurrence 2 in the figure illustrates clock stretching performed by the TWI master following a SUSPEND task.

A SUSPENDED event indicates that the SUSPEND task has taken effect; this event can be used to synchronize the software.

The TWI master will generate a LASTRX event when it is ready to receive the last byte, this is illustrated in The TWI master reading data from a slave on page 294. If RXD.MAXCNT > 1 the LASTRX event is generated after sending the ACK of the previously received byte. If RXD.MAXCNT = 1 the LASTRX event is generated after receiving the ACK following the address and READ bit.

The TWI master is stopped by triggering the STOP task, this task must be triggered before the NACK bit is supposed to be transmitted. The STOP task can be triggered at any time during the reception of the last byte. It is safe to use the shortcut between LASTRX and STOP to accomplish this.

Note that the TWI master does not stop by itself when the RAM buffer is full, or when an error occurs. The STOP task must be issued, through the use of a local or PPI shortcut, or in software as part of the error handler.

The TWI master cannot get stopped while it is suspended, so the STOP task has to be issued after the TWI master has been resumed.

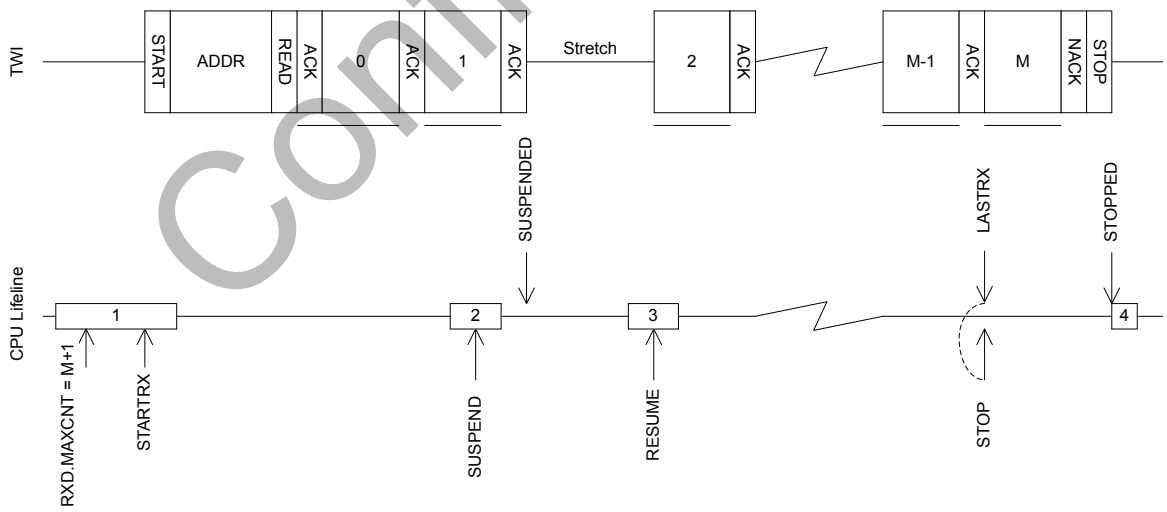

Figure 90: The TWI master reading data from a slave

6.19.4 Master repeated start sequence

A typical repeated start sequence is one in which the TWI master writes two bytes to the slave followed by reading four bytes from the slave. This example uses shortcuts to perform the simplest type of repeated start sequence, i.e. one write followed by one read. The same approach can be used to perform a repeated start sequence where the sequence is read followed by write.

The figure A repeated start sequence, where the TWI master writes two bytes followed by reading 4 bytes from the slave on page 295 illustrates this:

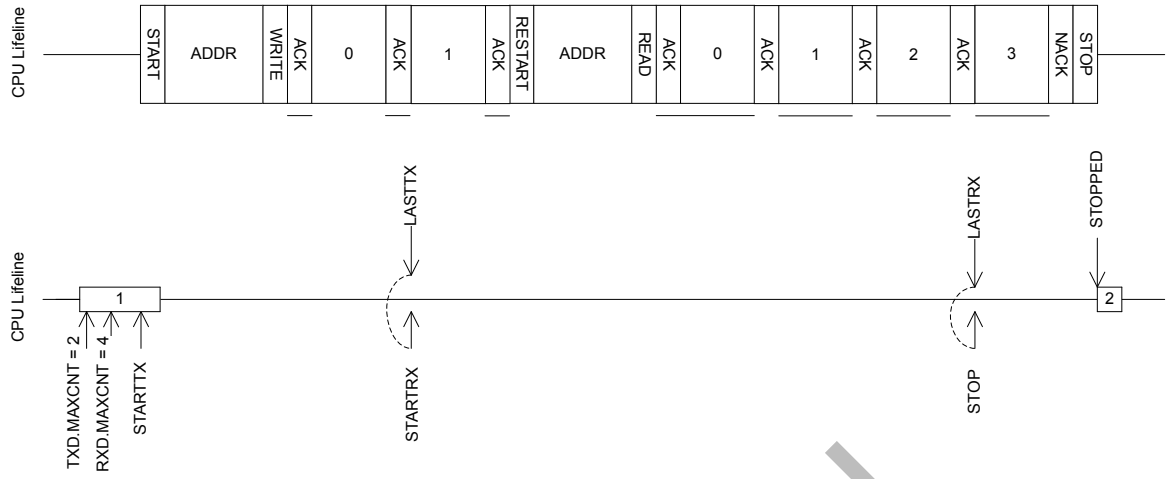

Figure 91: A repeated start sequence, where the TWI master writes two bytes followed by reading 4 bytes from the slave

If a more complex repeated start sequence is needed and the TWI firmware drive is serviced in a low priority interrupt it may be necessary to use the SUSPEND task and SUSPENDED event to guarantee that the correct tasks are generated at the correct time. This is illustrated in A double repeated start sequence using the SUSPEND task to secure safe operation in low priority interrupts on page 295.

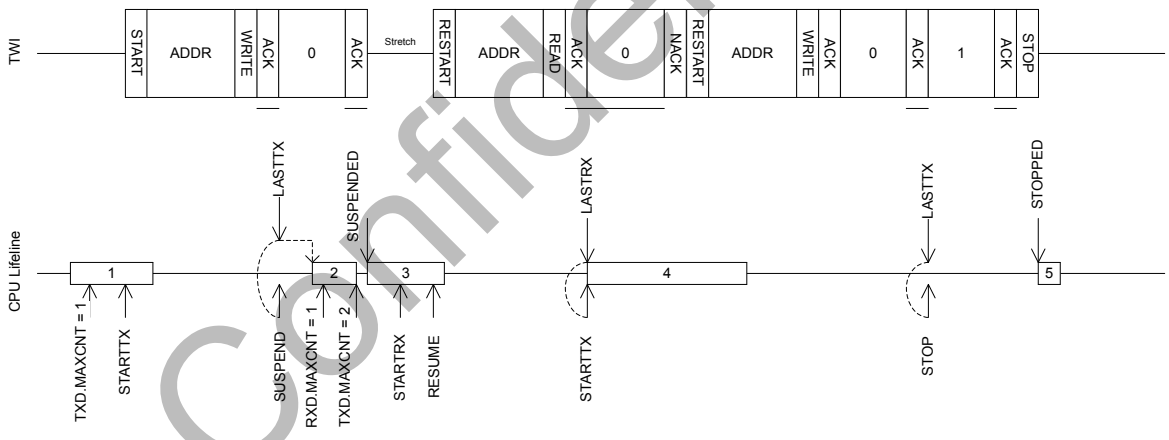

Figure 92: A double repeated start sequence using the SUSPEND task to secure safe operation in low priority interrupts

6.19.5 Low power

When putting the system in low power and the peripheral is not needed, lowest possible power consumption is achieved by stopping, and then disabling the peripheral.

The STOP task may not be always needed (the peripheral might already be stopped), but if it is sent, software shall wait until the STOPPED event was received as a response before disabling the peripheral through the ENABLE register.

6.19.6 Master mode pin configuration

The SCL and SDA signals associated with the TWI master are mapped to physical pins according to the configuration specified in the PSEL.SCL and PSEL.SDA registers respectively.

The PSEL.SCL and PSEL.SDA registers and their configurations are only used as long as the TWI master is enabled, and retained only as long as the device is in ON mode. When the peripheral is disabled, the pins

will behave as regular GPIOs, and use the configuration in their respective OUT bit field and PIN_CNF[n] register. PSEL.SCL, PSEL.SDA must only be configured when the TWI master is disabled.

To secure correct signal levels on the pins used by the TWI master when the system is in OFF mode, and when the TWI master is disabled, these pins must be configured in the GPIO peripheral as described in GPIO configuration before enabling peripheral on page 296.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

TWI master signal	TWI master pin	Direction	Output value	Drive strength
SCL	As specified in PSEL.SCL	Input	Not applicable	S0D1
SDA	As specified in PSEL.SDA	Input	Not applicable	S0D1

Table 83: GPIO configuration before enabling peripheral

6.19.7 Registers

Base address	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50008000 0x40008000	TWIM	TWIM0 : S TWIM0 : NS	US	SA	Two-wire interface master 0	
0x50009000 0x40009000	TWIM	TWIM1 : S TWIM1 : NS	US	SA	Two-wire interface master 1	
0x5000A000 0x4000A000	TWIM	TWIM2 : S TWIM2 : NS	US	SA	Two-wire interface master 2	
0x5000B000 0x4000B000	TWIM	TWIM3 : S TWIM3 : NS	US	SA	Two-wire interface master 3	

Table 84: Instances

Register	Offset	Security	Description
TASKS_STARTRX	0x000		Start TWI receive sequence
TASKS_STARTTX	0x008		Start TWI transmit sequence
TASKS_STOP	0x014		Stop TWI transaction. Must be issued while the TWI master is not suspended.
TASKS_SUSPEND	0x01C		Suspend TWI transaction
TASKS_RESUME	0x020		Resume TWI transaction
SUBSCRIBE_STARTRX	0x080		Subscribe configuration for task STARTRX
SUBSCRIBE_STARTTX	0x088		Subscribe configuration for task STARTTX
SUBSCRIBE_STOP	0x094		Subscribe configuration for task STOP
SUBSCRIBE_SUSPEND	0x09C		Subscribe configuration for task SUSPEND
SUBSCRIBE_RESUME	0x0A0		Subscribe configuration for task RESUME
EVENTS_STOPPED	0x104		TWI stopped
EVENTS_ERROR	0x124		TWI error
EVENTS_SUSPENDED	0x148		Last byte has been sent out after the SUSPEND task has been issued, TWI traffic is now suspended.
EVENTS_RXSTARTED	0x14C		Receive sequence started
EVENTS_TXSTARTED	0x150		Transmit sequence started
EVENTS_LASTRX	0x15C		Byte boundary, starting to receive the last byte
EVENTS_LASTTX	0x160		Byte boundary, starting to transmit the last byte
PUBLISH_STOPPED	0x184		Publish configuration for event STOPPED
PUBLISH_ERROR	0x1A4		Publish configuration for event ERROR
PUBLISH_SUSPENDED	0x1C8		Publish configuration for event SUSPENDED
PUBLISH_RXSTARTED	0x1CC		Publish configuration for event RXSTARTED
PUBLISH_TXSTARTED	0x1D0		Publish configuration for event TXSTARTED

Register	Offset	Security	Description
PUBLISH_LASTRX	0x1DC		Publish configuration for event LASTRX
PUBLISH_LASTTX	0x1E0		Publish configuration for event LASTTX
SHORTS	0x200		Shortcuts between local events and tasks
INTEN	0x300		Enable or disable interrupt
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
ERRORSRC	0x4C4		Error source
ENABLE	0x500		Enable TWIM
PSEL.SCL	0x508		Pin select for SCL signal
PSEL.SDA	0x50C		Pin select for SDA signal
FREQUENCY	0x524		TWI frequency. Accuracy depends on the HFCLK source selected.
RXD.PTR	0x534		Data pointer
RXD.MAXCNT	0x538		Maximum number of bytes in receive buffer
RXD.AMOUNT	0x53C		Number of bytes transferred in the last transaction
RXD.LIST	0x540		EasyDMA list type
TXD.PTR	0x544		Data pointer
TXD.MAXCNT	0x548		Maximum number of bytes in transmit buffer
TXD.AMOUNT	0x54C		Number of bytes transferred in the last transaction
TXD.LIST	0x550		EasyDMA list type
ADDRESS	0x588		Address used in the TWI transfer

Table 85: Register overview

6.19.7.1 TASKS_STARTRX

Address offset: 0x000

Start TWI receive sequence

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																																A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_STARTRX			Start TWI receive sequence
			Trigger	1	Trigger task

6.19.7.2 TASKS_STARTTX

Address offset: 0x008

Start TWI transmit sequence

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																																A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_STARTTX			Start TWI transmit sequence
			Trigger	1	Trigger task

6.19.7.3 TASKS_STOP

Address offset: 0x014

Stop TWI transaction. Must be issued while the TWI master is not suspended.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_STOP			Stop TWI transaction. Must be issued while the TWI master is not suspended.
			Trigger	1	Trigger task

6.19.7.4 TASKS_SUSPEND

Address offset: 0x01C

Suspend TWI transaction

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_SUSPEND			Suspend TWI transaction
			Trigger	1	Trigger task

6.19.7.5 TASKS_RESUME

Address offset: 0x020

Resume TWI transaction

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_RESUME			Resume TWI transaction
			Trigger	1	Trigger task

6.19.7.6 SUBSCRIBE_STARTRX

Address offset: 0x080

Subscribe configuration for task STARTRX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task STARTRX will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.19.7.7 SUBSCRIBE_STARTTX

Address offset: 0x088

Subscribe configuration for task STARTTX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task STARTTX will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.19.7.8 SUBSCRIBE_STOP

Address offset: 0x094

Subscribe configuration for task STOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.19.7.9 SUBSCRIBE_SUSPEND

Address offset: 0x09C

Subscribe configuration for task SUSPEND

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task SUSPEND will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.19.7.10 SUBSCRIBE_RESUME

Address offset: 0x0A0

Subscribe configuration for task RESUME

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task RESUME will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.19.7.11 EVENTS_STOPPED

Address offset: 0x104

TWI stopped

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_STOPPED			TWI stopped
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.19.7.12 EVENTS_ERROR

Address offset: 0x124

TWI error

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_ERROR			TWI error
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.19.7.13 EVENTS_SUSPENDED

Address offset: 0x148

Last byte has been sent out after the SUSPEND task has been issued, TWI traffic is now suspended.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_SUSPENDED			Last byte has been sent out after the SUSPEND task has been issued, TWI traffic is now suspended.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.19.7.14 EVENTS_RXSTARTED

Address offset: 0x14C

Receive sequence started

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_RXSTARTED			Receive sequence started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.19.7.15 EVENTS_TXSTARTED

Address offset: 0x150

Transmit sequence started

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_TXSTARTED			Transmit sequence started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.19.7.16 EVENTS_LASTRX

Address offset: 0x15C

Byte boundary, starting to receive the last byte

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_LASTRX			Byte boundary, starting to receive the last byte
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.19.7.17 EVENTS_LASTTX

Address offset: 0x160

Byte boundary, starting to transmit the last byte

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_LASTTX			Byte boundary, starting to transmit the last byte
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.19.7.18 PUBLISH_STOPPED

Address offset: 0x184

Publish configuration for event STOPPED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event STOPPED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.19.7.19 PUBLISH_ERROR

Address offset: 0x1A4

Publish configuration for event ERROR

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event ERROR will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.19.7.20 PUBLISH_SUSPENDED

Address offset: 0x1C8

Publish configuration for event SUSPENDED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event SUSPENDED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.19.7.21 PUBLISH_RXSTARTED

Address offset: 0x1CC

Publish configuration for event RXSTARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event RXSTARTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.19.7.22 PUBLISH_TXSTARTED

Address offset: 0x1D0

Publish configuration for event TXSTARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event TXSTARTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.19.7.23 PUBLISH_LASTRX

Address offset: 0x1DC

Publish configuration for event LASTRX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event LASTRX will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.19.7.24 PUBLISH_LASTTX

Address offset: 0x1E0

Publish configuration for event LASTTX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event LASTTX will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.19.7.25 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	LASTTX_STARTRX			Shortcut between event LASTTX and task STARTRX
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
B	RW	LASTTX_SUSPEND			Shortcut between event LASTTX and task SUSPEND
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
C	RW	LASTTX_STOP			Shortcut between event LASTTX and task STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
D	RW	LASTRX_STARTTX			Shortcut between event LASTRX and task STARTTX
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
E	RW	LASTRX_SUSPEND			Shortcut between event LASTRX and task SUSPEND
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
F	RW	LASTRX_STOP			Shortcut between event LASTRX and task STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

6.19.7.26 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				J I H G F D A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	STOPPED			Enable or disable interrupt for event STOPPED
			Disabled	0	Disable
			Enabled	1	Enable
D	RW	ERROR			Enable or disable interrupt for event ERROR
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	SUSPENDED			Enable or disable interrupt for event SUSPENDED
			Disabled	0	Disable
			Enabled	1	Enable
G	RW	RXSTARTED			Enable or disable interrupt for event RXSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
H	RW	TXSTARTED			Enable or disable interrupt for event TXSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
I	RW	LASTRX			Enable or disable interrupt for event LASTRX
			Disabled	0	Disable
			Enabled	1	Enable
J	RW	LASTTX			Enable or disable interrupt for event LASTTX
			Disabled	0	Disable
			Enabled	1	Enable

6.19.7.27 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				J I H G F D A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	STOPPED			Write '1' to enable interrupt for event STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	ERROR			Write '1' to enable interrupt for event ERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	SUSPENDED			Write '1' to enable interrupt for event SUSPENDED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	RXSTARTED			Write '1' to enable interrupt for event RXSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	TXSTARTED			Write '1' to enable interrupt for event TXSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	LASTRX			Write '1' to enable interrupt for event LASTRX
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	LASTTX			Write '1' to enable interrupt for event LASTTX
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.19.7.28 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				J I H G F D A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	STOPPED			Write '1' to disable interrupt for event STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	ERROR			Write '1' to disable interrupt for event ERROR
			Clear	1	Disable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				J I H G F D A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	SUSPENDED			Write '1' to disable interrupt for event SUSPENDED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	RXSTARTED			Write '1' to disable interrupt for event RXSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	TXSTARTED			Write '1' to disable interrupt for event TXSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	LASTRX			Write '1' to disable interrupt for event LASTRX
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	LASTTX			Write '1' to disable interrupt for event LASTTX
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.19.7.29 ERRORSRC

Address offset: 0x4C4

Error source

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	OVERRUN			Overrun error A new byte was received before previous byte got transferred into RXD buffer. (Previous data is lost)
			NotReceived	0	Error did not occur
			Received	1	Error occurred
B	RW	ANACK			NACK received after sending the address (write '1' to clear)
			NotReceived	0	Error did not occur
			Received	1	Error occurred
C	RW	DNACK			NACK received after sending a data byte (write '1' to clear)
			NotReceived	0	Error did not occur
			Received	1	Error occurred

6.19.7.30 ENABLE

Address offset: 0x500

Enable TWIM

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable TWIM
			Disabled	0	Disable TWIM
			Enabled	6	Enable TWIM

6.19.7.31 PSEL.SCL

Address offset: 0x508

Pin select for SCL signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.19.7.32 PSEL.SDA

Address offset: 0x50C

Pin select for SDA signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.19.7.33 FREQUENCY

Address offset: 0x524

TWI frequency. Accuracy depends on the HFCLK source selected.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x04000000				**0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	FREQUENCY			TWI master clock frequency
			K100	0x01980000	100 kbps
			K250	0x04000000	250 kbps
			K400	0x06400000	400 kbps

6.19.7.34 RXD.PTR

Address offset: 0x534

Data pointer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	PTR			Data pointer

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.19.7.35 RXD.MAXCNT

Address offset: 0x538

Maximum number of bytes in receive buffer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0x1FFF]	Maximum number of bytes in receive buffer

6.19.7.36 RXD.AMOUNT

Address offset: 0x53C

Number of bytes transferred in the last transaction

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0x1FFF]	Number of bytes transferred in the last transaction. In case of NACK error, includes the NACK'ed byte.

6.19.7.37 RXD.LIST

Address offset: 0x540

EasyDMA list type

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	LIST			List type
			Disabled	0	Disable EasyDMA list
			ArrayList	1	Use array list

6.19.7.38 TXD.PTR

Address offset: 0x544

Data pointer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	PTR			Data pointer

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.19.7.39 TXD.MAXCNT

Address offset: 0x548

Maximum number of bytes in transmit buffer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0x1FFF]	Maximum number of bytes in transmit buffer

6.19.7.40 TXD.AMOUNT

Address offset: 0x54C

Number of bytes transferred in the last transaction

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0x1FFF]	Number of bytes transferred in the last transaction. In case of NACK error, includes the NACK'ed byte.

