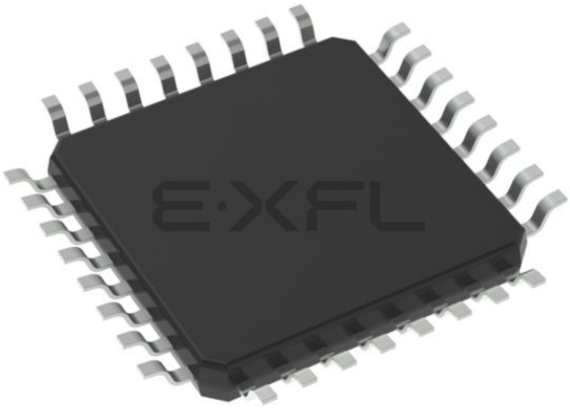


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM® Cortex®-M0+
Core Size	32-Bit Single-Core
Speed	48MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, WDT
Number of I/O	26
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 10x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	32-TQFP
Supplier Device Package	32-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsamc20e18a-ant

Name: STATUSA
Offset: 0x34 [ID-00000a18]
Reset: 0x000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				TSENS	FREQM	EIC	RTC	WDT
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM	PAC
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 12 – TSENS: Peripheral TSENS Write Protection Status

Bit 11 – FREQM: Peripheral FREQM Write Protection Status

Bit 10 – EIC: Peripheral EIC Write Protection Status

Bit 9 – RTC: Peripheral RTC Write Protection Status

Bit 8 – WDT: Peripheral WDT Write Protection Status

Bit 7 – GCLK: Peripheral GCLK Write Protection Status

Bit 6 – SUPC: Peripheral SUPC Write Protection Status

Bit 5 – OSC32KCTRL: Peripheral OSC32KCTRL Write Protection Status

Bit 4 – OSCCTRL: Peripheral OSCCTRL Write Protection Status

Bit 3 – RSTC: Peripheral RSTC Write Protection Status

Bit 2 – MCLK: Peripheral MCLK Write Protection Status

Bit 1 – PM: Peripheral PM Write Protection Status

Bit 0 – PAC: Peripheral PAC Write Protection Status

18.5.1 I/O Lines

Not applicable.

18.5.2 Power Management

The Reset Controller module is always on.

18.5.3 Clocks

The RSTC bus clock (CLK_RSTC_APB) can be enabled and disabled in the Main Clock Controller.

Related Links

MCLK – Main Clock
Peripheral Clock Masking

18.5.4 DMA

Not applicable.

18.5.5 Interrupts

Not applicable.

18.5.6 Events

Not applicable.

18.5.7 Debug Operation

When the CPU is halted in debug mode, the RSTC continues normal operation.

18.5.8 Register Access Protection

All registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC).

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

18.5.9 Analog Connections

Not applicable.

18.6 Functional Description

18.6.1 Principle of Operation

The Reset Controller collects the various Reset sources and generates Reset for the device.

18.6.2 Basic Operation

18.6.2.1 Initialization

After a power-on Reset, the RSTC is enabled and the Reset Cause (RCAUSE) register indicates the POR source.

18.6.2.2 Enabling, Disabling, and Resetting

The RSTC module is always enabled.

Name: INTENCLR
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	OVF							CMP0
Access	R/W							R/W
Reset	0							0

Bit	7	6	5	4	3	2	1	0
	PERn	PERn	PERn	PERn	PERn	PERn	PERn	PERn
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF: Overflow Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 8 – CMP0: Compare 0 Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Compare 0 Interrupt Enable bit, which disables the Compare 0 interrupt.

Value	Description
0	The Compare 0 interrupt is disabled.
1	The Compare 0 interrupt is enabled.