6.19.7.41 TXD.LIST

Address offset: 0x550

EasyDMA list type

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	LIST			List type
			Disabled	0	Disable EasyDMA list
			ArrayList	1	Use array list

6.19.7.42 ADDRESS

Address offset: 0x588

Address used in the TWI transfer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	ADDRESS			Address used in the TWI transfer

6.19.8 Electrical specification

6.19.8.1 TWIM interface electrical specifications

Symbol	Description	Min.	Typ.	Max.	Units
$f_{TWIM,SCL}$	Bit rates for TWIM[15]	100		400	kbps
$t_{TWIM,START}$	Time from STARTRX/STARTTX task to transmission started	..	..	..	µs

6.19.8.2 Two Wire Interface Master (TWIM) timing specifications

Symbol	Description	Min.	Typ.	Max.	Units
t_{TWIM,SU_DAT}	Data setup time before positive edge on SCL – all modes	300			ns
t_{TWIM,HD_DAT}	Data hold time after negative edge on SCL – all modes	500			ns
$t_{TWIM,HD\_STA,100kbps}$	TWIM master hold time for START and repeated START condition, 100 kbps	10000			ns
$t_{TWIM,HD\_STA,250kbps}$	TWIM master hold time for START and repeated START condition, 250kbps	4000			ns
$t_{TWIM,HD\_STA,400kbps}$	TWIM master hold time for START and repeated START condition, 400 kbps	2500			ns
$t_{TWIM,SU\_STO,100kbps}$	TWIM master setup time from SCL high to STOP condition, 100 kbps	5000			ns
$t_{TWIM,SU\_STO,250kbps}$	TWIM master setup time from SCL high to STOP condition, 250 kbps	2000			ns
$t_{TWIM,SU\_STO,400kbps}$	TWIM master setup time from SCL high to STOP condition, 400 kbps	1250			ns
$t_{TWIM,BUF,100kbps}$	TWIM master bus free time between STOP and START conditions, 100 kbps	5800			ns
$t_{TWIM,BUF,250kbps}$	TWIM master bus free time between STOP and START conditions, 250 kbps	2700			ns
$t_{TWIM,BUF,400kbps}$	TWIM master bus free time between STOP and START conditions, 400 kbps	2100			ns

[15] High bit rates or stronger pull-ups may require GPIOs to be set as High Drive, see GPIO chapter for more details.

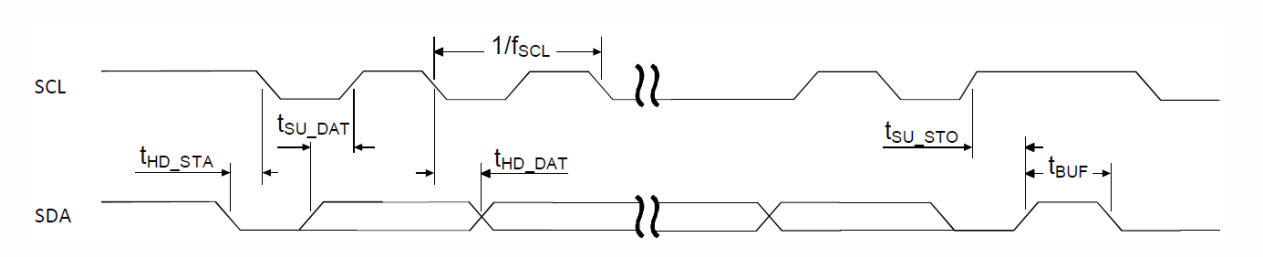

Figure 93: TWIM timing diagram, 1 byte transaction

6.19.9 Pullup resistor

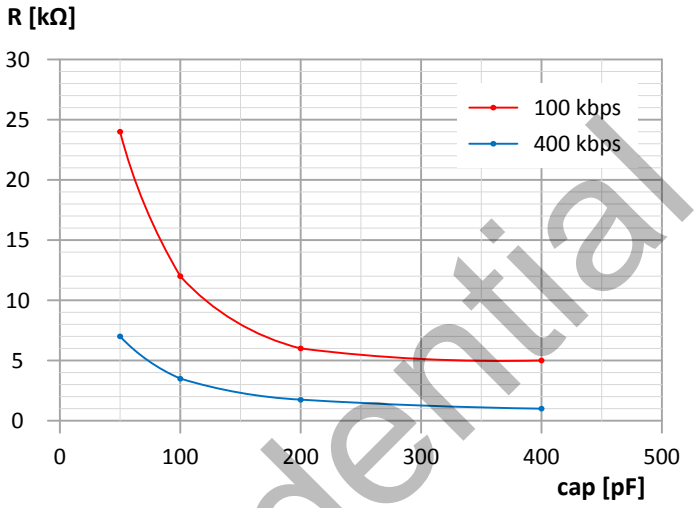

Figure 94: Recommended TWIM pullup value vs. line capacitance

- The I2C specification allows a line capacitance of 400 pF at most.
- The value of internal pullup resistor (R_{PU}) for nRF9160 can be found in GPIO — General purpose input/output on page 93.

6.20 TWIS — I²C compatible two-wire interface slave with EasyDMA

TWI slave with EasyDMA (TWIS) is compatible with I²C operating at 100 kHz and 400 kHz. The TWI transmitter and receiver implement EasyDMA.

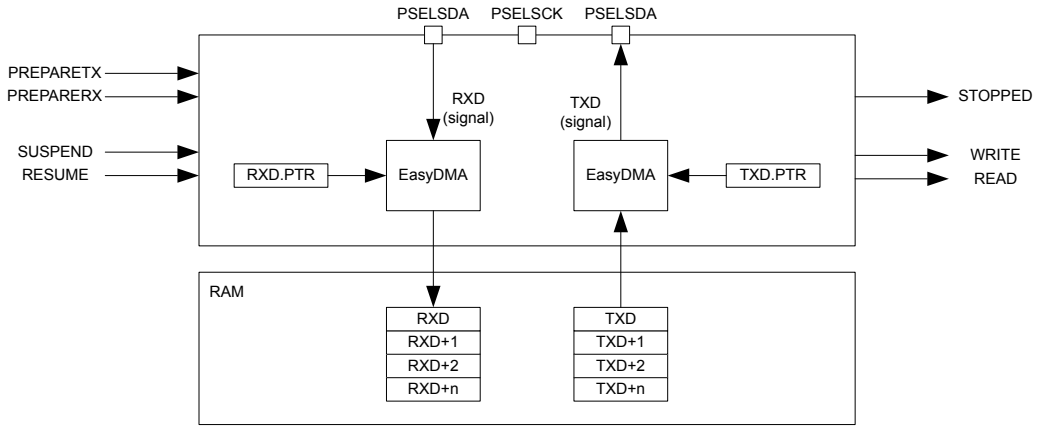

Figure 95: TWI slave with EasyDMA

A typical TWI setup consists of one master and one or more slaves. For an example, see A typical TWI setup comprising one master and three slaves on page 312. TWIS is only able to operate with a single master on the TWI bus.

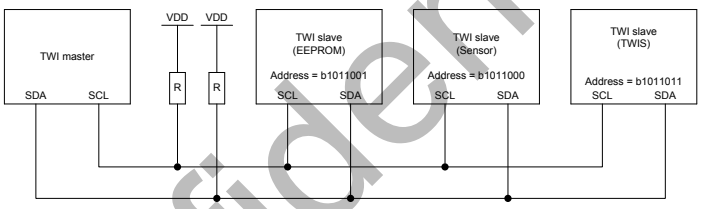

Figure 96: A typical TWI setup comprising one master and three slaves

The TWI slave state machine is illustrated in TWI slave state machine on page 313 and TWI slave state machine symbols on page 313 is explaining the different symbols used in the state machine.

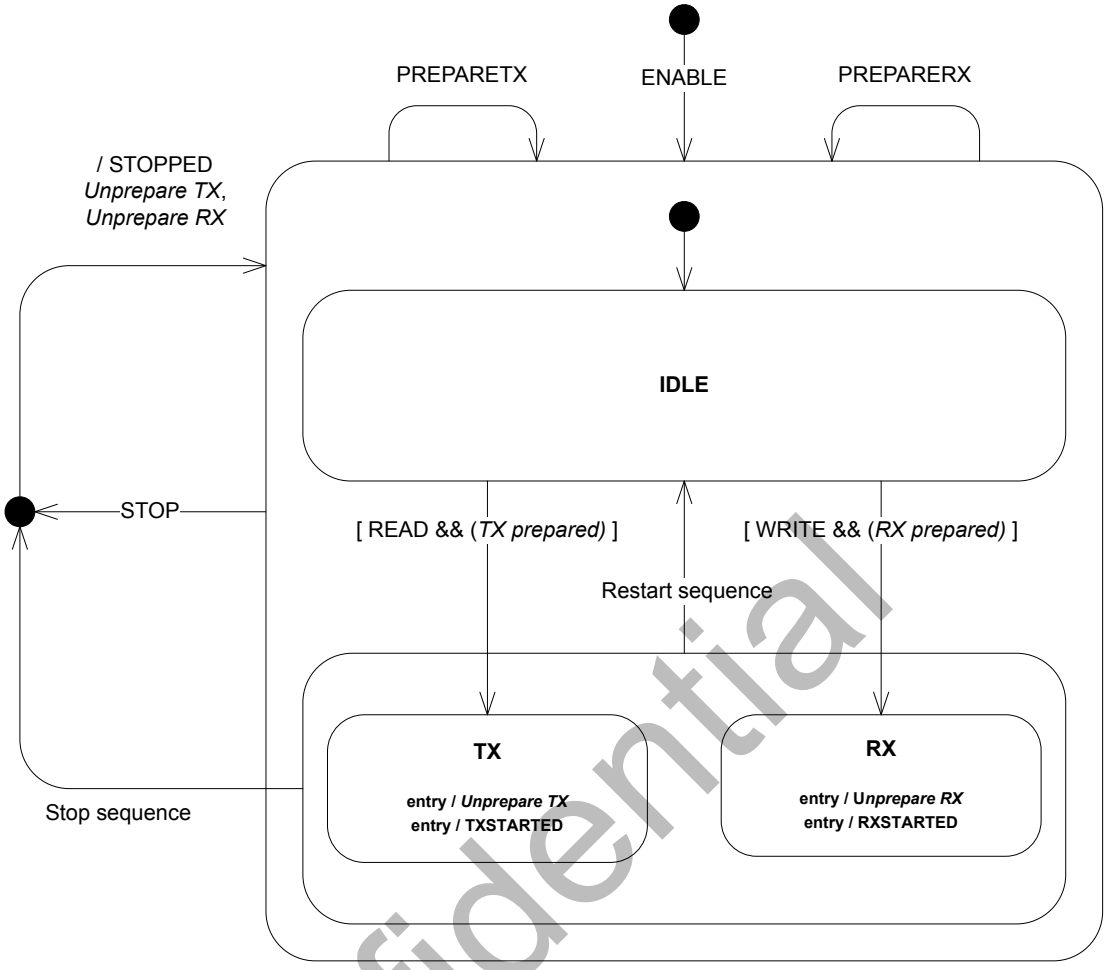

Figure 97: TWI slave state machine

Symbol	Type	Description
ENABLE	Register	The TWI slave has been enabled via the ENABLE register
PREPARETX	Task	The TASKS_PREPARETX task has been triggered
STOP	Task	The TASKS_STOP task has been triggered
PREPARERX	Task	The TASKS_PREPARERX task has been triggered
STOPPED	Event	The EVENTS_STOPPED event was generated
RXSTARTED	Event	The EVENTS_RXSTARTED event was generated
TXSTARTED	Event	The EVENTS_TXSTARTED event was generated
TX prepared	Internal	Internal flag indicating that a TASKS_PREPARETX task has been triggered. This flag is not visible to the user.
RX prepared	Internal	Internal flag indicating that a TASKS_PREPARERX task has been triggered. This flag is not visible to the user.
Unprepare TX	Internal	Clears the internal 'TX prepared' flag until next TASKS_PREPARETX task.
Unprepare RX	Internal	Clears the internal 'RX prepared' flag until next TASKS_PREPARERX task.
Stop sequence	TWI protocol	A TWI stop sequence was detected
Restart sequence	TWI protocol	A TWI restart sequence was detected

Table 86: TWI slave state machine symbols

The TWI slave supports clock stretching performed by the master.

The TWI slave operates in a low power mode while waiting for a TWI master to initiate a transfer. As long as the TWI slave is not addressed, it will remain in this low power mode.

To secure correct behaviour of the TWI slave, PSEL.SCL, PSEL.SDA, CONFIG and the ADDRESS[n] registers, must be configured prior to enabling the TWI slave through the ENABLE register. Similarly, changing these settings must be performed while the TWI slave is disabled. Failing to do so may result in unpredictable behaviour.

6.20.1 EasyDMA

The TWI slave implements EasyDMA for reading and writing to and from the RAM.

The STOPPED event indicates that EasyDMA has finished accessing the buffer in RAM.

If the TXD.PTR and the RXD.PTR are not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 21 for more information about the different memory regions.

6.20.2 TWI slave responding to a read command

Before the TWI slave can respond to a read command the TWI slave must be configured correctly and enabled via the ENABLE register. When enabled the TWI slave will be in its IDLE state where it will consume I_{IDLE} .

A read command is started when the TWI master generates a start condition on the TWI bus, followed by clocking out the address and the READ/WRITE bit set to 1 (WRITE=0, READ=1). The READ/WRITE bit is followed by an ACK/NACK bit (ACK=0 or NACK=1) response from the TWI slave.

The TWI slave is able to listen for up to two addresses at the same time. Which addresses to listen for is configured in the ADDRESS registers and the CONFIG register.

The TWI slave will only acknowledge (ACK) the read command if the address presented by the master matches one of the addresses the slave is configured to listen for. The TWI slave will generate a READ event when it acknowledges the read command.

The TWI slave is only able to detect a read command from the IDLE state.

The TWI slave will set an internal 'TX prepared' flag when the PREPARETX task is triggered.

When the read command is received the TWI slave will enter the TX state if the internal 'TX prepared' flag is set.

If the internal 'TX prepared' flag is not set when the read command is received, the TWI slave will stretch the master's clock until the PREPARETX task is triggered and the internal 'TX prepared' flag is set.

The TWI slave will generate the TXSTARTED event and clear the 'TX prepared' flag ('unprepare TX') when it enters the TX state. In this state the TWI slave will send the data bytes found in the transmit buffer to the master using the master's clock. The TWI slave will consume I_{TX} in this mode.

The TWI slave will go back to the IDLE state if the TWI slave receives a restart command when it is in the TX state.

The TWI slave is stopped when it receives the stop condition from the TWI master. A STOPPED event will be generated when the transaction has stopped. The TWI slave will clear the 'TX prepared' flag ('unprepare TX') and go back to the IDLE state when it has stopped.

The transmit buffer is located in RAM at the address specified in the TXD.PTR register. The TWI slave will only be able to send TXD.MAXCNT bytes from the transmit buffer for each transaction. If the TWI master forces the slave to send more than TXD.MAXCNT bytes, the slave will send the byte specified in the ORC register to the master instead. If this happens, an ERROR event will be generated.

The EasyDMA configuration registers, see TXD.PTR etc., are latched when the TXSTARTED event is generated.

The TWI slave can be forced to stop by triggering the STOP task. A STOPPED event will be generated when the TWI slave has stopped. The TWI slave will clear the 'TX prepared' flag and go back to the IDLE state when it has stopped, see also Terminating an ongoing TWI transaction on page 317.

Each byte sent from the slave will be followed by an ACK/NACK bit sent from the master. The TWI master will generate a NACK following the last byte that it wants to receive to tell the slave to release the bus so that the TWI master can generate the stop condition. The TXD.AMOUNT register can be queried after a transaction to see how many bytes were sent.

A typical TWI slave read command response is illustrated in The TWI slave responding to a read command on page 315. Occurrence 2 in the figure illustrates clock stretching performed by the TWI slave following a SUSPEND task.

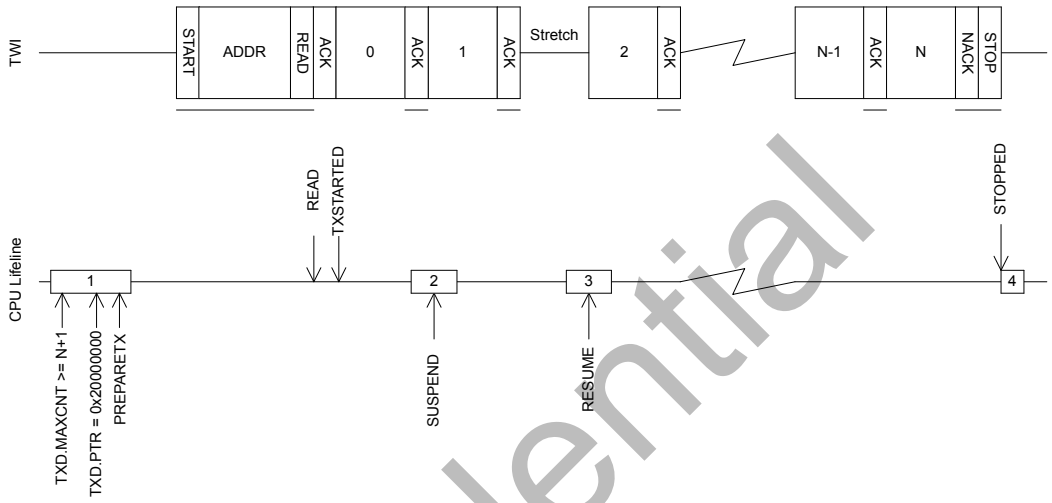

Figure 98: The TWI slave responding to a read command

6.20.3 TWI slave responding to a write command

Before the TWI slave can respond to a write command the TWI slave must be configured correctly and enabled via the ENABLE register. When enabled the TWI slave will be in its IDLE state where it will consume I_{IDLE}.

A write command is started when the TWI master generates a start condition on the TWI bus, followed by clocking out the address and the READ/WRITE bit set to 0 (WRITE=0, READ=1). The READ/WRITE bit is followed by an ACK/NACK bit (ACK=0 or NACK=1) response from the slave.

The TWI slave is able to listen for up to two addresses at the same time. Which addresses to listen for is configured in the ADDRESS registers and the CONFIG register.

The TWI slave will only acknowledge (ACK) the write command if the address presented by the master matches one of the addresses the slave is configured to listen for. The TWI slave will generate a WRITE event if it acknowledges the write command.

The TWI slave is only able to detect a write command from the IDLE state.

The TWI slave will set an internal 'RX prepared' flag when the PREPARERX task is triggered.

When the write command is received the TWI slave will enter the RX state if the internal 'RX prepared' flag is set.

If the internal 'RX prepared' flag is not set when the write command is received, the TWI slave will stretch the master's clock until the PREPARERX task is triggered and the internal 'RX prepared' flag is set.

The TWI slave will generate the RXSTARTED event and clear the internal 'RX prepared' flag ('unprepare RX') when it enters the RX state. In this state the TWI slave will be able to receive the bytes sent by the TWI master. The TWI slave will consume I_{RX} in this mode.

The TWI slave will go back to the IDLE state if the TWI slave receives a restart command when it is in the RX state.

The TWI slave is stopped when it receives the stop condition from the TWI master. A STOPPED event will be generated when the transaction has stopped. The TWI slave will clear the internal 'RX prepared' flag ('unprepare RX') and go back to the IDLE state when it has stopped.

The receive buffer is located in RAM at the address specified in the TXD.PTR register. The TWI slave will only be able to receive as many bytes as specified in the RXD.MAXCNT register. If the TWI master tries to send more bytes to the slave than the slave is able to receive,these bytes will be discarded and the bytes will be NACKed by the slave. If this happens, an ERROR event will be generated.

The EasyDMA configuration registers, see RXD.PTR etc., are latched when the RXSTARTED event is generated.

The TWI slave can be forced to stop by triggering the STOP task. A STOPPED event will be generated when the TWI slave has stopped. The TWI slave will clear the internal 'RX prepared' flag and go back to the IDLE state when it has stopped, see also Terminating an ongoing TWI transaction on page 317.

The TWI slave will generate an ACK after every byte received from the master. The RXD.AMOUNT register can be queried after a transaction to see how many bytes were received.

A typical TWI slave write command response is illustrated in The TWI slave responding to a write command on page 316. Occurrence 2 in the figure illustrates clock stretching performed by the TWI slave following a SUSPEND task.

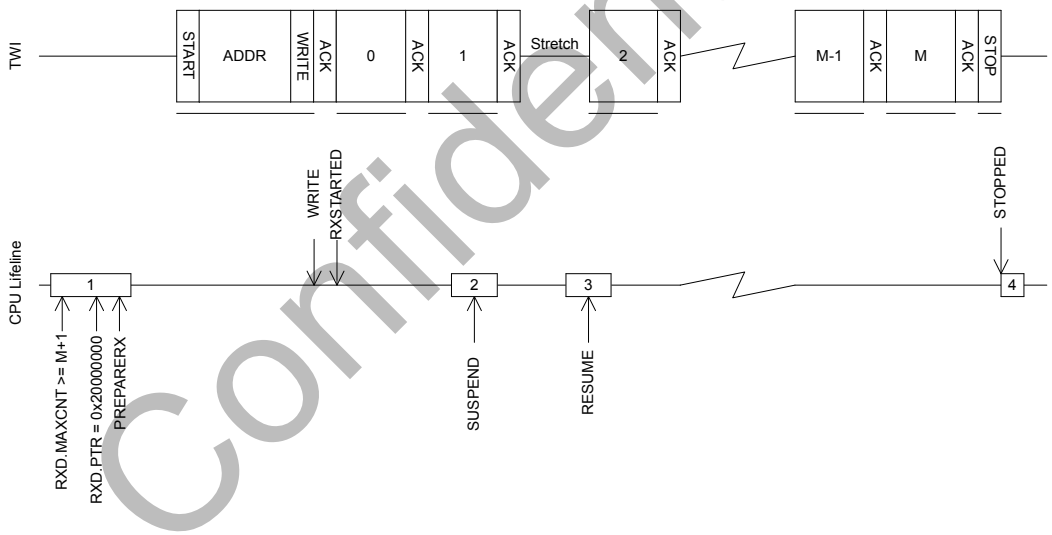

Figure 99: The TWI slave responding to a write command

6.20.4 Master repeated start sequence

An example of a repeated start sequence is one in which the TWI master writes two bytes to the slave followed by reading four bytes from the slave.

This is illustrated in A repeated start sequence, where the TWI master writes two bytes followed by reading four bytes from the slave on page 317.

It is here assumed that the receiver does not know in advance what the master wants to read, and that this information is provided in the first two bytes received in the write part of the repeated start sequence. To guarantee that the CPU is able to process the received data before the TWI slave starts to reply to the read command, the SUSPEND task is triggered via a shortcut from the READ event generated when the read command is received. When the CPU has processed the incoming data and prepared the correct data response, the CPU will resume the transaction by triggering the RESUME task.

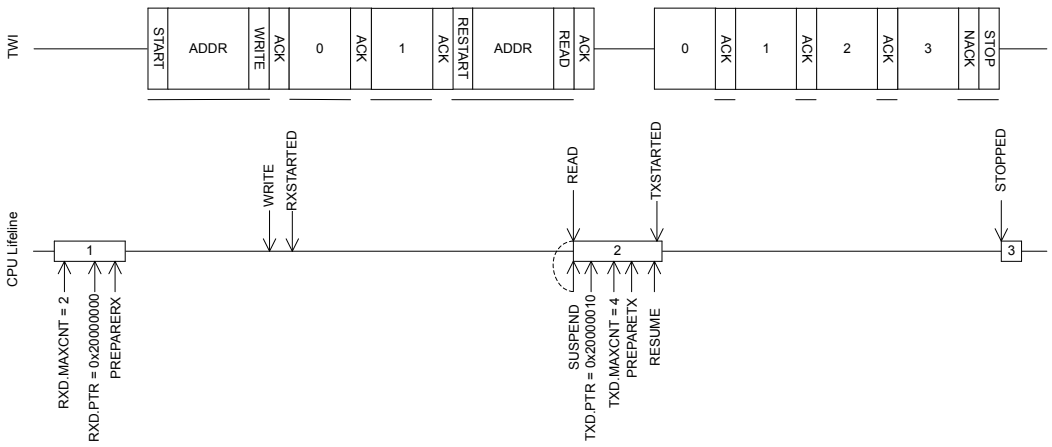

Figure 100: A repeated start sequence, where the TWI master writes two bytes followed by reading four bytes from the slave

6.20.5 Terminating an ongoing TWI transaction

In some situations, e.g. if the external TWI master is not responding correctly, it may be required to terminate an ongoing transaction.

This can be achieved by triggering the STOP task. In this situation a STOPPED event will be generated when the TWI has stopped independent of whether or not a STOP condition has been generated on the TWI bus. The TWI slave will release the bus when it has stopped and go back to its IDLE state.

6.20.6 Low power

When putting the system in low power and the peripheral is not needed, lowest possible power consumption is achieved by stopping, and then disabling the peripheral.

The STOP task may not be always needed (the peripheral might already be stopped), but if it is sent, software shall wait until the STOPPED event was received as a response before disabling the peripheral through the ENABLE register.

6.20.7 Slave mode pin configuration

The SCL and SDA signals associated with the TWI slave are mapped to physical pins according to the configuration specified in the PSEL.SCL and PSEL.SDA registers respectively.

The PSEL.SCL and PSEL.SDA registers and their configurations are only used as long as the TWI slave is enabled, and retained only as long as the device is in ON mode. When the peripheral is disabled, the pins will behave as regular GPIOs, and use the configuration in their respective OUT bit field and PIN_CNF[n] register. PSEL.SCL and PSEL.SDA must only be configured when the TWI slave is disabled.

To secure correct signal levels on the pins used by the TWI slave when the system is in OFF mode, and when the TWI slave is disabled, these pins must be configured in the GPIO peripheral as described in GPIO configuration before enabling peripheral on page 317.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

TWI slave signal	TWI slave pin	Direction	Output value	Drive strength
SCL	As specified in PSEL.SCL	Input	Not applicable	S0D1
SDA	As specified in PSEL.SDA	Input	Not applicable	S0D1

Table 87: GPIO configuration before enabling peripheral

6.20.8 Registers

Base address	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50008000 0x40008000	TWIS	TWIS0 : S TWIS0 : NS	US	SA	Two-wire interface slave 0	
0x50009000 0x40009000	TWIS	TWIS1 : S TWIS1 : NS	US	SA	Two-wire interface slave 1	
0x5000A000 0x4000A000	TWIS	TWIS2 : S TWIS2 : NS	US	SA	Two-wire interface slave 2	
0x5000B000 0x4000B000	TWIS	TWIS3 : S TWIS3 : NS	US	SA	Two-wire interface slave 3	

Table 88: Instances

Register	Offset	Security	Description
TASKS_STOP	0x014		Stop TWI transaction
TASKS_SUSPEND	0x01C		Suspend TWI transaction
TASKS_RESUME	0x020		Resume TWI transaction
TASKS_PREPARERX	0x030		Prepare the TWI slave to respond to a write command
TASKS_PREPARETX	0x034		Prepare the TWI slave to respond to a read command
SUBSCRIBE_STOP	0x094		Subscribe configuration for task STOP
SUBSCRIBE_SUSPEND	0x09C		Subscribe configuration for task SUSPEND
SUBSCRIBE_RESUME	0x0A0		Subscribe configuration for task RESUME
SUBSCRIBE_PREPARERX	0x0B0		Subscribe configuration for task PREPARERX
SUBSCRIBE_PREPARETX	0x0B4		Subscribe configuration for task PREPARETX
EVENTS_STOPPED	0x104		TWI stopped
EVENTS_ERROR	0x124		TWI error
EVENTS_RXSTARTED	0x14C		Receive sequence started
EVENTS_TXSTARTED	0x150		Transmit sequence started
EVENTS_WRITE	0x164		Write command received
EVENTS_READ	0x168		Read command received
PUBLISH_STOPPED	0x184		Publish configuration for event STOPPED
PUBLISH_ERROR	0x1A4		Publish configuration for event ERROR
PUBLISH_RXSTARTED	0x1CC		Publish configuration for event RXSTARTED
PUBLISH_TXSTARTED	0x1D0		Publish configuration for event TXSTARTED
PUBLISH_WRITE	0x1E4		Publish configuration for event WRITE
PUBLISH_READ	0x1E8		Publish configuration for event READ
SHORTS	0x200		Shortcuts between local events and tasks
INTEN	0x300		Enable or disable interrupt
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
ERRORSRC	0x4D0		Error source
MATCH	0x4D4		Status register indicating which address had a match
ENABLE	0x500		Enable TWIS
PSEL.SCL	0x508		Pin select for SCL signal
PSEL.SDA	0x50C		Pin select for SDA signal
RXD.PTR	0x534		RXD Data pointer
RXD.MAXCNT	0x538		Maximum number of bytes in RXD buffer
RXD.AMOUNT	0x53C		Number of bytes transferred in the last RXD transaction
TXD.PTR	0x544		TXD Data pointer
TXD.MAXCNT	0x548		Maximum number of bytes in TXD buffer
TXD.AMOUNT	0x54C		Number of bytes transferred in the last TXD transaction
ADDRESS[0]	0x588		TWI slave address 0

Register	Offset	Security	Description
ADDRESS[1]	0x58C		TWI slave address 1
CONFIG	0x594		Configuration register for the address match mechanism
ORC	0x5C0		Over-read character. Character sent out in case of an over-read of the transmit buffer.

Table 89: Register overview

6.20.8.1 TASKS_STOP

Address offset: 0x014

Stop TWI transaction

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_STOP			Stop TWI transaction
			Trigger	1	Trigger task

6.20.8.2 TASKS_SUSPEND

Address offset: 0x01C

Suspend TWI transaction

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_SUSPEND			Suspend TWI transaction
			Trigger	1	Trigger task

6.20.8.3 TASKS_RESUME

Address offset: 0x020

Resume TWI transaction

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_RESUME			Resume TWI transaction
			Trigger	1	Trigger task

6.20.8.4 TASKS_PREPARERX

Address offset: 0x030

Prepare the TWI slave to respond to a write command

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_PREPARERX			Prepare the TWI slave to respond to a write command
			Trigger	1	Trigger task

6.20.8.5 TASKS_PREPARETX

Address offset: 0x034

Prepare the TWI slave to respond to a read command

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_PREPARETX			Prepare the TWI slave to respond to a read command
			Trigger	1	Trigger task

6.20.8.6 SUBSCRIBE_STOP

Address offset: 0x094

Subscribe configuration for task STOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.20.8.7 SUBSCRIBE_SUSPEND

Address offset: 0x09C

Subscribe configuration for task SUSPEND

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task SUSPEND will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.20.8.8 SUBSCRIBE_RESUME

Address offset: 0x0A0

Subscribe configuration for task RESUME

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			B A A A A
Reset 0x00000000			**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task RESUME will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.20.8.9 SUBSCRIBE_PREPARERX

Address offset: 0x0B0

Subscribe configuration for task PREPARERX

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			B A A A A
Reset 0x00000000			**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task PREPARERX will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.20.8.10 SUBSCRIBE_PREPARETX

Address offset: 0x0B4

Subscribe configuration for task PREPARETX

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			B A A A A
Reset 0x00000000			**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task PREPARETX will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.20.8.11 EVENTS_STOPPED

Address offset: 0x104

TWI stopped

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			A
Reset 0x00000000			**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_STOPPED			TWI stopped
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.20.8.12 EVENTS_ERROR

Address offset: 0x124

TWI error

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_ERROR			TWI error
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.20.8.13 EVENTS_RXSTARTED

Address offset: 0x14C

Receive sequence started

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_RXSTARTED			Receive sequence started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.20.8.14 EVENTS_TXSTARTED

Address offset: 0x150

Transmit sequence started

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_TXSTARTED			Transmit sequence started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.20.8.15 EVENTS_WRITE

Address offset: 0x164

Write command received

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_WRITE			Write command received
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.20.8.16 EVENTS_READ

Address offset: 0x168

Read command received

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			A
Reset 0x00000000			**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_READ			Read command received
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.20.8.17 PUBLISH_STOPPED

Address offset: 0x184

Publish configuration for event STOPPED

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			B A A A A
Reset 0x00000000			**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event STOPPED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.20.8.18 PUBLISH_ERROR

Address offset: 0x1A4

Publish configuration for event ERROR

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			B A A A A
Reset 0x00000000			**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event ERROR will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.20.8.19 PUBLISH_RXSTARTED

Address offset: 0x1CC

Publish configuration for event RXSTARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event RXSTARTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.20.8.20 PUBLISH_TXSTARTED

Address offset: 0x1D0

Publish configuration for event TXSTARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event TXSTARTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.20.8.21 PUBLISH_WRITE

Address offset: 0x1E4

Publish configuration for event WRITE

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event WRITE will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.20.8.22 PUBLISH_READ

Address offset: 0x1E8

Publish configuration for event READ

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event READ will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.20.8.23 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	WRITE_SUSPEND			Shortcut between event WRITE and task SUSPEND
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
B	RW	READ_SUSPEND			Shortcut between event READ and task SUSPEND
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

6.20.8.24 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	STOPPED			Enable or disable interrupt for event STOPPED
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	ERROR			Enable or disable interrupt for event ERROR
			Disabled	0	Disable
			Enabled	1	Enable
E	RW	RXSTARTED			Enable or disable interrupt for event RXSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	TXSTARTED			Enable or disable interrupt for event TXSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
G	RW	WRITE			Enable or disable interrupt for event WRITE
			Disabled	0	Disable
			Enabled	1	Enable
H	RW	READ			Enable or disable interrupt for event READ
			Disabled	0	Disable
			Enabled	1	Enable

6.20.8.25 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	STOPPED			Write '1' to enable interrupt for event STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ERROR			Write '1' to enable interrupt for event ERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	RXSTARTED			Write '1' to enable interrupt for event RXSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	TXSTARTED			Write '1' to enable interrupt for event TXSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	WRITE			Write '1' to enable interrupt for event WRITE
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	READ			Write '1' to enable interrupt for event READ
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.20.8.26 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	STOPPED			Write '1' to disable interrupt for event STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ERROR			Write '1' to disable interrupt for event ERROR
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	RXSTARTED			Write '1' to disable interrupt for event RXSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	TXSTARTED			Write '1' to disable interrupt for event TXSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
			Enabled	1	Read: Enabled
G	RW	WRITE			Write '1' to disable interrupt for event WRITE
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	READ			Write '1' to disable interrupt for event READ
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.20.8.27 ERRORSRC

Address offset: 0x4D0

Error source

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	OVERFLOW			RX buffer overflow detected, and prevented
			NotDetected	0	Error did not occur
			Detected	1	Error occurred
B	RW	DNACK			NACK sent after receiving a data byte
			NotReceived	0	Error did not occur
			Received	1	Error occurred
C	RW	OVERREAD			TX buffer over-read detected, and prevented
			NotDetected	0	Error did not occur
			Detected	1	Error occurred

6.20.8.28 MATCH

Address offset: 0x4D4

Status register indicating which address had a match

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	R	MATCH		[0..1]	Which of the addresses in {ADDRESS} matched the incoming address

6.20.8.29 ENABLE

Address offset: 0x500

Enable TWIS

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable TWIS
			Disabled	0	Disable TWIS
			Enabled	9	Enable TWIS

6.20.8.30 PSEL.SCL

Address offset: 0x508

Pin select for SCL signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.20.8.31 PSEL.SDA

Address offset: 0x50C

Pin select for SDA signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.20.8.32 RXD.PTR

Address offset: 0x534

RXD Data pointer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	PTR			RXD Data pointer

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.20.8.33 RXD.MAXCNT

Address offset: 0x538

Maximum number of bytes in RXD buffer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0x1FFF]	Maximum number of bytes in RXD buffer

6.20.8.34 RXD.AMOUNT

Address offset: 0x53C

Number of bytes transferred in the last RXD transaction

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0x1FFF]	Number of bytes transferred in the last RXD transaction

6.20.8.35 TXD.PTR

Address offset: 0x544

TXD Data pointer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	PTR			TXD Data pointer

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.20.8.36 TXD.MAXCNT

Address offset: 0x548

Maximum number of bytes in TXD buffer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0x1FFF]	Maximum number of bytes in TXD buffer

6.20.8.37 TXD.AMOUNT

Address offset: 0x54C

Number of bytes transferred in the last TXD transaction

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0x1FFF]	Number of bytes transferred in the last TXD transaction

6.20.8.38 ADDRESS[n] (n=0..1)

Address offset: 0x588 + (n × 0x4)

TWI slave address n

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	ADDRESS			TWI slave address

6.20.8.39 CONFIG

Address offset: 0x594

Configuration register for the address match mechanism

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	B A
Reset 0x00000001	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**

ID	Acce	Field	Value ID	Value	Description
A-B	RW	ADDRESS[i] (i=0..1)			Enable or disable address matching on ADDRESS[i]
			Disabled	0	Disabled
			Enabled	1	Enabled

6.20.8.40 ORC

Address offset: 0x5C0

Over-read character. Character sent out in case of an over-read of the transmit buffer.

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	ORC			Over-read character. Character sent out in case of an over-read of the transmit buffer.

6.20.9 Electrical specification

6.20.9.1 TWIS slave timing specifications

Symbol	Description	Min.	Typ.	Max.	Units
$f_{TWIS,SCL}$	Bit rates for TWIS[16]	..	..	..	kbps
$t_{TWIS,START}$	Time from PREPARERX/PREPARETX task to ready to receive/ transmit	..	..	..	µs
t_{TWIS,SU_DAT}	Data setup time before positive edge on SCL – all modes	300			ns
t_{TWIS,HD_DAT}	Data hold time after negative edge on SCL – all modes	500			ns
$t_{TWIS,HD\_STA,100kbps}$	TWI slave hold time from for START condition (SDA low to SCL low), 100 kbps	5200			ns
$t_{TWIS,HD\_STA,400kbps}$	TWI slave hold time from for START condition (SDA low to SCL low), 400 kbps	1300			ns
$t_{TWIS,SU\_STO,100kbps}$	TWI slave setup time from SCL high to STOP condition, 100 kbps	5200			ns
$t_{TWIS,SU\_STO,400kbps}$	TWI slave setup time from SCL high to STOP condition, 400 kbps	1300			ns
$t_{TWIS,BUF,100kbps}$	TWI slave bus free time between STOP and START conditions, 100 kbps		4700		ns
$t_{TWIS,BUF,400kbps}$	TWI slave bus free time between STOP and START conditions, 400 kbps		1300		ns

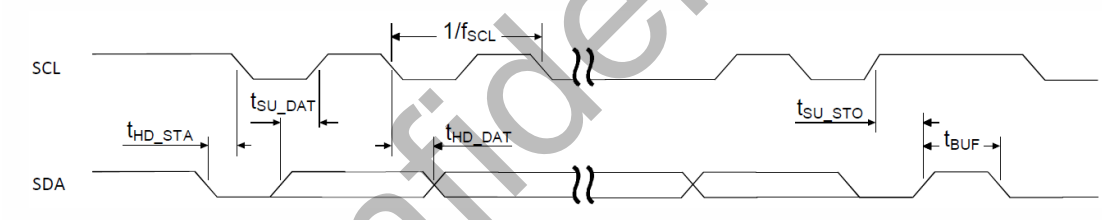

Figure 101: TWIS timing diagram, 1 byte transaction

6.21 UARTE — Universal asynchronous receiver/ transmitter with EasyDMA

The Universal asynchronous receiver/transmitter with EasyDMA (UARTE) offers fast, full-duplex, asynchronous serial communication with built-in flow control (CTS, RTS) support in hardware at a rate up to 1 Mbps, and EasyDMA data transfer from/to RAM.

Listed here are the main features for UARTE:

- Full-duplex operation
- Automatic hardware flow control
- Optional even parity bit checking and generation
- EasyDMA
- Up to 1 Mbps baudrate
- Return to IDLE between transactions supported (when using HW flow control)
- One or two stop bit

[16] High bit rates or stronger pull-ups may require GPIOs to be set as High Drive, see GPIO chapter for more details.

- Least significant bit (LSB) first

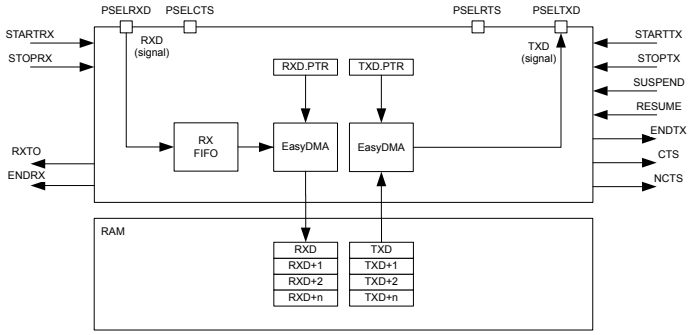

Figure 102: UARTE configuration

The GPIOs used for each UART interface can be chosen from any GPIO on the device and are independently configurable. This enables great flexibility in device pinout and efficient use of board space and signal routing.