Bits 7:0 – PERn: Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Periodic Interval n Interrupt Enable bit, which disables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

24.8.4 Interrupt Enable Set in COUNT32 mode (CTRLA.MODE=0)

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

24.11 Register Summary - CLOCK

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	MATCHCLR	CLKREP			MODE[1:0]		ENABLE	SWRST
0x01		15:8	CLOCKSYNC				PRESCALER[3:0]			
0x02 ... 0x03	Reserved									
0x04	EVCTRL	7:0	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn
0x05		15:8	OVFEO							ALARMO0
0x06		23:16								
0x07		31:24								
0x08	INTENCLR	7:0	PERn	PERn	PERn	PERn	PERn	PERn	PERn	PERn
0x09		15:8	OVF							ALARM0
0x0A	INTENSET	7:0	PERn	PERn	PERn	PERn	PERn	PERn	PERn	PERn
0x0B		15:8	OVF							ALARM0
0x0C	INTFLAG	7:0	PERn	PERn	PERn	PERn	PERn	PERn	PERn	PERn
0x0D		15:8	OVF							ALARM0
0x0E	DBGCTRL	7:0								DBGRUN
0x0F	Reserved									
0x10	SYNCBUSY	7:0			ALARM0		CLOCK	FREQCORR	ENABLE	SWRST
0x11		15:8	CLOCKSYNC				MASK0			
0x12		23:16								
0x13		31:24								
0x14	FREQCORR	7:0	SIGN	VALUE[6:0]						
0x15 ... 0x17	Reserved									
0x18	CLOCK	7:0	MINUTE[1:0]		SECOND[5:0]					
0x19		15:8	HOUR[3:0]				MINUTE[5:2]			
0x1A		23:16	MONTH[1:0]		DAY[4:0]					HOUR[4:4]
0x1B		31:24	YEAR[5:0]						MONTH[3:2]	
0x1C ... 0x1F	Reserved									
0x20	ALARM	7:0	MINUTE[1:0]		SECOND[5:0]					
0x21		15:8	HOUR[3:0]				MINUTE[5:2]			
0x22		23:16	MONTH[1:0]		DAY[4:0]					HOUR[4:4]
0x23		31:24	YEAR[5:0]						MONTH[3:2]	
0x24	MASK	7:0						SEL[2:0]		

24.12 Register Description - CLOCK

This Register Description section is valid if the RTC is in Clock/Calendar mode (CTRLA.MODE=2).

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Bit	31	30	29	28	27	26	25	24
	DESCADDR[31:24]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DESCADDR[23:16]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DESCADDR[15:8]							
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DESCADDR[7:0]							
Access								
Reset								

Bits 31:0 – DESCADDR[31:0]: Next Descriptor Address
This bit group holds the SRAM address of the next descriptor. The value must be 128-bit aligned. If the value of this SRAM register is 0x00000000, the transaction will be terminated when the DMAC tries to load the next transfer descriptor.

Bit	7	6	5	4	3	2	1	0
							ERROR	READY
Access							R/W	R/W
Reset							0	0

Bit 1 – ERROR: Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the ERROR interrupt enable.

This bit will read as the current value of the ERROR interrupt enable.

Bit 0 – READY: NVM Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the READY interrupt enable.

This bit will read as the current value of the READY interrupt enable.

27.8.5 Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Name: INTENSET
Offset: 0x10 [ID-00000b2c]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
							ERROR	READY
Access							R/W	R/W
Reset							0	0

Bit 1 – ERROR: Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit sets the ERROR interrupt enable.

This bit will read as the current value of the ERROR interrupt enable.

Bit 0 – READY: NVM Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit sets the READY interrupt enable.

This bit will read as the current value of the READY interrupt enable.

27.8.6 Interrupt Flag Status and Clear

Table 30-1. SERCOM Modes

CTRLA.MODE	Description
0x0	USART with external clock
0x1	USART with internal clock
0x2	SPI in slave operation
0x3	SPI in master operation
0x4	I^2C slave operation
0x5	I^2C master operation
0x6-0x7	Reserved

For further initialization information, see the respective SERCOM mode chapters:

Related Links

SERCOM USART – SERCOM Universal Synchronous and Asynchronous Receiver and Transmitter
SERCOM SPI – SERCOM Serial Peripheral Interface
SERCOM I2C – SERCOM Inter-Integrated Circuit

30.6.2.2 Enabling, Disabling, and Resetting

This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE), and disabled by writing '0' to it.

Writing ‘1’ to the Software Reset bit in the Control A register (CTRLA.SWRST) will reset all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

Refer to the CTRLA register description for details.

30.6.2.3 Clock Generation – Baud-Rate Generator

The baud-rate generator, as shown in Figure 30-3, generates internal clocks for asynchronous and synchronous communication. The output frequency (f_{BAUD}) is determined by the Baud register (BAUD) setting and the baud reference frequency (f_{ref}). The baud reference clock is the serial engine clock, and it can be internal or external.