6.21.1 EasyDMA

The UARTE implements EasyDMA for reading and writing to and from the RAM.

If the TXD.PTR and the RXD.PTR are not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 21 for more information about the different memory regions.

The .PTR and .MAXCNT registers are double-buffered. They can be updated and prepared for the next RX/TX transmission immediately after having received the RXSTARTED/TXSTARTED event.

The ENDRX/ENDTX event indicates that EasyDMA has finished accessing respectively the RX/TX buffer in RAM.

6.21.2 Transmission

The first step of a DMA transmission is storing bytes in the transmit buffer and configuring EasyDMA. This is achieved by writing the initial address pointer to TXD.PTR, and the number of bytes in the RAM buffer to TXD.MAXCNT. The UARTE transmission is started by triggering the STARTTX task.

After each byte has been sent over the TXD line, a TXDRDY event will be generated.

When all bytes in the TXD buffer, as specified in the TXD.MAXCNT register, have been transmitted, the UARTE transmission will end automatically and an ENDTX event will be generated.

A UARTE transmission sequence is stopped by triggering the STOPTX task, a TXSTOPPED event will be generated when the UARTE transmitter has stopped.

If the ENDTX event has not already been generated when the UARTE transmitter has come to a stop, the UARTE will generate the ENDTX event explicitly even though all bytes in the TXD buffer, as specified in the TXD.MAXCNT register, have not been transmitted.

If flow control is enabled through the HWFC field in the CONFIG register, a transmission will be automatically suspended when CTS is deactivated and resumed when CTS is activated again, as illustrated in UARTE transmission on page 333. A byte that is in transmission when CTS is deactivated will be fully transmitted before the transmission is suspended.

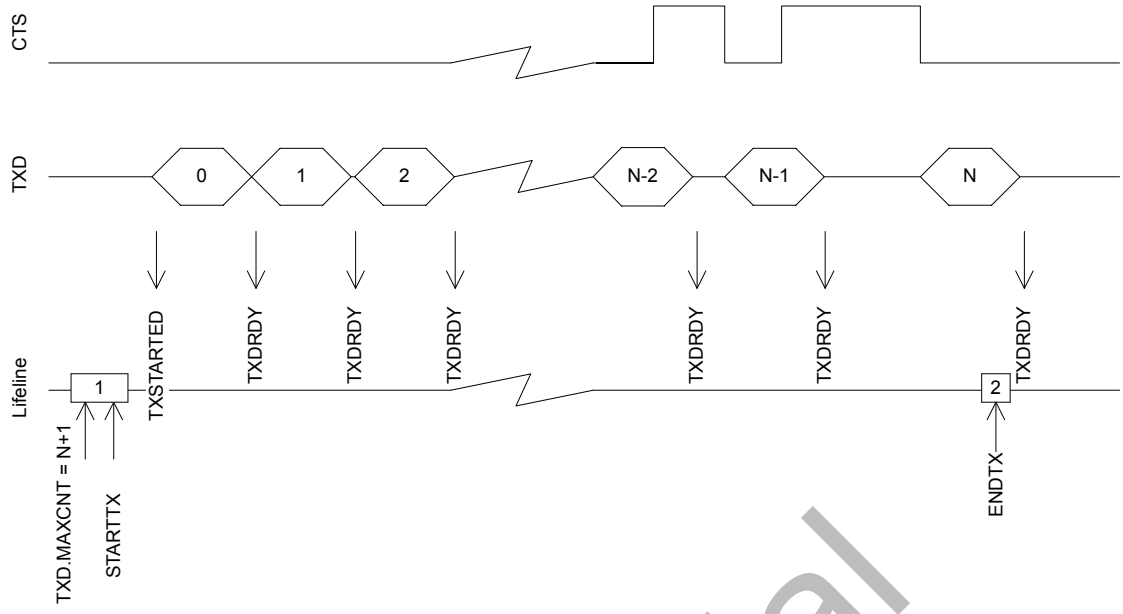

Figure 103: UARTE transmission

The UARTE transmitter will be in its lowest activity level, and consume the least amount of energy, when it is stopped, i.e. before it is started via STARTTX or after it has been stopped via STOPTX and the TXSTOPPED event has been generated. See POWER — Power control on page 60 for more information about power modes.

6.21.3 Reception

The UARTE receiver is started by triggering the STARTRX task. The UARTE receiver is using EasyDMA to store incoming data in an RX buffer in RAM.

The RX buffer is located at the address specified in the RXD.PTR register. The RXD.PTR register is double-buffered and it can be updated and prepared for the next STARTRX task immediately after the RXSTARTED event is generated. The size of the RX buffer is specified in the RXD.MAXCNT register and the UARTE will generate an ENDRX event when it has filled up the RX buffer, see UARTE reception on page 334.

For each byte received over the RXD line, an RXDRDY event will be generated. This event is likely to occur before the corresponding data has been transferred to Data RAM.

The RXD.AMOUNT register can be queried following an ENDRX event to see how many new bytes have been transferred to the RX buffer in RAM since the previous ENDRX event.

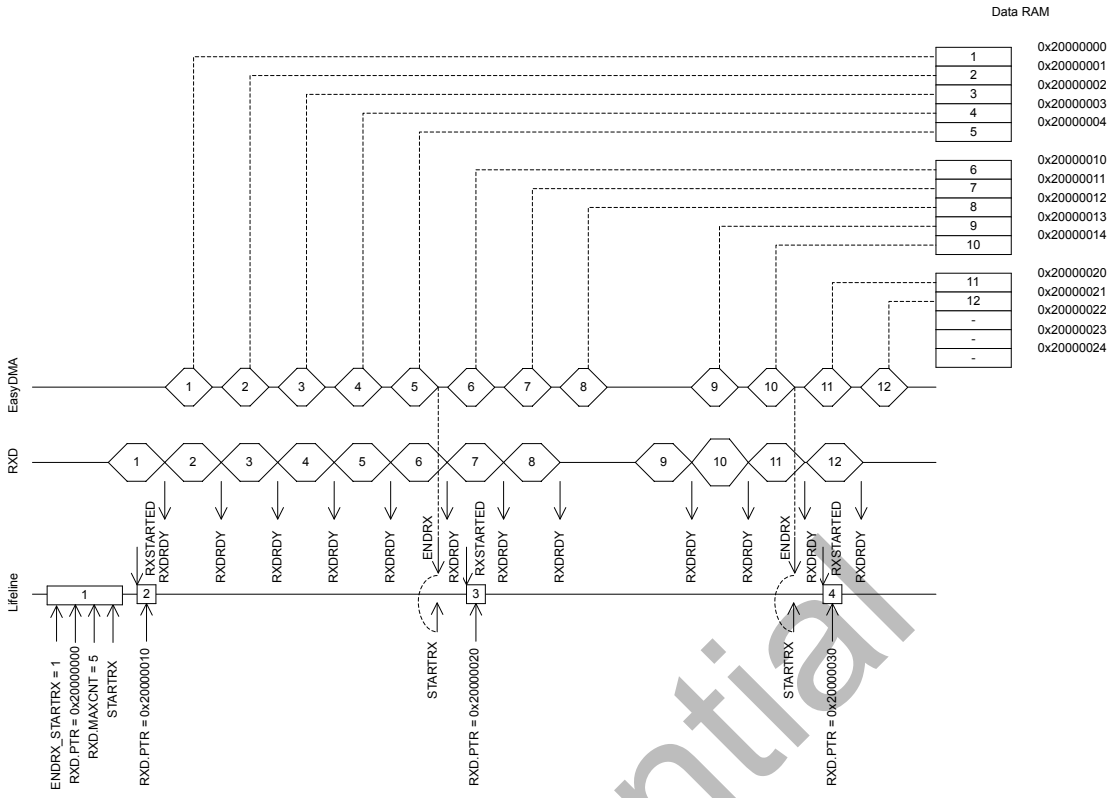

Figure 104: UARTE reception

The UARTE receiver is stopped by triggering the STOPRX task. An RXTO event is generated when the UARTE has stopped. The UARTE will make sure that an impending ENDRX event will be generated before the RXTO event is generated. This means that the UARTE will guarantee that no ENDRX event will be generated after RXTO, unless the UARTE is restarted or a FLUSHRX command is issued after the RXTO event is generated.

Important: If the ENDRX event has not already been generated when the UARTE receiver has come to a stop, which implies that all pending content in the RX FIFO has been moved to the RX buffer, the UARTE will generate the ENDRX event explicitly even though the RX buffer is not full. In this scenario the ENDRX event will be generated before the RXTO event is generated.

To be able to know how many bytes have actually been received into the RX buffer, the CPU can read the RXD.AMOUNT register following the ENDRX event or the RXTO event.

The UARTE is able to receive up to four bytes after the STOPRX task has been triggered as long as these are sent in succession immediately after the RTS signal is deactivated. This is possible because after the RTS is deactivated the UARTE is able to receive bytes for an extended period equal to the time it takes to send 4 bytes on the configured baud rate.

After the RXTO event is generated the internal RX FIFO may still contain data, and to move this data to RAM the FLUSHRX task must be triggered. To make sure that this data does not overwrite data in the RX buffer, the RX buffer should be emptied or the RXD.PTR should be updated before the FLUSHRX task is triggered. To make sure that all data in the RX FIFO is moved to the RX buffer, the RXD.MAXCNT register must be set to RXD.MAXCNT > 4, see UARTE reception with forced stop via STOPRX on page 335. The UARTE will generate the ENDRX event after completing the FLUSHRX task even if the RX FIFO was empty or if the RX buffer does not get filled up. To be able to know how many bytes have actually been received into the RX buffer in this case, the CPU can read the RXD.AMOUNT register following the ENDRX event.

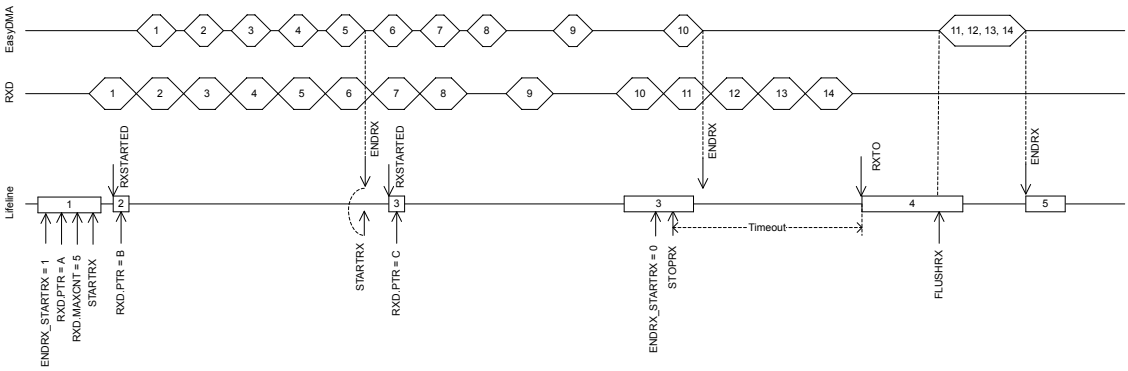

Figure 105: UARTE reception with forced stop via STOPRX

If HW flow control is enabled through the HWFC field in the CONFIG register, the RTS signal will be deactivated when the receiver is stopped via the STOPRX task or when the UARTE is only able to receive four more bytes in its internal RX FIFO.

With flow control disabled, the UARTE will function in the same way as when the flow control is enabled except that the RTS line will not be used. This means that no signal will be generated when the UARTE has reached the point where it is only able to receive four more bytes in its internal RX FIFO. Data received when the internal RX FIFO is filled up, will be lost.

The UARTE receiver will be in its lowest activity level, and consume the least amount of energy, when it is stopped, i.e. before it is started via STARTRX or after it has been stopped via STOPRX and the RXTO event has been generated. See POWER — Power control on page 60 for more information about power modes.

6.21.4 Error conditions

An ERROR event, in the form of a framing error, will be generated if a valid stop bit is not detected in a frame. Another ERROR event, in the form of a break condition, will be generated if the RXD line is held active low for longer than the length of a data frame. Effectively, a framing error is always generated before a break condition occurs.

An ERROR event will not stop reception. If the error was a parity error, the received byte will still be transferred into Data RAM, and so will following incoming bytes. If there was a framing error (wrong stop bit), that specific byte will NOT be stored into Data RAM, but following incoming bytes will.

6.21.5 Using the UARTE without flow control

If flow control is not enabled, the interface will behave as if the CTS and RTS lines are kept active all the time.

6.21.6 Parity and stop bit configuration

When parity is enabled through the PARITY field in the CONFIG register, the parity will be generated automatically from the even parity of TXD and RXD for transmission and reception respectively.

The amount of stop bits can be configured through the STOP field in the CONFIG register.

6.21.7 Low power

When putting the system in low power and the peripheral is not needed, lowest possible power consumption is achieved by stopping, and then disabling the peripheral.

The STOPTX and STOPRX tasks may not be always needed (the peripheral might already be stopped), but if STOPTX and/or STOPRX is sent, software shall wait until the TXSTOPPED and/or RXTO event is received in response, before disabling the peripheral through the ENABLE register.

6.21.8 Pin configuration

The different signals RXD, CTS (Clear To Send, active low), RTS (Request To Send, active low), and TXD associated with the UARTE are mapped to physical pins according to the configuration specified in the PSEL.RXD, PSEL.CTS, PSEL.RTS, and PSEL.TXD registers respectively.

The PSEL.RXD, PSEL.CTS, PSEL.RTS, and PSEL.TXD registers and their configurations are only used as long as the UARTE is enabled, and retained only for the duration the device is in ON mode. PSEL.RXD, PSEL.RTS, PSEL.RTS and PSEL.TXD must only be configured when the UARTE is disabled.

To secure correct signal levels on the pins by the UARTE when the system is in OFF mode, the pins must be configured in the GPIO peripheral as described in GPIO configuration before enabling peripheral on page 336.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

UARTE signal	UARTE pin	Direction	Output value
RXD	As specified in PSEL.RXD	Input	Not applicable
CTS	As specified in PSEL.CTS	Input	Not applicable
RTS	As specified in PSEL.RTS	Output	1
TXD	As specified in PSEL.TXD	Output	1

Table 90: GPIO configuration before enabling peripheral

6.21.9 Registers

Base address	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50008000 0x40008000	UARTE	UARTE0 : S UARTE0 : NS	US	SA	Universal asynchronous receiver/transmitter with EasyDMA 0	
0x50009000 0x40009000	UARTE	UARTE1 : S UARTE1 : NS	US	SA	Universal asynchronous receiver/transmitter with EasyDMA 1	
0x5000A000 0x4000A000	UARTE	UARTE2 : S UARTE2 : NS	US	SA	Universal asynchronous receiver/transmitter with EasyDMA 2	
0x5000B000 0x4000B000	UARTE	UARTE3 : S UARTE3 : NS	US	SA	Universal asynchronous receiver/transmitter with EasyDMA 3	

Table 91: Instances

Register	Offset	Security	Description
TASKS_STARTRX	0x000		Start UART receiver
TASKS_STOPRX	0x004		Stop UART receiver
TASKS_STARTTX	0x008		Start UART transmitter
TASKS_STOPTX	0x00C		Stop UART transmitter
TASKS_FLUSHRX	0x02C		Flush RX FIFO into RX buffer
SUBSCRIBE_STARTRX	0x080		Subscribe configuration for task STARTRX
SUBSCRIBE_STOPRX	0x084		Subscribe configuration for task STOPRX
SUBSCRIBE_STARTTX	0x088		Subscribe configuration for task STARTTX
SUBSCRIBE_STOPTX	0x08C		Subscribe configuration for task STOPTX
SUBSCRIBE_FLUSHRX	0x0AC		Subscribe configuration for task FLUSHRX
EVENTS_CTS	0x100		CTS is activated (set low). Clear To Send.

Register	Offset	Security	Description
EVENTS_NCTS	0x104		CTS is deactivated (set high). Not Clear To Send.
EVENTS_RXDRDY	0x108		Data received in RXD (but potentially not yet transferred to Data RAM)
EVENTS_ENDRX	0x110		Receive buffer is filled up
EVENTS_TXDRDY	0x11C		Data sent from TXD
EVENTS_ENDTX	0x120		Last TX byte transmitted
EVENTS_ERROR	0x124		Error detected
EVENTS_RXTO	0x144		Receiver timeout
EVENTS_RXSTARTED	0x14C		UART receiver has started
EVENTS_TXSTARTED	0x150		UART transmitter has started
EVENTS_TXSTOPPED	0x158		Transmitter stopped
PUBLISH_CTS	0x180		Publish configuration for event CTS
PUBLISH_NCTS	0x184		Publish configuration for event NCTS
PUBLISH_RXDRDY	0x188		Publish configuration for event RXDRDY
PUBLISH_ENDRX	0x190		Publish configuration for event ENDRX
PUBLISH_TXDRDY	0x19C		Publish configuration for event TXDRDY
PUBLISH_ENDTX	0x1A0		Publish configuration for event ENDTX
PUBLISH_ERROR	0x1A4		Publish configuration for event ERROR
PUBLISH_RXTO	0x1C4		Publish configuration for event RXTO
PUBLISH_RXSTARTED	0x1CC		Publish configuration for event RXSTARTED
PUBLISH_TXSTARTED	0x1D0		Publish configuration for event TXSTARTED
PUBLISH_TXSTOPPED	0x1D8		Publish configuration for event TXSTOPPED
SHORTS	0x200		Shortcuts between local events and tasks
INTEN	0x300		Enable or disable interrupt
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
ERRORSRC	0x480		Error source Note : this register is read / write one to clear.
ENABLE	0x500		Enable UART
PSEL.RTS	0x508		Pin select for RTS signal
PSEL.TXD	0x50C		Pin select for TXD signal
PSEL.CTS	0x510		Pin select for CTS signal
PSEL.RXD	0x514		Pin select for RXD signal
BAUDRATE	0x524		Baud rate. Accuracy depends on the HFCLK source selected.
RXD.PTR	0x534		Data pointer
RXD.MAXCNT	0x538		Maximum number of bytes in receive buffer
RXD.AMOUNT	0x53C		Number of bytes transferred in the last transaction
TXD.PTR	0x544		Data pointer
TXD.MAXCNT	0x548		Maximum number of bytes in transmit buffer
TXD.AMOUNT	0x54C		Number of bytes transferred in the last transaction
CONFIG	0x56C		Configuration of parity and hardware flow control

Table 92: Register overview

6.21.9.1 TASKS_STARTRX

Address offset: 0x000

Start UART receiver

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_STARTRX			Start UART receiver
			Trigger	1	Trigger task

6.21.9.2 TASKS_STOPRX

Address offset: 0x004

Stop UART receiver

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_STOPRX			Stop UART receiver
			Trigger	1	Trigger task

6.21.9.3 TASKS_STARTTX

Address offset: 0x008

Start UART transmitter

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_STARTTX			Start UART transmitter
			Trigger	1	Trigger task

6.21.9.4 TASKS_STOPTX

Address offset: 0x00C

Stop UART transmitter

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_STOPTX			Stop UART transmitter
			Trigger	1	Trigger task

6.21.9.5 TASKS_FLUSHRX

Address offset: 0x02C

Flush RX FIFO into RX buffer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_FLUSHRX			Flush RX FIFO into RX buffer
			Trigger	1	Trigger task

6.21.9.6 SUBSCRIBE_STARTRX

Address offset: 0x080

Subscribe configuration for task STARTRX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task STARTRX will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.21.9.7 SUBSCRIBE_STOPRX

Address offset: 0x084

Subscribe configuration for task STOPRX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task STOPRX will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.21.9.8 SUBSCRIBE_STARTTX

Address offset: 0x088

Subscribe configuration for task STARTTX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task STARTTX will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.21.9.9 SUBSCRIBE_STOPTX

Address offset: 0x08C

Subscribe configuration for task STOPTX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task STOPTX will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.21.9.10 SUBSCRIBE_FLUSHRX

Address offset: 0x0AC

Subscribe configuration for task FLUSHRX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task FLUSHRX will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.21.9.11 EVENTS_CTS

Address offset: 0x100

CTS is activated (set low). Clear To Send.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_CTS			CTS is activated (set low). Clear To Send.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.21.9.12 EVENTS_NCTS

Address offset: 0x104

CTS is deactivated (set high). Not Clear To Send.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_NCTS			CTS is deactivated (set high). Not Clear To Send.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.21.9.13 EVENTS_RXDRDY

Address offset: 0x108

Data received in RXD (but potentially not yet transferred to Data RAM)

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			A
Reset 0x00000000			**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_RXDRDY			Data received in RXD (but potentially not yet transferred to Data RAM)
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.21.9.14 EVENTS_ENDRX

Address offset: 0x110

Receive buffer is filled up

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			A
Reset 0x00000000			**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_ENDRX			Receive buffer is filled up
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.21.9.15 EVENTS_TXDRDY

Address offset: 0x11C

Data sent from TXD

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			A
Reset 0x00000000			**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_TXDRDY			Data sent from TXD
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.21.9.16 EVENTS_ENDTX

Address offset: 0x120

Last TX byte transmitted

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_ENDTX			Last TX byte transmitted
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.21.9.17 EVENTS_ERROR

Address offset: 0x124

Error detected

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_ERROR			Error detected
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.21.9.18 EVENTS_RXTO

Address offset: 0x144

Receiver timeout

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_RXTO			Receiver timeout
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.21.9.19 EVENTS_RXSTARTED

Address offset: 0x14C

UART receiver has started

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_RXSTARTED			UART receiver has started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.21.9.20 EVENTS_TXSTARTED

Address offset: 0x150

UART transmitter has started

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			A
Reset 0x00000000			**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_TXSTARTED			UART transmitter has started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.21.9.21 EVENTS_TXSTOPPED

Address offset: 0x158

Transmitter stopped

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			A
Reset 0x00000000			**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_TXSTOPPED			Transmitter stopped
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.21.9.22 PUBLISH_CTS

Address offset: 0x180

Publish configuration for event CTS

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			B A A A A
Reset 0x00000000			**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event CTS will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.21.9.23 PUBLISH_NCTS

Address offset: 0x184

Publish configuration for event NCTS

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			B A A A A
Reset 0x00000000			**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event NCTS will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.21.9.24 PUBLISH_RXDRDY

Address offset: 0x188

Publish configuration for event RXDRDY

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event RXDRDY will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.21.9.25 PUBLISH_ENDRX

Address offset: 0x190

Publish configuration for event ENDRX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event ENDRX will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.21.9.26 PUBLISH_TXDRDY

Address offset: 0x19C

Publish configuration for event TXDRDY

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event TXDRDY will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.21.9.27 PUBLISH_ENDTX

Address offset: 0x1A0

Publish configuration for event ENDTX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event ENDTX will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.21.9.28 PUBLISH_ERROR

Address offset: 0x1A4

Publish configuration for event ERROR

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event ERROR will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.21.9.29 PUBLISH_RXTO

Address offset: 0x1C4

Publish configuration for event RXTO

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event RXTO will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.21.9.30 PUBLISH_RXSTARTED

Address offset: 0x1CC

Publish configuration for event RXSTARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event RXSTARTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.21.9.31 PUBLISH_TXSTARTED

Address offset: 0x1D0

Publish configuration for event TXSTARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event TXSTARTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.21.9.32 PUBLISH_TXSTOPPED

Address offset: 0x1D8

Publish configuration for event TXSTOPPED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event TXSTOPPED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.21.9.33 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D C
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
C	RW	ENDRX_STARTRX			Shortcut between event ENDRX and task STARTRX
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
D	RW	ENDRX_STOPRX			Shortcut between event ENDRX and task STOPRX
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

6.21.9.34 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CTS			Enable or disable interrupt for event CTS
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	NCTS			Enable or disable interrupt for event NCTS
			Disabled	0	Disable
			Enabled	1	Enable
C	RW	RXDRDY			Enable or disable interrupt for event RXDRDY
			Disabled	0	Disable
			Enabled	1	Enable
D	RW	ENDRX			Enable or disable interrupt for event ENDRX
			Disabled	0	Disable
			Enabled	1	Enable
E	RW	TXDRDY			Enable or disable interrupt for event TXDRDY
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	ENDTX			Enable or disable interrupt for event ENDTX
			Disabled	0	Disable
			Enabled	1	Enable
G	RW	ERROR			Enable or disable interrupt for event ERROR
			Disabled	0	Disable
			Enabled	1	Enable
H	RW	RXTO			Enable or disable interrupt for event RXTO
			Disabled	0	Disable
			Enabled	1	Enable
I	RW	RXSTARTED			Enable or disable interrupt for event RXSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
J	RW	TXSTARTED			Enable or disable interrupt for event TXSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
L	RW	TXSTOPPED			Enable or disable interrupt for event TXSTOPPED
			Disabled	0	Disable
			Enabled	1	Enable

6.21.9.35 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CTS			Write '1' to enable interrupt for event CTS
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	NCTS			Write '1' to enable interrupt for event NCTS
			Set	1	Enable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	RXDRDY			Write '1' to enable interrupt for event RXDRDY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	ENDRX			Write '1' to enable interrupt for event ENDRX
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	TXDRDY			Write '1' to enable interrupt for event TXDRDY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	ENDTX			Write '1' to enable interrupt for event ENDTX
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	ERROR			Write '1' to enable interrupt for event ERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	RXTO			Write '1' to enable interrupt for event RXTO
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	RXSTARTED			Write '1' to enable interrupt for event RXSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	TXSTARTED			Write '1' to enable interrupt for event TXSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	TXSTOPPED			Write '1' to enable interrupt for event TXSTOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.21.9.36 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CTS			Write '1' to disable interrupt for event CTS

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	L J I H G F E D C B A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	NCTS			Write '1' to disable interrupt for event NCTS
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	RXDRDY			Write '1' to disable interrupt for event RXDRDY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	ENDRX			Write '1' to disable interrupt for event ENDRX
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	TXDRDY			Write '1' to disable interrupt for event TXDRDY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	ENDTX			Write '1' to disable interrupt for event ENDTX
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	ERROR			Write '1' to disable interrupt for event ERROR
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	RXTO			Write '1' to disable interrupt for event RXTO
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	RXSTARTED			Write '1' to disable interrupt for event RXSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	TXSTARTED			Write '1' to disable interrupt for event TXSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	TXSTOPPED			Write '1' to disable interrupt for event TXSTOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.21.9.37 ERRORSRC

Address offset: 0x480

Error source

Note : this register is read / write one to clear.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	OVERRUN			Overrun error A start bit is received while the previous data still lies in RXD. (Previous data is lost.)
			NotPresent	0	Read: error not present
			Present	1	Read: error present
B	RW	PARITY			Parity error A character with bad parity is received, if HW parity check is enabled.
			NotPresent	0	Read: error not present
			Present	1	Read: error present
C	RW	FRAMING			Framing error occurred A valid stop bit is not detected on the serial data input after all bits in a character have been received.
			NotPresent	0	Read: error not present
			Present	1	Read: error present
D	RW	BREAK			Break condition The serial data input is '0' for longer than the length of a data frame. (The data frame length is 10 bits without parity bit, and 11 bits with parity bit.).
			NotPresent	0	Read: error not present
			Present	1	Read: error present

6.21.9.38 ENABLE

Address offset: 0x500

Enable UART

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable UARTE
			Disabled	0	Disable UARTE
			Enabled	8	Enable UARTE

6.21.9.39 PSEL.RTS

Address offset: 0x508

Pin select for RTS signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.21.9.40 PSEL.TXD

Address offset: 0x50C

Pin select for TXD signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.21.9.41 PSEL.CTS

Address offset: 0x510

Pin select for CTS signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.21.9.42 PSEL.RXD

Address offset: 0x514

Pin select for RXD signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.21.9.43 BAUDRATE

Address offset: 0x524

Baud rate. Accuracy depends on the HFCLK source selected.

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x04000000			**0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	BAUDRATE			Baud rate
			Baud1200	0x0004F000	1200 baud (actual rate: 1205)
			Baud2400	0x0009D000	2400 baud (actual rate: 2396)
			Baud4800	0x0013B000	4800 baud (actual rate: 4808)
			Baud9600	0x00275000	9600 baud (actual rate: 9598)
			Baud14400	0x003AF000	14400 baud (actual rate: 14401)
			Baud19200	0x004EA000	19200 baud (actual rate: 19208)
			Baud28800	0x0075C000	28800 baud (actual rate: 28777)
			Baud31250	0x00800000	31250 baud
			Baud38400	0x009D0000	38400 baud (actual rate: 38369)
			Baud56000	0x00E50000	56000 baud (actual rate: 55944)
			Baud57600	0x00EB0000	57600 baud (actual rate: 57554)
			Baud76800	0x013A9000	76800 baud (actual rate: 76923)
			Baud115200	0x01D60000	115200 baud (actual rate: 115108)
			Baud230400	0x03B00000	230400 baud (actual rate: 231884)
			Baud250000	0x04000000	250000 baud
			Baud460800	0x07400000	460800 baud (actual rate: 457143)
			Baud921600	0x0F000000	921600 baud (actual rate: 941176)
			Baud1M	0x10000000	1Mega baud

6.21.9.44 RXD.PTR

Address offset: 0x534

Data pointer

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000			**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	PTR			Data pointer

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.21.9.45 RXD.MAXCNT

Address offset: 0x538

Maximum number of bytes in receive buffer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0x1FFF]	Maximum number of bytes in receive buffer

6.21.9.46 RXD.AMOUNT

Address offset: 0x53C

Number of bytes transferred in the last transaction

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0x1FFF]	Number of bytes transferred in the last transaction

6.21.9.47 TXD.PTR

Address offset: 0x544

Data pointer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	PTR			Data pointer

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.21.9.48 TXD.MAXCNT

Address offset: 0x548

Maximum number of bytes in transmit buffer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0x1FFF]	Maximum number of bytes in transmit buffer

6.21.9.49 TXD.AMOUNT

Address offset: 0x54C

Number of bytes transferred in the last transaction

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0x1FFF]	Number of bytes transferred in the last transaction

6.21.9.50 CONFIG

Address offset: 0x56C

Configuration of parity and hardware flow control

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B B B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	HWFC			Hardware flow control
			Disabled	0	Disabled
			Enabled	1	Enabled
B	RW	PARITY			Parity
			Excluded	0x0	Exclude parity bit
			Included	0x7	Include even parity bit
C	RW	STOP			Stop bits
			One	0	One stop bit
			Two	1	Two stop bits

6.21.10 Electrical specification

6.21.10.1 UARTE electrical specification

Symbol	Description	Min.	Typ.	Max.	Units
f_{UARTE}	Baud rate for UARTE[17].			1000	kbps
$t_{UARTE,CTSH}$	CTS high time	1			µs
$t_{UARTE,START}$	Time from STARTRX/STARTTX task to transmission started	..	..	..	µs

6.22 WDT — Watchdog timer

A countdown watchdog timer using the low-frequency clock source (LFCLK) offers configurable and robust protection against application lock-up.

The watchdog timer is started by triggering the START task.

The watchdog can be paused during long CPU sleep periods for low power applications and when the debugger has halted the CPU. The watchdog is implemented as a down-counter that generates a TIMEOUT event when it wraps over after counting down to 0. When the watchdog timer is started through the START task, the watchdog counter is loaded with the value specified in the CRV register. This counter is also reloaded with the value specified in the CRV register when a reload request is granted.

[17] High baud rates may require GPIOs to be set as High Drive, see GPIO chapter for more details.

The watchdog's timeout period is given by:

```
timeout [s] = ( CRV + 1 ) / 32768
```

When started, the watchdog will automatically force the 32.768 kHz RC oscillator on as long as no other 32.768 kHz clock source is running and generating the 32.768 kHz system clock, see chapter CLOCK — Clock control on page 66.

6.22.1 Reload criteria

The watchdog has eight separate reload request registers, which shall be used to request the watchdog to reload its counter with the value specified in the CRV register. To reload the watchdog counter, the special value 0x6E524635 needs to be written to all enabled reload registers.

One or more RR registers can be individually enabled through the RREN register.

6.22.2 Temporarily pausing the watchdog

By default, the watchdog will be active counting down the down-counter while the CPU is sleeping and when it is halted by the debugger. It is however possible to configure the watchdog to automatically pause while the CPU is sleeping as well as when it is halted by the debugger.

6.22.3 Watchdog reset

A TIMEOUT event will automatically lead to a watchdog reset.

See Reset on page 55 for more information about reset sources. If the watchdog is configured to generate an interrupt on the TIMEOUT event, the watchdog reset will be postponed with two 32.768 kHz clock cycles after the TIMEOUT event has been generated. Once the TIMEOUT event has been generated, the impending watchdog reset will always be effectuated.

The watchdog must be configured before it is started. After it is started, the watchdog's configuration registers, which comprise registers CRV, RREN, and CONFIG, will be blocked for further configuration.

The watchdog can be reset from several reset sources, see Reset behavior on page 56.

When the device starts running again, after a reset, or waking up from OFF mode, the watchdog configuration registers will be available for configuration again.

6.22.4 Registers

Base address	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50018000	WDT	WDT : S	US	NA	Watchdog timer	
0x40018000		WDT : NS				

Table 93: Instances

Register	Offset	Security	Description
TASKS_START	0x000		Start the watchdog
SUBSCRIBE_START	0x080		Subscribe configuration for task START
EVENTS_TIMEOUT	0x100		Watchdog timeout
PUBLISH_TIMEOUT	0x180		Publish configuration for event TIMEOUT
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
RUNSTATUS	0x400		Run status

Register	Offset	Security	Description
REQSTATUS	0x404		Request status
CRV	0x504		Counter reload value
RREN	0x508		Enable register for reload request registers
CONFIG	0x50C		Configuration register
RR[0]	0x600		Reload request 0
RR[1]	0x604		Reload request 1
RR[2]	0x608		Reload request 2
RR[3]	0x60C		Reload request 3
RR[4]	0x610		Reload request 4
RR[5]	0x614		Reload request 5
RR[6]	0x618		Reload request 6
RR[7]	0x61C		Reload request 7

Table 94: Register overview

6.22.4.1 TASKS_START

Address offset: 0x000

Start the watchdog

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_START			Start the watchdog
			Trigger	1	Trigger task

6.22.4.2 SUBSCRIBE_START

Address offset: 0x080

Subscribe configuration for task START

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					B A A A A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.22.4.3 EVENTS_TIMEOUT

Address offset: 0x100

Watchdog timeout

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_TIMEOUT			Watchdog timeout
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.22.4.4 PUBLISH_TIMEOUT

Address offset: 0x180

Publish configuration for event TIMEOUT

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event TIMEOUT will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.22.4.5 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	TIMEOUT			Write '1' to enable interrupt for event TIMEOUT
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.22.4.6 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	TIMEOUT			Write '1' to disable interrupt for event TIMEOUT
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.22.4.7 RUNSTATUS

Address offset: 0x400

Run status

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	R	RUNSTATUSWDT			Indicates whether or not the watchdog is running
			NotRunning	0	Watchdog not running
			Running	1	Watchdog is running

6.22.4.8 REQSTATUS

Address offset: 0x404

Request status

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E D C B A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	Acce	Field	Value ID	Value	Description
A-H	R	RR[i] (i=0..7)			Request status for RR[i] register
			DisabledOrRequested	0	RR[i] register is not enabled, or are already requesting reload
			EnabledAndUnrequested	1	RR[i] register is enabled, and are not yet requesting reload

6.22.4.9 CRV

Address offset: 0x504

Counter reload value

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	Acce	Field	Value ID	Value	Description
A	RW	CRV		[0x0000000F..0xFFFFFFFF]	Counter reload value in number of cycles of the 32.768 kHz clock

6.22.4.10 RREN

Address offset: 0x508

Enable register for reload request registers

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E D C B A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	Acce	Field	Value ID	Value	Description
A-H	RW	RR[i] (i=0..7)			Enable or disable RR[i] register
			Disabled	0	Disable RR[i] register
			Enabled	1	Enable RR[i] register

6.22.4.11 CONFIG

Address offset: 0x50C

Configuration register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	Acce	Field	Value ID	Value	Description
A	RW	SLEEP			Configure the watchdog to either be paused, or kept running, while the CPU is sleeping
			Pause	0	Pause watchdog while the CPU is sleeping
			Run	1	Keep the watchdog running while the CPU is sleeping
C	RW	HALT			Configure the watchdog to either be paused, or kept running, while the CPU is halted by the debugger
			Pause	0	Pause watchdog while the CPU is halted by the debugger
			Run	1	Keep the watchdog running while the CPU is halted by the debugger

6.22.4.12 RR[n] (n=0..7)

Address offset: 0x600 + (n × 0x4)

Reload request n

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	W	RR			Reload request register
			Reload	0x6E524635	Value to request a reload of the watchdog timer

6.22.5 Electrical specification

6.22.5.1 Watchdog Timer Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
t_{WDT}	Time out interval	31 µs		36 h	

7 LTE modem

7.1 Introduction

The long term evolution (LTE) modem consists of baseband processing and RF parts, which together implement a complete 3GPP LTE release 13 (Rel-13) Cat-M1 and Cat-NB1 and LTE release 14 (Rel-14) Cat-NB1 and Cat-NB2 capable product.

As illustrated in the image below, the following is a part of the LTE modem:

- RF transceiver
- Modem baseband (BB)
- Embedded flash/RAM
- Modem host processor and peripherals

The modem baseband and host processor provide functions for the LTE L1, L2 and L3 (layer 1, 2 and 3 respectively) as well as IP communication layers. Modem peripherals provide hardware services for modem operating system and for modem secure execution environment.

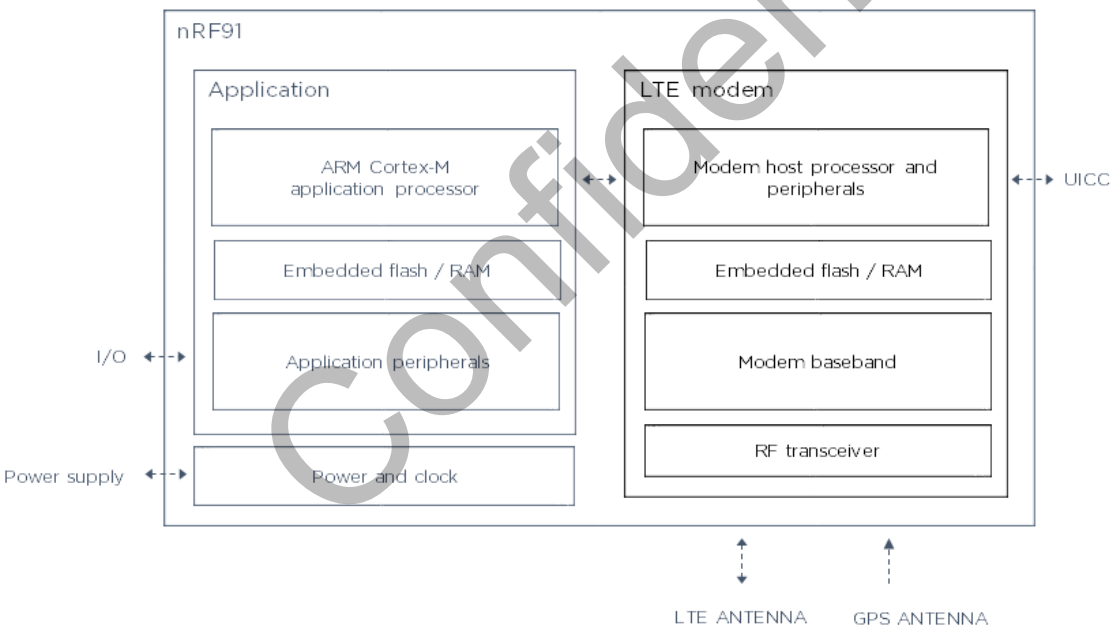

Figure 106: LTE modem within the nRF91

Application and modem domains are interacting through interprocessor communication (IPC) mechanism. LTE modem is accessible to user through the modem API.

The application processor is the master in the system and responsible for starting and stopping of the modem. LTE modem enables the clocks and power required for its own operation. Shared resources, such as e.g. clocks, are handled within the platform and require no user involvement. In cases where a hard fault is detected in the modem, the application domain will perform a hard reset for the modem.

Note: For details regarding the modem API, please refer to *nRF Connect SDK* document and *nRF91 AT Commands, Command Reference Guide* document.