For asynchronous communication, the /16 (divide-by-16) output is used when transmitting, whereas the /1 (divide-by-1) output is used while receiving.

For synchronous communication, the /2 (divide-by-2) output is used.

This functionality is automatically configured, depending on the selected operating mode.

- External clocking, CTRLA.RUNSTDBY=0: External clock will be disconnected, after any ongoing transfer was completed. All reception will be dropped.

31.6.6 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in the CTRLA register (CTRLA.SWRST)
- Enable bit in the CTRLA register (CTRLA.ENABLE)
- Receiver Enable bit in the CTRLB register (CTRLB.RXEN)
- Transmitter Enable bit in the Control B register (CTRLB.TXEN)

Note: CTRLB.RXEN is write-synchronized somewhat differently. See also CTRLB for details.

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Related Links

Register Synchronization

RXPO[1:0]	Name	Description
0x0	PAD[0]	SERCOM PAD[0] is used for data reception
0x1	PAD[1]	SERCOM PAD[1] is used for data reception
0x2	PAD[2]	SERCOM PAD[2] is used for data reception
0x3	PAD[3]	SERCOM PAD[3] is used for data reception

Bits 17:16 – TXPO[1:0]: Transmit Data Pinout
These bits define the transmit data (TxD) and XCK pin configurations.

This bit is not synchronized.

TXPO	TxD Pin Location	XCK Pin Location (When Applicable)	RTS/TE	CTS
0x0	SERCOM PAD[0]	SERCOM PAD[1]	N/A	N/A
0x1	SERCOM PAD[2]	SERCOM PAD[3]	N/A	N/A
0x2	SERCOM PAD[0]	N/A	SERCOM PAD[2]	SERCOM PAD[3]
0x3	SERCOM_PAD[0]	SERCOM_PAD[1]	SERCOM_PAD[2]	N/A

Bits 15:13 – SAMPR[2:0]: Sample Rate
These bits select the sample rate.

These bits are not synchronized.

SAMPR[2:0]	Description
0x0	16x over-sampling using arithmetic baud rate generation.
0x1	16x over-sampling using fractional baud rate generation.
0x2	8x over-sampling using arithmetic baud rate generation.
0x3	8x over-sampling using fractional baud rate generation.
0x4	3x over-sampling using arithmetic baud rate generation.
0x5-0x7	Reserved

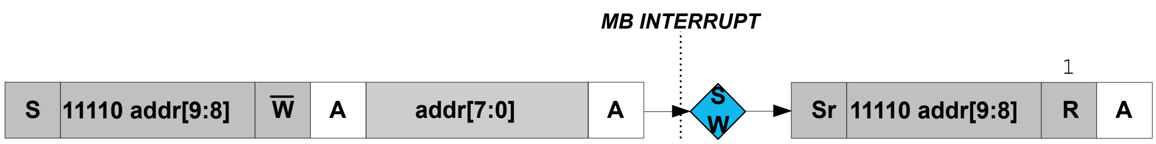
Bit 8 – IBON: Immediate Buffer Overflow Notification
This bit controls when the buffer overflow status bit (STATUS.BUFOVF) is asserted when a buffer overflow occurs.

Value	Description
0	STATUS.BUFOVF is asserted when it occurs in the data stream.
1	STATUS.BUFOVF is asserted immediately upon buffer overflow.

Bit 7 – RUNSTDBY: Run In Standby
This bit defines the functionality in standby sleep mode.

This bit is not synchronized.

Figure 33-9. 10-bit Address Transmission for a Read Transaction



This implies the following procedure for a 10-bit read operation:

1. Write the 10-bit address to ADDR.ADDR[10:1]. ADDR.TENBITEN must be '1', the direction bit (ADDR.ADDR[0]) must be '0' (can be written simultaneously with ADDR).
2. Once the Master on Bus interrupt is asserted, Write ADDR[7:0] register to '11110 `address[9:8]` 1'. ADDR.TENBITEN must be cleared (can be written simultaneously with ADDR).
3. Proceed to transmit data.