Key features of the LTE modem are:

- Complete modem with baseband and RF transceiver
- 3GPP release 13 compliant LTE categories:
  - Cat-M1 (eMTC - enhanced machine type communication)
  - Cat-NB1 (NB-IoT - narrowband internet of things (IoT))
- 3GPP release 14 compliant LTE categories:
  - Cat-NB1 (NB-IoT)
  - Cat-NB2 (NB-IoT)
- Power saving modes
- Supporting LTE bands from 700 MHz to 2.2 GHz through a single typical 50 Ω antenna pin.
- RX sensitivity: -108 dBm for Cat-M1 and -114 dBm for Cat-NB1 and Cat-NB2
  - As defined in 3GPP conformance test specification TS 36.521-1
- 1.8 V MIPI RFFE (RF front-end) digital control interface and MAGPIO control interface for external RF applications.
- 1.8 V UICC (universal integrated circuit card) interface, based on ISO/IEC 7816-3 and compliant with:
  - ICC (ETSI TS 102 221)
  - eUICC (ETSI TS 103 383)

Note: nRF9160 is able to run different modem FW builds that define the final modem feature set in a specific nRF9160 based application. For details regarding the supported features configurable in LTE modem FW, please refer to *3GPP Features in nRF9160 Modem Firmware* document.

7.2 SIM card interface

LTE modem supports the UICC (universal integrated circuit card) interface.

Only the UICCs with the electrical interface specified in ISO/IEC 7816-3 are supported, meaning that the UICCs with IC-USB, CLF or MMC interfaces are not supported.

The supported UICC/eUICC interface is compliant with:

- ETSI TS 102 221: Smart Cards; UICC-Terminal interface; Physical and logical characteristics
- ETSI TS 103 383: Smart Cards; Embedded UICC; Requirements Specification

The physical interface towards the eUICC is the same as towards the removable UICC.

Only the class C (supply voltage 1.8 V nominal) operation is supported. Support for the legacy class B (supply voltage 3.0 V nominal) operation must be built with external components, including the external power supply and the level shifters towards the LTE modem UICC interface.

LTE modem controls the physical interfaces towards the UICC and implements the transport protocol over the four-pin ISO/IEC 7816-3 interface:

- VCC (power supply): LTE modem drives this
- CLK (clock signal): LTE modem drives this

- RST (reset signal): LTE modem drives this
- I/O (input/output serial data): Bi-directional

The interface and the connections between LTE modem, UICC connector, and the ESD device is shown in the figure below.

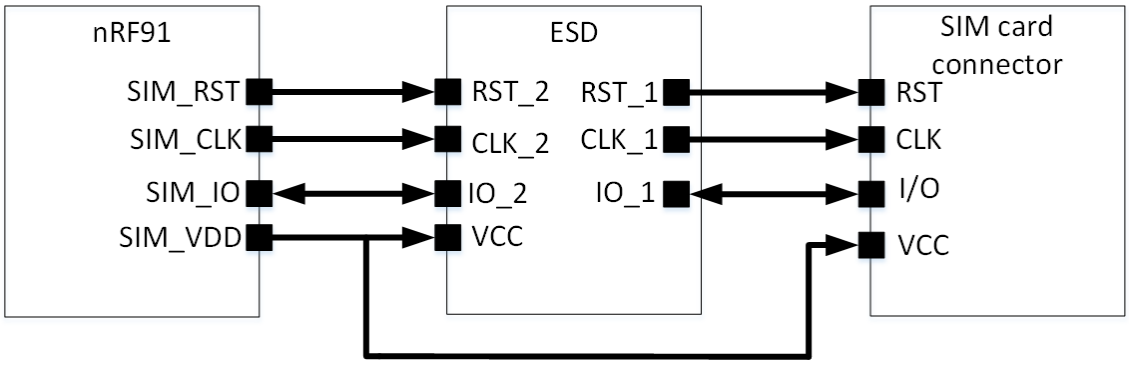

Figure 107: Connections between LTE modem, card connector, and the ESD device

Only standard transmission speeds are supported as specified in ETSI TS 102 221.

Important: LTE modem must be stopped through the modem API, before removing the UICC.

An ESD (electrostatic discharge) protection device compatible with UICC cards must be used between the removable card and the LTE modem, to protect LTE modem against a harmful electrostatic discharge from the card connector.

7.3 LTE modem coexistence interface

LTE modem uses a dedicated three-pin interface for RF interference avoidance towards a companion radio device e.g. an external Bluetooth® Low Energy device.

The inputs and outputs for this interface:

- COEX0: Input to the LTE modem from the external device. When active high, indicates that the external device transceiver is turned on.
- COEX1: Output from the LTE modem to the external device. Active high time mark pulse, which is synchronous to LTE system time.
- COEX2: Output from the LTE modem to the external device. When active high, indicates that the LTE modem transceiver is turned on.
  - COEX2 can also be treated as active low grant from LTE modem to the external device, indicating grant to transmit.
  - **Note:** COEX2 pin requires an external pull-down resistor in 100 kΩ size range to be used.

Simultaneous receiving by LTE modem and external device is always possible, and by so means no coexistence signaling needed when only receiving is done on the external device side.

7.4 LTE modem RF control external interface

LTE modem provides dedicated 1.8 V digital interfaces for controlling external RF applications, such as antenna tuner devices:

- MIPI RFFE interface pins: VIO, SCLK, SDATA.
- MAGPIO interface pins: MAGPIO0, MAGPIO1, MAGPIO2.

LTE modem drives these outputs timing accurately according to LTE protocol timing to set e.g. the correct antenna tuner settings per used frequency.

User needs to inform the LTE modem through the modem API about the particular RF application e.g. antenna tuner device configuration, so that LTE modem knows how to drive it.

Note: For details regarding the modem API and supported RF external control features, please refer to *nRF91 AT Commands, Command Reference Guide* document.

7.5 RF front-end interface

nRF9160 has a single-ended (SE) 50 Ω antenna interface to connect directly to antenna.

7.6 Electrical specification

7.6.1 Key RF parameters for Cat-M1

Note: The bands listed in this table define the certified bands.

Symbol	Description	Min.	Typ.	Max.	Units
Supported LTE	Supported LTE standards		LTE Rel-13 Cat-M1 HD-FDD		
Bands supported	Certified bands supported		USA and Canada: B4, B13. Europe: B3, B20.		
Transmission bandwidth	Maximum bandwidth		1.4		MHz

7.6.2 Key RF parameters for Cat-NB1 and Cat-NB2

Note: The bands listed in this table define the certified bands.

Symbol	Description	Min.	Typ.	Max.	Units
Supported LTE	Supported LTE standards		LTE Rel-13 Cat-NB1 HD-FDD, LTE Rel-14 Cat-NB1 and Cat-NB2 HD-FDD		
Bands supported	Certified bands supported		For Cat-NB1, Europe: B3, B20		
Transmission bandwidth	Maximum bandwidth		200		kHz

7.6.3 Receiver parameters for Cat-M1

Symbol	Description	Min.	Typ.	Max.	Units
$Freq_{range\_ANT\_RX}$	RX operation frequency range at ANT (pin 61)	729		2200	MHz
Z_{in}	Input impedance, single-ended		50		Ω
Sensitivity, low band	LTE 1.4 MHz without coverage extension	-103	-108		dBm
Sensitivity, mid band	LTE 1.4 MHz without coverage extension	-103	-107		dBm

7.6.4 Receiver parameters for Cat-NB1 and Cat-NB2

Symbol	Description	Min.	Typ.	Max.	Units
$Freq_{range\_ANT\_RX}$	RX operation frequency range at ANT (pin 61)	729		2200	MHz
Z_{in}	Input impedance, single-ended		50		Ω
Sensitivity, low band	NB 200 kHz without coverage extension	-108	-114		dBm
Sensitivity, mid band	NB 200 kHz without coverage extension	-108	-113		dBm

7.6.5 Transmitter parameters for Cat-M1

Symbol	Description	Min.	Typ.	Max.	Units
$Freq_{range\_ANT\_TX}$	TX operation frequency range at ANT (pin 61)	699		1910	MHz
Z_{out}	Output impedance, single-ended		50		Ω
Maximum output power	Maximum output power, 3GPP specification		23		dBm
Minimum output power	Minimum output power, 3GPP specification		-40		dBm
Pout maximum accuracy	Pout maximum accuracy, internal specification		+-2		dB

7.6.6 Transmitter parameters for Cat-NB1 and Cat-NB2

Symbol	Description	Min.	Typ.	Max.	Units
$Freq_{range\_ANT\_TX}$	TX operation frequency range at ANT (pin 61)	699		1910	MHz
Z_{out}	Output impedance, single-ended		50		Ω
Maximum output power	Maximum output power, 3GPP specification		23		dBm
Minimum output power	Minimum output power, 3GPP specification		-40		dBm
Pout maximum accuracy	>Pout maximum accuracy, Internal specification		+-2		dB

8 GPS receiver

The GPS receiver supports GPS L1C/A reception. Operation is time multiplexed with LTE modem, and it is possible to obtain the GPS position while LTE is in DRX or PSM mode.

GPS receiver is visible to user only through GPS API.

Key features of the GPS receiver are:

- GPS L1C/A is supported
- Modes of operation:
  - Single shot (cold start mode by default)
  - Position fix per fixed interval, e.g. 2 minutes (starts with cold start, sequential fixes with hot start)
  - Duty cycled mode, tracking duty cycling e.g. fix per every second, time multiplexed between GPS and LTE M1/NB1
- Power saving modes
- Antenna interface options:
  - Dedicated GPS antenna or shared antenna with LTE M1/NB1
  - With or without external low-noise amplifier (LNA)
- Performance:
  - Acquisition sensitivity: -144 dBm for cold start, -147 dBm for hot start
  - Acquisition time: 30 seconds for cold start, 5 seconds for hot start
  - Tracking sensitivity: -149 dBm (open sky received power ≥ -130 dBm)
  - Accuracy: 5 m

9 Debug and trace

9.1 Debug

The debug system offers a flexible and powerful mechanism for non-intrusive debugging.

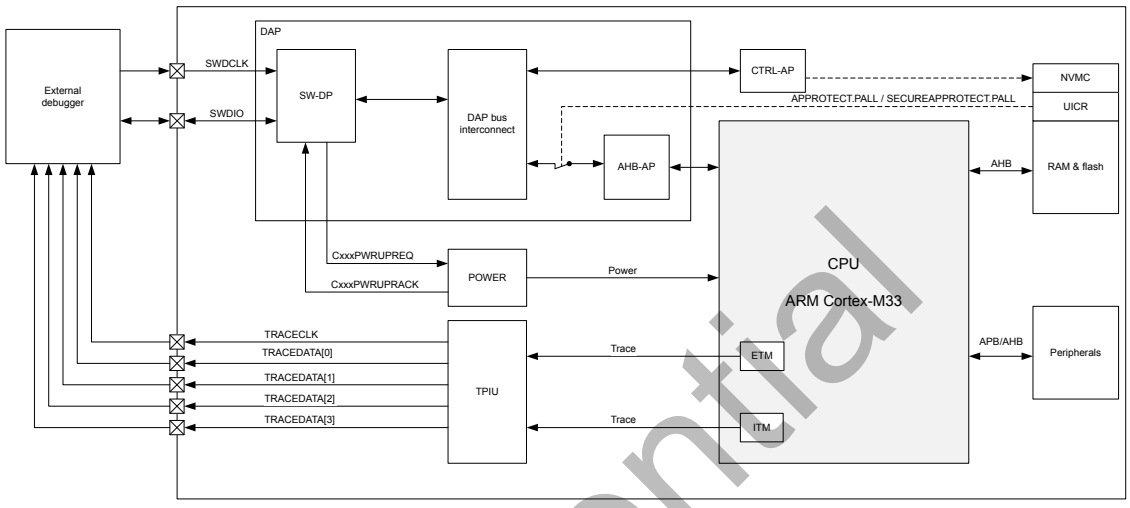

Figure 108: Debug and trace overview

The main features of the debug system are:

- Two-pin serial wire debug (SWD) interface
- Breakpoint unit (BPU) supports eight hardware breakpoint comparators
- Data watchpoint and trace (DWT) unit supports eight watchpoint comparators

Note: When a system contains multiple CPU domains, it is important to notice that if one domain (subsystem A) has master rights on another domain (subsystem B), the master subsystem can have access to data from the slave subsytem. In this example, even if subsystem B is locked by APPROTECT or ERASEPROTECT, subsystem A can access some data for subsystem B. Consequently, even if the security permissions are managed per subsystem, it is mandatory to have a global approach to the protection. Protecting a slave subsystem does not guarantee system security if the master subsystem is not protected.

9.1.1 Special consideration regarding debugger access

A debugger can be restricted to debug non-secure code only, and access non-secure memory regions and peripherals using register SECUREAPPROTECT on page 44. Register APPROTECT on page 43 will block all debugger access.

Debugger accesses are controlled as described in table below.

Debugging capability	UICR.APPROTECT.PALL	UICR.SECUREAPPROTECT.PALL
Secure and non-secure code	Unprotected	Unprotected
Non-secure code only	Unprotected	Protected
No debugging possible	Protected	-

Table 95: Debugger access control

If a RAM or flash region has its permission set to allow code execution, the content of this region will be visible to the debugger even if the read permission is not set. This allows a debugger to display the content of the code being executed.

9.1.2 DAP - Debug access port

An external debugger can access the device via the debug access port (DAP).

The DAP implements a standard ARM® CoreSight™ serial wire debug port (SW-DP). The SW-DP implements the serial wire debug (SWD) protocol that is a two-pin serial interface, see SWDCLK and SWDIO in Debug and trace overview on page 367.

In addition to the default access port in the application CPU (AHB-AP), the DAP includes a custom control access port (CTRL-AP). The CTRL-AP is described in more detail in CTRL-AP - Control access port on page 370.

Note:

- The SWDIO line has an internal pull-up resistor.
- The SWDCLK line has an internal pull-down resistor.

There are several access ports that connect to different parts of the system. An overview is given in table Access port overview on page 368. The AHB-AP and APB-AP are standard ARM components, and documented in *ARM CoreSight SoC-400 Technical Reference Manual, revision r3p2*. The control access port (CTRL-AP) is proprietary, and described in more detail in CTRL-AP - Control access port on page 370.

AP ID	Type	Description
0	AHB-AP	Application subsystem access port
4	CTRL-AP	Application subsystem control access port

Table 96: Access port overview

9.1.3 Debug interface mode

Before the external debugger can access the CPU's access port (AHB-AP) or the control access port (CTRL-AP), the debugger must first request the device to power up via CxxxPWRUPREQ in the SWJ-DP.

As long as the debugger is requesting power via CxxxPWRUPREQ, the device will be in debug interface mode. Otherwise, the device is in normal mode. When a debug session is over, the external debugger must make sure to put the device back into normal mode and then a pin reset should be performed. The reason is that the overall power consumption is higher in debug interface mode compared to normal mode.

Some peripherals behave differently in debug interface mode compared to normal mode. The differences are described in more detail in the chapters of the peripherals that are affected.

For details on how to use the debug capabilities, please read the debug documentation of your IDE.

If the device is in System OFF when power is requested via CxxxPWRUPREQ, the system will wake up and the DIF flag in RESETREAS on page 65 will be set.

9.1.4 Real-time debug

The device supports real-time debugging, which allows interrupts to execute to completion in real time when breakpoints are set in thread mode or lower priority interrupts.

Real-time debugging thus enables the developer to set a breakpoint and single-step through their code without a failure of the real-time event-driven threads running at higher priority. For example, this enables the device to continue to service the high-priority interrupts of an external controller or sensor without failure or loss of state synchronization while the developer steps through code in a low-priority thread.

9.1.5 Registers

Register	Offset	Security	Description
TARGETID	0x042		The TARGETID register provides information about the target when the host is connected to a single device. The TARGETID register is Accessed by a read of DP register 0x4 when the DPBANKSEL bit in the SELECT Register is set to 0x2.

Table 97: Register overview

9.1.5.1 TARGETID

Address offset: 0x042

The TARGETID register provides information about the target when the host is connected to a single device.

The TARGETID register is Accessed by a read of DP register 0x4 when the DPBANKSEL bit in the SELECT Register is set to 0x2.

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				D	D	D	D	C	C	C	C	C	C	C	C	C	C	C	C					B	B	B	B	B	B	B	B	B	B	B	A
Reset 0x10090289				**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**1**
ID	Acce	Field	Value ID	Value	Description																														
A	R	UNUSED			Reserved, read-as-one.																														
B	R	TDESIGNER			An 11-bit code JEDEC JEP106 continuation code and identity code. The ID identifies the designer of the part.																														
			NordicSemi	0x144	Nordic Semiconductor ASA																														
C	R	TPARTNO			Part number.																														
			nRF91	9	nRF91 Series																														
D	R	TREVISION			Target revision.																														
			<keyword keyref="devicename" />	1	nRF9160																														

9.1.6 Electrical specification

9.1.6.1 Trace port

Symbol	Description	Min.	Typ.	Max.	Units
T_{cyc}	Clock period, as defined by ARM (See ARM Infocenter, Embedded Trace Macrocell Architecture Specification, Trace Port Physical Interface, Timing specifications)	..	..	..	ns

9.1.7 Trace

The device supports ETM and ITM trace.

Trace data from the ETM and the ITM is sent to an external debugger via a 4-bit wide parallel trace port (TPIU), see TRACEDATA[0] through TRACEDATA[3], and TRACECLK in Debug and trace overview on page 367.

For details on how to use the trace capabilities, please read the debug documentation of your IDE.

TPIU's trace pins are multiplexed with GPIOs, see Pin assignments on page 381 for more information.

Trace speed is configured in the TRACEPORTSPEED on page 380 register. The speed of the trace pins depends on the DRIVE setting of the GPIOs that the trace pins are multiplexed with. Only S0S1 and H0H1 drives are suitable for debugging. S0S1 is the default DRIVE at reset. If parallel or serial trace port signals are not fast enough in the debugging conditions, all GPIOs in use for tracing should be set to high drive (H0H1). The user shall make sure that DRIVE setting for these GPIOs is not overwritten by software during the debugging session.

9.1.8 CTRL-AP - Control access port

The control access port (CTRL-AP) is a custom access port that enables control of the device when other access ports in the debug access port (DAP) are disabled by the access port protection.

For overview of other access ports in DAP, see DAP - Debug access port on page 368.

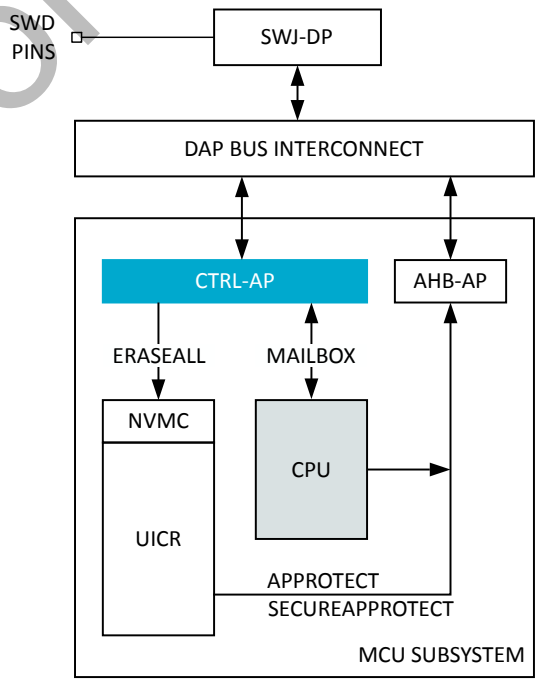

Figure 109: Control access port details

Access port protection (APPROTECT) blocks the debugger access to the AHB-AP, and prevents read and write access to all CPU registers and memory-mapped addresses. It is possible to enable access port protection for both secure and non-secure mode, using registers UICR.SECUREAPPROTECT and UICR.APPROTECT respectively. The debugger can use register APPROTECT.STATUS on page 374 to read the status of secure and non-secure access port protection.

Control access port has the following features:

- Soft reset
- Erase all
- Mailbox interface
- Debug of protected devices

9.1.8.1 Reset request

The debugger can request the device to perform a soft reset.

Register RESET on page 373 is used to request the soft reset. Once the soft reset is performed, the reset reason is accessible to on-chip firmware through register . For more information about the soft reset, see Reset on page 55.

9.1.8.2 Erase all

Erase all function gives debugger the possibility of triggering an erase of flash, user information configuration registers (UICR), RAM, including all peripheral settings, as well as removing the access port protection.

To trigger an erase all function, the debugger can write the register ERASEALL on page 373. Register ERASEALLSTATUS on page 373 will read as busy for the duration of the operation. After the next reset, the access port protection is removed.

For slave MCU's, the ERASEALL command will also affect the application MCU. The ERASEALL command is performed on the application MCU first, independently of how the application is protected, and then on the slave MCU.

Erase all protection

It is possible to prevent debugger from performing an erase all operation by writing to register ERASEPROTECT on page 44. Once the register is configured and the device reset, the control access port ERASEALL operation is disabled, and all flash write and erase operations are restricted to firmware. In addition, it is still possible to write/erase from debugger as long as APPROTECT on page 43 is not set.

> **Note:** Setting ERASEPROTECT on page 44 has no effect on debugger access, only on erase all operation.

Register ERASEPROTECT.STATUS on page 374 holds the status for erase protection.

9.1.8.3 Mailbox interface

CTRL-AP implements a mailbox interface which enables the CPU to communicate with a debugger over the SWD interface.

The mailbox interface consists of a transmit register MAILBOX.TXDATA on page 374 with its corresponding status register MAILBOX.TXSTATUS on page 375, and a receive register MAILBOX.RXDATA on page 375 with its corresponding status register MAILBOX.RXSTATUS on page 375. Status bits in registers TXSTATUS/RXSTATUS will be set and cleared automatically when registers TXDATA/RXDATA are written to and read from, independently of the direction.

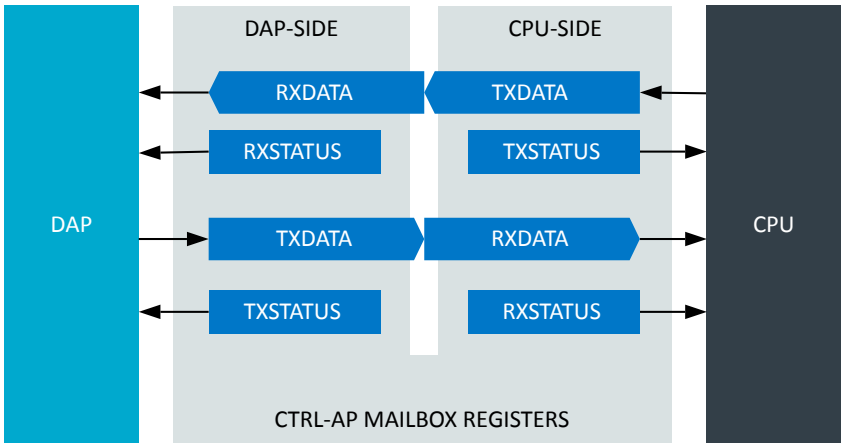

Figure 110: Mailbox register interface

Mailbox transfer sequence

1. Sender writes TXDATA
2. Hardware sets sender's TXSTATUS to DataPending
3. Hardware sets receiver's RXSTATUS to DataPending
4. Receiver reads RXDATA
5. Hardware sets receiver's RXSTATUS to NoDataPending
6. Hardware sets sender's TXSTATUS to NoDataPending

9.1.8.4 Unlocking of access port

The access port protection mechanisms can be temporarily bypassed to erase or debug the device.

Note: The mailbox feature of the CTRL-AP can be used by firmware to authenticate the debugger before allowing it to use the access port.

Disabling the erase all protection

To bypass ERASEPROTECT setting, making it possible for the access port to erase all memories, both the debugger and firmware must set the ERASEALL field in their respective ERASEPROTECTDISABLE registers. As soon as both registers have been written, the device is automatically erased using erase all function as described in Erase all on page 371, and then the access port is made available.

Note: To prevent misuse, the write-once register ERASEPROTECT.DISABLE on page 377 should be set to Default as early in the start-up process as possible. Once written, it will not be possible to remove the erase protection until next reset.

9.1.8.5 Registers

Register	Offset	Security	Description
RESET	0x000		Soft reset request.
ERASEALL	0x004		Perform a secure erase of the device. The device will be returned to factory default settings upon next reset.
ERASEALLSTATUS	0x008		Status register for the ERASEALL operation
APPROTECT.STATUS	0x00C		Status register for access port protection
ERASEPROTECT.STATUS	0x018		Status register for UICR ERASEPROTECT configuration
ERASEPROTECT.DISABLE	0x01C		Unlock ERASEPROTECT and perform ERASEALL
MAILBOX.TXDATA	0x020		Data sent from the debugger to the CPU

Register	Offset	Security	Description
MAILBOX.TXSTATUS	0x024		Status to indicate if data sent from the debugger to the CPU has been read
MAILBOX.RXDATA	0x028		Data sent from the CPU to the debugger
MAILBOX.RXSTATUS	0x02C		Status to indicate if data sent from the CPU to the debugger has been read
IDR	0x0FC		CTRL-AP Identification Register, IDR

Table 98: Register overview

9.1.8.5.1 RESET

Address offset: 0x000

Soft reset request.

This register is automatically deactivated by writing Erase to ERASEALL, it is then kept inactive until a reset source affecting the debug system is asserted. See Reset behavior on page 56.

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	RESET			Soft reset request and status
			NoReset	0	Write to release reset Reading '0' means reset is not active
			Reset	1	Write to hold reset Reading '1' means reset is active

9.1.8.5.2 ERASEALL

Address offset: 0x004

Perform a secure erase of the device. The device will be returned to factory default settings upon next reset.

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	W	ERASEALL			Erase flash, SRAM and UICR in sequence
			NoOperation	0	No operation
			Erase	1	Erase flash, SRAM and UICR in sequence

9.1.8.5.3 ERASEALLSTATUS

Address offset: 0x008

Status register for the ERASEALL operation

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	R	ERASEALLSTATUS			Status register for the ERASEALL operation
			Ready	0	ERASEALL is ready
			Busy	1	ERASEALL is busy (on-going)

9.1.8.5.4 APPROTECT.STATUS

Address offset: 0x00C

Status register for access port protection

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	R	APPROTECT			Status bit for access port protection
			Enabled	0	APPROTECT is enabled
			Disabled	1	APPROTECT is disabled
B	R	SECUREAPPROTECT			Status bit for secure access port protection
			Enabled	0	SECUREAPPROTECT is enabled
			Disabled	1	SECUREAPPROTECT is disabled

9.1.8.5.5 ERASEPROTECT.STATUS

Address offset: 0x018

Status register for UICR ERASEPROTECT configuration

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	R	PALL			ERASEALL status
			Enabled	0	ERASEALL protection is enabled
			Disabled	1	ERASELL protection is not enabled and device can be erased

9.1.8.5.6 ERASEPROTECT.DISABLE

Address offset: 0x01C

Unlock ERASEPROTECT and perform ERASEALL

This register can only be written once per reset

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW1	KEY			Initiate secure erase even though ERASEPROTECT is enabled if KEY fields match

9.1.8.5.7 MAILBOX.TXDATA

Address offset: 0x020

Data sent from the debugger to the CPU

Writing to this register will automatically set field DataPending in register TXSTATUS

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	Data			Data sent from debugger

9.1.8.5.8 MAILBOX.TXSTATUS

Address offset: 0x024

Status to indicate if data sent from the debugger to the CPU has been read

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	R	Status			Status of register DATA
			NoDataPending	0	No data pending in register TXDATA
			DataPending	1	Data pending in register TXDATA

9.1.8.5.9 MAILBOX.RXDATA

Address offset: 0x028

Data sent from the CPU to the debugger

Reading from this register will automatically set field NoDataPending in register RXSTATUS

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	R	Data			Data sent from CPU

9.1.8.5.10 MAILBOX.RXSTATUS

Address offset: 0x02C

Status to indicate if data sent from the CPU to the debugger has been read

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	R	Status			Status of register DATA
			NoDataPending	0	No data pending in register RXDATA
			DataPending	1	Data pending in register RXDATA

9.1.8.5.11 IDR

Address offset: 0x0FC

CTRL-AP Identification Register, IDR

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E E E E D D D D C C C C C C C B B B B A A A A A A A A
Reset 0x12880000				**0 0 0 1 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	R	APID			AP Identification
B	R	CLASS			Access Port (AP) class
			NotDefined	0x0	No defined class
			MEMAP	0x8	Memory Access Port
C	R	JEP106ID			JEDEC JEP106 identity code
D	R	JEP106CONT			JEDEC JEP106 continuation code
E	R	REVISION			Revision

9.1.8.6 Registers

Base address	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50006000	CTRLAPPERI	CTRL_AP_PERI	S	NA	CTRL-AP-PERI	

Table 99: Instances

Register	Offset	Security	Description
MAILBOX.RXDATA	0x400		Data sent from the debugger to the CPU
MAILBOX.RXSTATUS	0x404		Status to indicate if data sent from the debugger to the CPU has been read
MAILBOX.TXDATA	0x480		Data sent from the CPU to the debugger
MAILBOX.TXSTATUS	0x484		Status to indicate if data sent from the CPU to the debugger status has been read
ERASEPROTECT.LOCK	0x500		Lock ERASEALL mechanism
ERASEPROTECT.DISABLE	0x504		Unlock ERASEPROTECT and perform ERASEALL

Table 100: Register overview

9.1.8.6.1 MAILBOX.RXDATA

Address offset: 0x400

Data sent from the debugger to the CPU

Reading from this register will automatically set field NoDataPending in register RXSTATUS

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	R	RXDATA			Data received from debugger

9.1.8.6.2 MAILBOX.RXSTATUS

Address offset: 0x404

Status to indicate if data sent from the debugger to the CPU has been read

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	R	RXSTATUS			Status of data in register RXDATA
			NoDataPending	0	No data pending in register RXDATA
			DataPending	1	Data pending in register RXDATA

9.1.8.6.3 MAILBOX.TXDATA

Address offset: 0x480

Data sent from the CPU to the debugger

Writing to this register will automatically set field DataPending in register TXSTATUS

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	TXDATA			Data sent to debugger

9.1.8.6.4 MAILBOX.TXSTATUS

Address offset: 0x484

Status to indicate if data sent from the CPU to the debugger status has been read

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	R	TXSTATUS			Status of data in register TXDATA
			NoDataPending	0	No data pending in register TXDATA
			DataPending	1	Data pending in register TXDATA

9.1.8.6.5 ERASEPROTECT.LOCK

Address offset: 0x500

Lock ERASEALL mechanism

This register can only be written once per reset

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW1	ERASEPROTECTLOCK			Enable or disable the ERASEALL mechanism
			Unlocked	0	ERASEALL can be issued
			Locked	1	ERASEALL is locked

9.1.8.6.6 ERASEPROTECT.DISABLE

Address offset: 0x504

Unlock ERASEPROTECT and perform ERASEALL

This register can only be written once per reset

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	KEY			Initiate secure erase even though ERASEPROTECT is enabled if KEY fields match

9.2 TAD - Trace and debug control

Configuration interface for trace and debug

9.2.1 Registers

Base address	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0xE0080000	TAD	TAD	S	NA	Trace and debug control	

Table 101: Instances

Register	Offset	Security	Description
ENABLE	0x500		Enable debug domain and aquire selected GPIOs
PSEL.TRACECLK	0x504		Pin number configuration for TRACECLK
PSEL.TRACEDATA0	0x508		Pin number configuration for TRACEDATA[0]
PSEL.TRACEDATA1	0x50C		Pin number configuration for TRACEDATA[1]
PSEL.TRACEDATA2	0x510		Pin number configuration for TRACEDATA[2]
PSEL.TRACEDATA3	0x514		Pin number configuration for TRACEDATA[3]
TRACEPORTSPEED	0x518		Clocking options for the Trace Port debug interface

Table 102: Register overview

9.2.1.1 ENABLE

Address offset: 0x500

Enable debug domain and aquire selected GPIOs

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	ENABLE			
			DISABLED	0	Disable debug domain and release selected GPIOs
			ENABLED	1	Enable debug domain and aquire selected GPIOs

9.2.1.2 PSEL.TRACECLK

Address offset: 0x504

Pin number configuration for TRACECLK

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			B A A A A A
Reset 0xFFFFFFFF			**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**

ID	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

9.2.1.3 PSEL.TRACEDATA0

Address offset: 0x508

Pin number configuration for TRACEDATA[0]

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			B A A A A A
Reset 0xFFFFFFFF			**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**

ID	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

9.2.1.4 PSEL.TRACEDATA1

Address offset: 0x50C

Pin number configuration for TRACEDATA[1]

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			B A A A A A
Reset 0xFFFFFFFF			**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**

ID	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

9.2.1.5 PSEL.TRACEDATA2

Address offset: 0x510

Pin number configuration for TRACEDATA[2]

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			B A A A A A
Reset 0xFFFFFFFF			**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**

ID	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

9.2.1.6 PSEL.TRACEDATA3

Address offset: 0x514

Pin number configuration for TRACEDATA[3]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

9.2.1.7 TRACEPORTSPEED

Address offset: 0x518

Clocking options for the Trace Port debug interface

This register is a retained register. Reset behavior is the same as debug components.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	TRACEPORTSPEED			Speed of Trace Port clock. Note that the TRACECLK pin will output this clock divided by two.
			32MHz	0	32 MHz Trace Port clock (TRACECLK = 16 MHz)
			16MHz	1	16 MHz Trace Port clock (TRACECLK = 8 MHz)
			8MHz	2	8 MHz Trace Port clock (TRACECLK = 4 MHz)
			4MHz	3	4 MHz Trace Port clock (TRACECLK = 2 MHz)

10 Hardware and layout

10.1 Pin assignments

This section describes the pin assignment and the pin functions.

This device provides flexibility when it comes to routing and configuration of the GPIO pins. However, some pins have recommendations for how the pin should be configured or what it should be used for. See LGA pin assignments on page 381 for more information about this.

10.1.1 Pin assignments

The pin assignment table and figure describe the assignments for this variant of the chip.

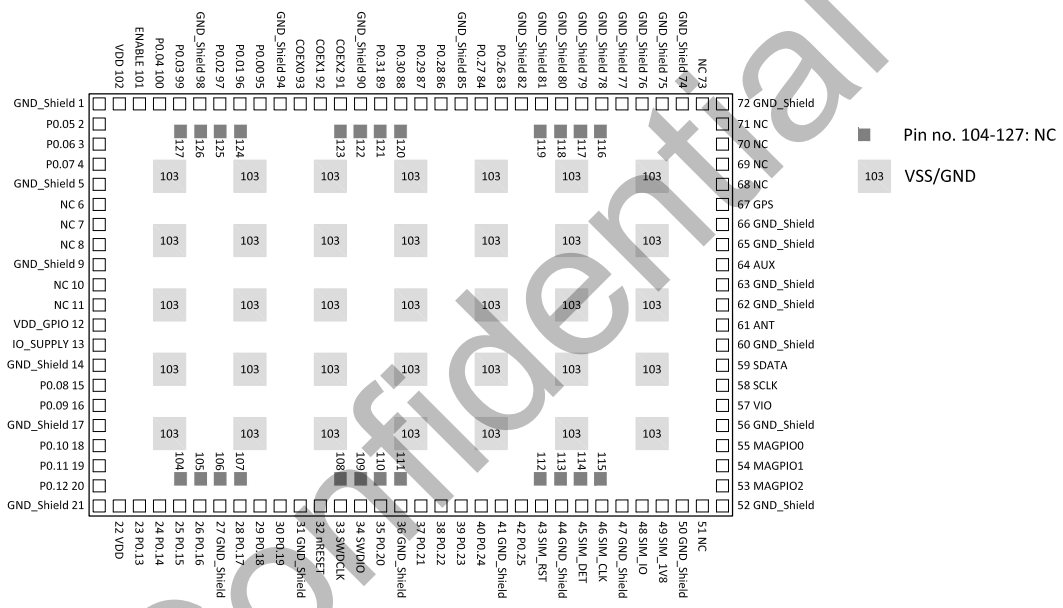

Figure 111: LGA pin assignments, top view

Pin no	Pin name	Function	Description
1	GND_Shield	Power	Ground
2	P0.05	Digital I/O (SoC)	General purpose I/O
3	P0.06	Digital I/O (SoC)	General purpose I/O
4	P0.07	Digital I/O (SoC)	General purpose I/O
5	GND_Shield	Power	Ground
6	NC		Not connected/reserved for Nordic use
7	NC		Not connected/reserved for Nordic use
8	NC		Not connected/reserved for Nordic use
9	GND_Shield	Power	Ground
10	NC		Not connected/reserved for Nordic use
11	NC		Not connected/reserved for Nordic use
12	VDD_GPIO	Power	GPIO power supply input and logic level
13	IO_SUPPLY	Power	Reserved for Nordic use
14	GND_Shield	Power	Ground
15	P0.08	Digital I/O (SoC)	General purpose I/O
16	P0.09	Digital I/O (SoC)	General purpose I/O

Pin no	Pin name	Function	Description
17	GND_Shield	Power	Ground
18	P0.10	Digital I/O (SoC)	General purpose I/O
19	P0.11	Digital I/O (SoC)	General purpose I/O
20	P0.12	Digital I/O (SoC)	General purpose I/O
21	GND_Shield	Power	Ground
22	VDD	Power	Supply voltage input, 3.1-5.5 V
23	P0.13	Digital I/O (SoC)	General purpose I/O.
	AIN0	Analog input	Analog input.
24	P0.14	Digital I/O (SoC)	General purpose I/O.
	AIN1	Analog input	Analog input.
25	P0.15	Digital I/O (SoC)	General purpose I/O.
	AIN2	Analog input	Analog input.
26	P0.16	Digital I/O (SoC)	General purpose I/O.
	AIN3	Analog input	Analog input.
27	GND_Shield	Power	Ground
28	P0.17	Digital I/O (SoC)	General purpose I/O.
	AIN4	Analog input	Analog input.
29	P0.18	Digital I/O (SoC)	General purpose I/O.
	AIN5	Analog input	Analog input.
30	P0.19	Digital I/O (SoC)	General purpose I/O.
	AIN6	Analog input	Analog input.
31	GND_Shield	Power	Ground
32	nRESET	Digital I/O (SoC)	System reset
33	SWDCLK	Digital input	Serial wire debug clock input for debug and programming
34	SWDIO	Digital I/O	Serial wire debug I/O for debug and programming
35	P0.20	Digital I/O (SoC)	General purpose I/O.
	AIN7	Analog input	Analog input.
36	GND_Shield	Power	Ground
37	P0.21	Digital I/O (SoC)	General purpose I/O.
	TRACECLK	Trace clock	Trace buffer clock (optional).
38	P0.22	Digital I/O (SoC)	General purpose I/O.
	TRACEDATA0	Trace data	Trace buffer TRACEDATA[0] (optional).
39	P0.23	Digital I/O (SoC)	General purpose I/O.
	TRACEDATA1	Trace data	Trace buffer TRACEDATA[1] (optional).
40	P0.24	Digital I/O (SoC)	General purpose I/O.
	TRACEDATA2	Trace data	Trace buffer TRACEDATA[2] (optional).
41	GND_Shield	Power	Ground
42	P0.25	Digital I/O (SoC)	General purpose I/O.
	TRACEDATA3	Trace data	Trace buffer TRACEDATA[3] (optional).
43	SIM_RST	Digital I/O (SoC)	SIM reset
44	GND_Shield	Power	Ground
45	SIM_DET	Digital I/O (SoC)	SIM detect
46	SIM_CLK	Digital I/O (SoC)	SIM clock
47	GND_Shield	Power	Ground
48	SIM_IO	Digital I/O (SoC)	SIM data
49	SIM_1V8	Power	SIM 1.8 V power supply output
50	GND_Shield	Power	Ground
51	NC		Not connected/reserved for Nordic use
52	GND_Shield	Power	Ground

Pin no	Pin name	Function	Description
53	MAGPIO2	Digital I/O (SoC)	Reserved for Nordic use
54	MAGPIO1	Digital I/O (SoC)	Reserved for Nordic use
55	MAGPIO0	Digital I/O (SoC)	Reserved for Nordic use
56	GND_Shield	Power	Ground
57	VIO	Power	Reserved for Nordic use
58	SCLK	Digital I/O (SoC)	Reserved for Nordic use
59	SDATA	Digital I/O (SoC)	Reserved for Nordic use
60	GND_Shield	Power	Ground
61	ANT	RF	Single-ended radio antenna connection
62	GND_Shield	Power	Ground
63	GND_Shield	Power	Ground
64	AUX	RF	ANT output port
65	GND_Shield	Power	Ground
66	GND_Shield	Power	Ground
67	GPS	RF	GPS receiver input
68	NC		Not connected/reserved for Nordic use
69	NC		Not connected/reserved for Nordic use
70	NC		Not connected/reserved for Nordic use
71	NC		Not connected/reserved for Nordic use
72	GND_Shield	Power	Ground
73	NC		Not connected/reserved for Nordic use
74	GND_Shield	Power	Ground
75	GND_Shield	Power	Ground
76	GND_Shield	Power	Ground
77	GND_Shield	Power	Ground
78	GND_Shield	Power	Ground
79	GND_Shield	Power	Ground
80	GND_Shield	Power	Ground
81	GND_Shield	Power	Ground
82	GND_Shield	Power	Ground
83	P0.26	Digital I/O (SoC)	General purpose I/O
84	P0.27	Digital I/O (SoC)	General purpose I/O
85	GND_Shield	Power	Ground
86	P0.28	Digital I/O (SoC)	General purpose I/O
87	P0.29	Digital I/O (SoC)	General purpose I/O
88	P0.30	Digital I/O (SoC)	General purpose I/O
89	P0.31	Digital I/O (SoC)	General purpose I/O
90	GND_Shield	Power	Ground
91	COEX2	Digital I/O (SoC)	Coexistence interface
92	COEX1	Digital I/O (SoC)	Coexistence interface
93	COEX0	Digital I/O (SoC)	Coexistence interface
94	GND_Shield	Power	Ground
95	P0.00	Digital I/O (SoC)	General purpose I/O
96	P0.01	Digital I/O (SoC)	General purpose I/O
97	P0.02	Digital I/O (SoC)	General purpose I/O
98	GND_Shield	Power	Ground
99	P0.03	Digital I/O (SoC)	General purpose I/O
100	P0.04	Digital I/O (SoC)	General purpose I/O
101	ENABLE		Enable
102	VDD	Power	Supply voltage 3.1-5.5 V
103	VSS	Power	Ground
104-127	NC		Not connected/reserved for Nordic use

Table 103: LGA pin assignments

10.2 Mechanical specifications

The mechanical specifications for the packages show the dimensions in millimeters.

10.2.1 16.00 x 10.50 mm package

Dimensions in millimeters for the nRF9160 LGA 16.00 x 10.50 mm package.

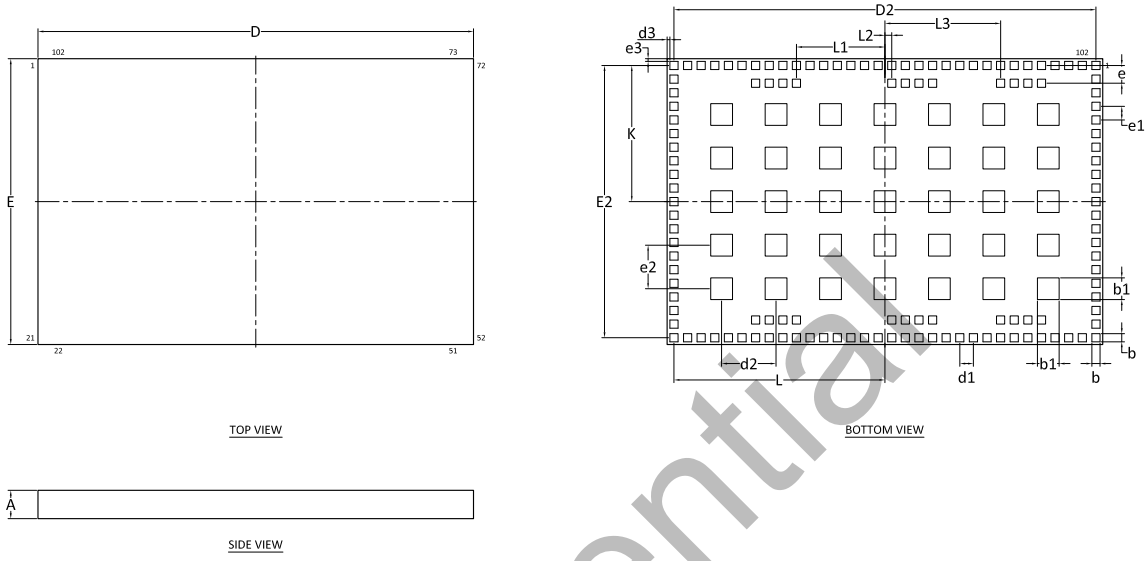

Figure 112: LGA 16.00 x 10.50 mm package

	A	b	b1	D	E	e	d1	e1	D2	E2	d2	e2	d3	e3	K	L	L1	L2	L3
Min.	0.98			15.90	10.40														
Nom.	1.04	0.30	0.80	16.00	10.50	0.65	0.50	0.50	15.50	10.00	2.00	1.60	0.10	0.10	5.00	7.75	3.25	0.25	4.25
Max.	1.10			16.10	10.60														

Table 104: LGA dimensions in millimeters

10.3 Reference circuitry

To ensure good RF performance when designing PCBs, it is highly recommended to use the PCB layouts and component values provided by Nordic Semiconductor.

Documentation for the different package reference circuits, including Altium Designer files, PCB layout files, and PCB production files can be downloaded from www.nordicsemi.com.

10.3.1 LGA schematic

The bill of material (BOM) is TBD.

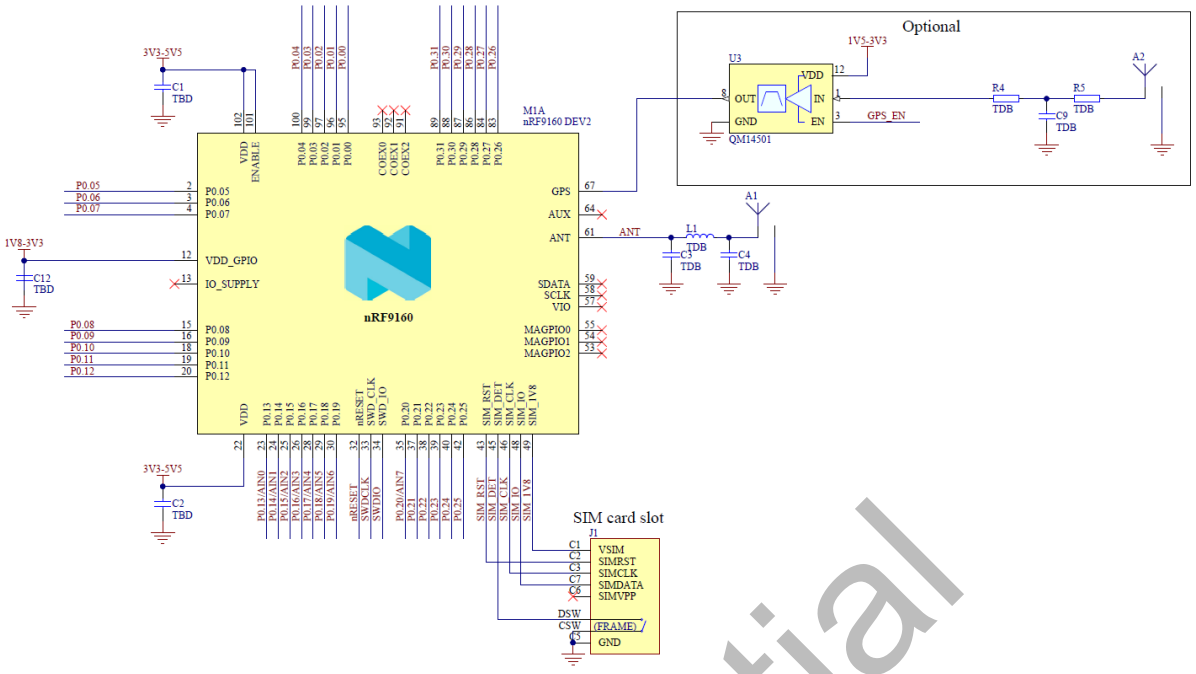

Figure 113: LGA schematic

11 Recommended operating conditions

The operating conditions are the physical parameters that the chip can operate within.

Symbol	Parameter	Notes	Min.	Nom.	Max.	Units
VDD	Battery input voltage	Including voltage drop, ripple and spikes. RF 3GPP compliancy requires 3.3 V.	3.0	3.8	5.5	V
VDD_GPIO	GPIO input voltage		1.7		3.6	V
$GPIO_H$	GPIO high level voltage				VDD_GPIO	V
$MAGPIO_H$	MAGPIO high level voltage			1.8	1.8	V
$t_{R\_VDD\_GPIO}$	VDD_GPIO rise time (0 V to 1.7 V)	IO_SUPPLY should be ramped up before or in parallel with VDD_GPIO			60	ms
TA	Operating temperature		-40	25	85	°C

Table 105: Recommended operating conditions

Note: There can be excessive leakage at VDD and/or VDD_GPIO if any of these supply voltages is outside its range given in Recommended operating conditions on page 386.

11.1 VDD_GPIO considerations

VDD_GPIO is the supply to the general purpose I/O.

The following restrictions should be taken into considerations:

- VDD_GPIO should be applied after VDD has been supplied
- VDD_GPIO should be removed before removing VDD
- If VDD is supplied and VDD_GPIO is grounded, an extra current consumption can be generated on VDD

12 Absolute maximum ratings

Maximum ratings are the extreme limits to which the chip can be exposed for a limited amount of time without permanently damaging it. Exposure to absolute maximum ratings for prolonged periods of time may affect the reliability of the device.

	Note	Min.	Max.	Unit
Supply voltages				
VDD		-0.3	5.5	V
VDD_GPIO		-0.3	3.9	V
VSS			0	V
I/O pin voltage				
$V_{I/O}$, VDD_GPIO ≤ 3.6 V		-0.3	VDD_GPIO + 0.3	V
$V_{I/O}$, VDD_GPIO > 3.6 V		-0.3	3.9	V
Radio				
RF input level			10	dBm
Environmental (LGA package)				
Storage temperature		-40	125	°C
MSL	Moisture Sensitivity Level		2	
ESD HBM	Human Body Model		2	kV
ESD CDM	Charged Device Model		500	V
Flash memory				
Endurance		10 000		Write/erase cycles
Retention		10 years at 85°C		

Table 106: Absolute maximum ratings

13 Ordering information

This chapter contains information on IC marking, ordering codes, and container sizes.

13.1 IC marking

The nRF9160 IC package is marked like described below.

N	9	1	6	0	
<P	P>	<V	V>	<H>	<P>
<Y	Y>	<W	W>	<L	L>

Figure 114: Package marking

13.2 Box labels

Here are the box labels used for the nRF9160.

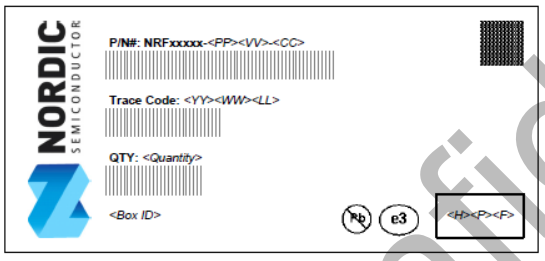

Figure 115: Inner box label

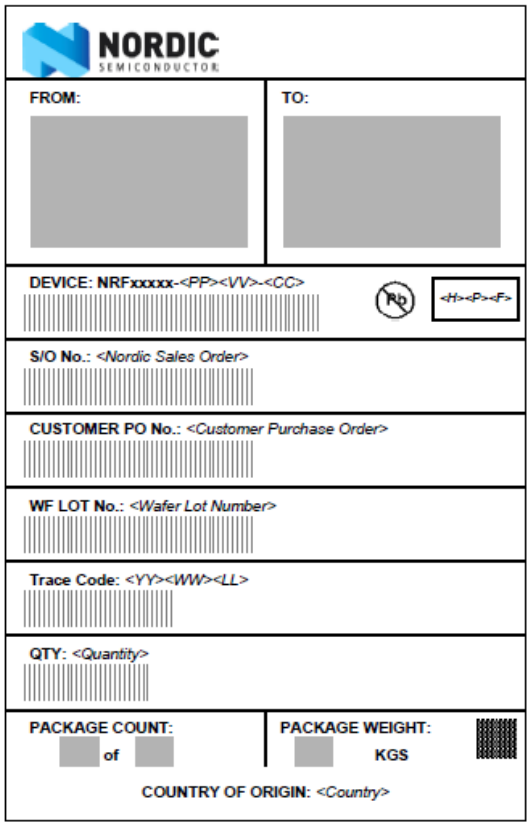

Figure 116: Outer box label

13.3 Order code

Here are the nRF9160 order codes and definitions.

n	R	F	9	1	6	0	-	<P	P>	<V	V>	-	<C	C>

Figure 117: Order code

Abbreviation	Definition and implemented codes
N91/nRF91	nRF91 Series product
60	Part code
<PP>	Package variant code
<VV>	Function variant code
<H><P><F>	Build code H - Hardware version code P - Production configuration code (production site, etc.) F - Firmware version code (only visible on shipping container label)
<YY><WW><LL>	Tracking code YY - Year code WW - Assembly week number LL - Wafer lot code
<CC>	Container code

Table 107: Abbreviations

13.4 Code ranges and values

Defined here are the nRF9160 code ranges and values.

<PP>	Package	Size (mm)	Pin/Ball count	Pitch (mm)

Table 108: Package variant codes

<VV>	Flash (kB)	RAM (kB)

Table 109: Function variant codes

<H>	Description
[A . . Z]	Hardware version/revision identifier (incremental)

Table 110: Hardware version codes

<P>	Description
[0 . . 9]	Production device identifier (incremental)
[A . . Z]	Engineering device identifier (incremental)

Table 111: Production configuration codes

<F>	Description
[A . . N, P . . Z]	Version of preprogrammed firmware
[0]	Delivered without preprogrammed firmware

Table 112: Production version codes

<YY>	Description
[15 . . 99]	Production year: 2015 to 2099

Table 113: Year codes

<WW>	Description
[1 . . 52]	Week of production

Table 114: Week codes

<LL>	Description
[AA . . ZZ]	Wafer production lot identifier

Table 115: Lot codes

<CC>	Description
R7	7" Reel
R	13" Reel
T	Tray

Table 116: Container codes

13.5 Product options

Defined here are the nRF9160 product options.

Order code	Minimum ordering quantity (MOQ)	Comment
nRF9160-SICA-R	2500	LTE-M/NB-IoT/GPS product
nRF9160-SIAA-R	2500	LTE-M only product
nRF9160-SIBA-R	2500	NB-IoT only product

Table 117: nRF9160 order codes

Order code	Description
nRF9160-DK	

Table 118: Development tools order code

14 Glossary/acronyms

Term/acronym	Definition
BCH	Broadcast channel
BW	Bandwidth
Cat-0	LTE-M user equipment (UE) category with 20 MHz UE bandwidth and single RX, specified in 3GPP Release 12
Cat-1	LTE-M user equipment (UE) category with 2RX, specified in 3GPP Release 8
Cat-M1	LTE-M user equipment (UE) category with 1.4 MHz UE bandwidth and single RX, specified in 3GPP Release 13
Cat-M2	NB-IoT user equipment (UE) category with 200 kHz UE bandwidth single RX, specified in 3GPP Release 13
CE	coverage extension
DL	downlink
DL-SCH	Dowlink Shared Channel
DRX	discontinous reception
ECID	Enhanced Cell ID
eDRX	extended discontinuous reception, enhanced DRX
EPDCCH	Enhanced Physical Downlink Control Channel
FDD	frequency division duplex
FDMA	frequency division multiple access
FFT	fast Fourier transform
GPS	Global Positioning System
HD	half-duplex
HW	hardware
IoT	Internet of Things
LTE	Long Term Evolution
LTE-M	LTE machine-type communication (MTC). 3GPP machine-type communication technology containing user equipment (UE) categories CAT-1, CAT-0, and CAT-M1 with UE bandwidth from 20 MHz to 1.4 MHz
MBB	mobile broadband
MPDCCH	Physical Downlink Control Channel for Machine-Type Communication (MTC)
MTC	machine-type communication
NB-IoT	narrowband IoT. 3GPP Internet of Things (IoT) technology with narrow UE bandwidth (200 kHz)
NPBCH	Narrowband Physical Broadcast Channel
NB-PSS/SSS	narrowband primary synchronization signal/secondary synchronization signal
NPDCCH	Narrowband Physical Dowlink Control Channel
NPDSCH	Narrowband Physical Dowlink Shared Channel
NPRACH	Narrowband Physical Random Access Channel
NPUSCH	Narrowband Physical Uplink Shared Channel
OFDM	orthogonal frequency division multiplexing
OTDOA	Observed Time Difference of Arrival. A positioning feature introduced in 3GPP Release 9. The time interval that is observed by a target device between the reception of downlink signals from two different cells. If a signal from cell 1 is received at the moment t1, and a signal from cell 2 is received at the moment t2, the OTDOA is t2 - t1

Term/acronym	Definition
PA	power amplifier
PCH	Paging Channel
PDCCH	Physical Downlink Control Channel
PDSCH	Physical Downlink Shared Channel
PRB	physical resource block
PSM	power saving mode
PSS	primary synchronization signal
PUSCH	Physical Uplink Shared Channel
QAM	quadrature amplitude modulation
QPSK	quadrature phase-shift keying
RRC	Radio Resource Control
RSRP	Reference Signal Received Power
SC-FDMA	single-carrier frequency division multiple access
SIB	system information block
SSS	secondary synchronization signal
SW	software
TBCC	tail-biting convolutional code
TX	transmit, transmission, transmitter
UE	user equipment
UL	uplink

15 FCC/ISED regulatory notices

Modification statement

Nordic Semiconductor has not approved any changes or modifications to this device by the user. Any changes or modifications could void the user's authority to operate the equipment.

Nordic Semiconductor n'approuve aucune modification apportée à l'appareil par l'utilisateur, quelle qu'en soit la nature. Tout changement ou modification peuvent annuler le droit d'utilisation de l'appareil par l'utilisateur.

Interference statement

This device complies with Part 15 of the FCC Rules and Industry Canada's licence-exempt RSS standards. Operation is subject to the following two conditions: (1) this device may not cause interference, and (2) this device must accept any interference, including interference that may cause undesired operation of the device.

Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes: (1) l'appareil ne doit pas produire de brouillage, et (2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

Wireless notice

This equipment complies with FCC and ISED radiation exposure limits set forth for an uncontrolled environment. The antenna should be installed and operated with minimum distance of 20 cm between the radiator and your body. This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.

Cet appareil est conforme aux limites d'exposition aux rayonnements de l'ISDE pour un environnement non contrôlé. L'antenne doit être installée de façon à garder une distance minimale de 20 centimètres entre la source de rayonnements et votre corps. L'émetteur ne doit pas être colocalisé ni fonctionner conjointement avec à autre antenne ou autre émetteur.

Permitted antenna

This radio transmitter has been approved by FCC and ISED to operate with the antenna types listed below with the maximum permissible gain indicated. Antenna types not included in this list, having a gain greater than the maximum gain indicated for that type, are strictly prohibited for use with this device.

Type	Band	Max gain
SMD	Band 4	6 dBi
	Band 13	6.9 dBi

Le présent émetteur radio a été approuvé par ISDE pour fonctionner avec les types d'antenne énumérés ci dessous et ayant un gain admissible maximal. Les types d'antenne non inclus dans cette liste, et dont le gain est supérieur au gain maximal indiqué, sont strictement interdits pour l'exploitation de l'émetteur.

Type	Bande	Gain maximal
CMS	Bande 4	6 dBi
	Bande 13	6.9 dBi

FCC Class B digital device notice

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna
- Increase the separation between the equipment and receiver
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected
- Consult the dealer or an experienced radio/TV technician for help

CAN ICES-3 (B)/NMB-3 (B)

This Class B digital apparatus complies with Canadian ICES-003.

Cet appareil numérique de classe B est conforme à la norme canadienne ICES-003.

Labeling requirements for the host device

The host device shall be properly labelled to identify the modules within the host device. The certification label of the module shall be clearly visible at all times when installed in the host device, otherwise the host device must be labelled to display the FCC ID and IC of the module, preceded by the words "Contains transmitter module", or the word "Contains", or similar wording expressing the same meaning, as follows:

Contains FCC ID: 2ANPO00NRF9160
Contains IC: 24529-NRF9160

L'équipement hôte doit être correctement étiqueté pour identifier les modules dans l'équipement. L'étiquette de certification du module doit être clairement visible en tout temps lorsqu'il est installé dans l'hôte, l'équipement hôte doit être étiqueté pour afficher le FCC ID et IC du module, précédé des mots "Contient le module émetteur", ou le mot "Contient", ou un libellé similaire exprimant la même signification, comme suit:

Contient FCC ID: 2ANPO00NRF9160
Contient IC: 24529-NRF9160

16 Liability disclaimer

Nordic Semiconductor ASA reserves the right to make changes without further notice to the product to improve reliability, function or design. Nordic Semiconductor ASA does not assume any liability arising out of the application or use of any product or circuits described herein.

16.1 RoHS and REACH statement

Nordic Semiconductor products meet the requirements of Directive 2002/95/EC of the European Parliament and of the Council on the Restriction of Hazardous Substances (RoHS) and the requirements of the REACH regulation (EC 1907/2006) on Registration, Evaluation, Authorization and Restriction of Chemicals.

The SVHC (Substances of Very High Concern) candidate list is continually being updated. Complete hazardous substance reports, material composition reports and latest version of Nordic's REACH statement can be found on our website .

16.2 Life support applications

Nordic Semiconductor products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury.

Nordic Semiconductor ASA customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nordic Semiconductor ASA for any damages resulting from such improper use or sale.