33.6.2.5 I²C Slave Operation

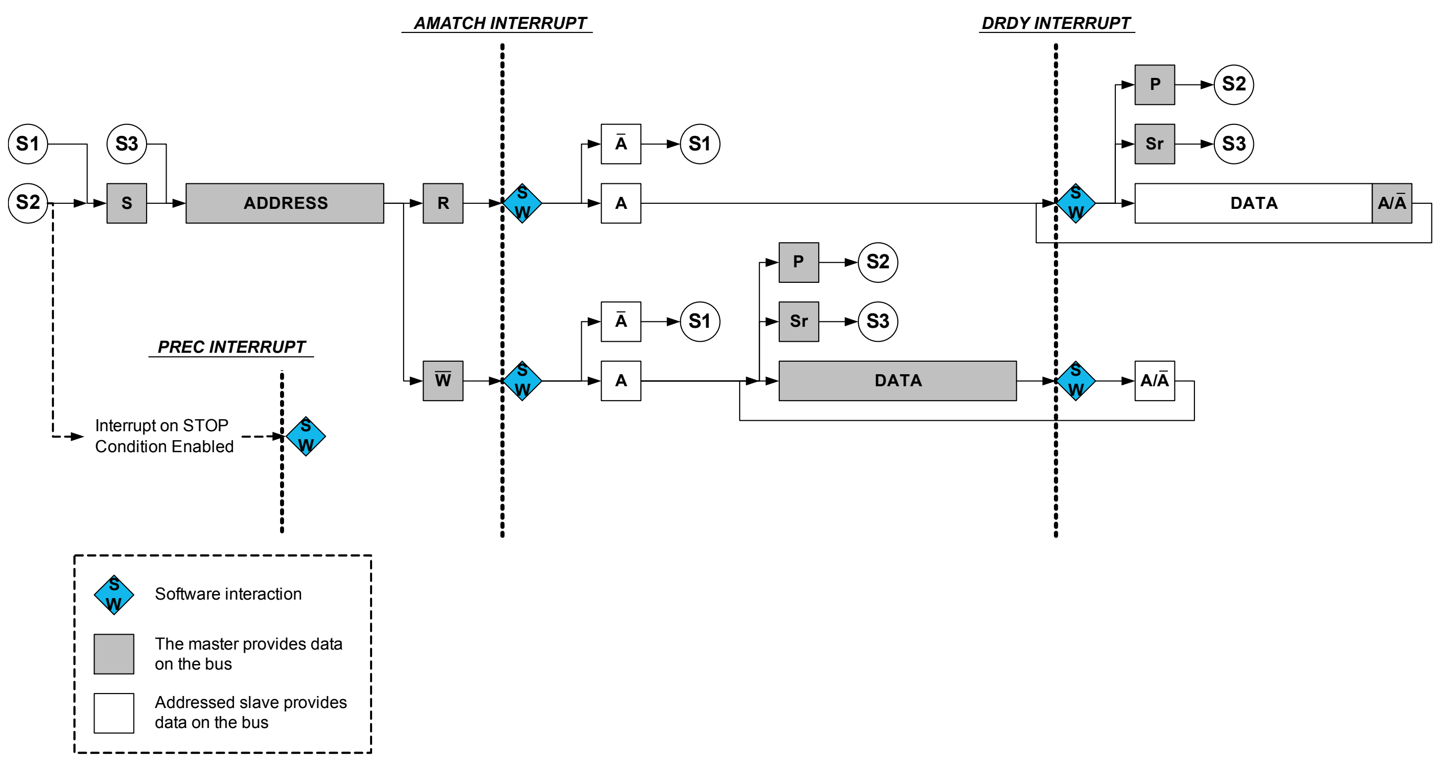
The I²C slave is byte-oriented and interrupt-based. The number of interrupts generated is kept at a minimum by automatic handling of most events. The software driver complexity and code size are reduced by auto-triggering of operations, and a special smart mode, which can be enabled by the Smart Mode Enable bit in the Control A register (CTRLA.SMEN).

The I²C slave has two interrupt strategies.

When SCL Stretch Mode bit (CTRLA.SCLSM) is '0', SCL is stretched before or after the acknowledge bit. In this mode, the I²C slave operates according to I²C Slave Behavioral Diagram (SCLSM=0). The circles labelled "S*n*" (S1, S2..) indicate the nodes the bus logic can jump to, based on software or hardware interaction.

This diagram is used as reference for the description of the I²C slave operation throughout the document.

Figure 33-10. I²C Slave Behavioral Diagram (SCLSM=0)

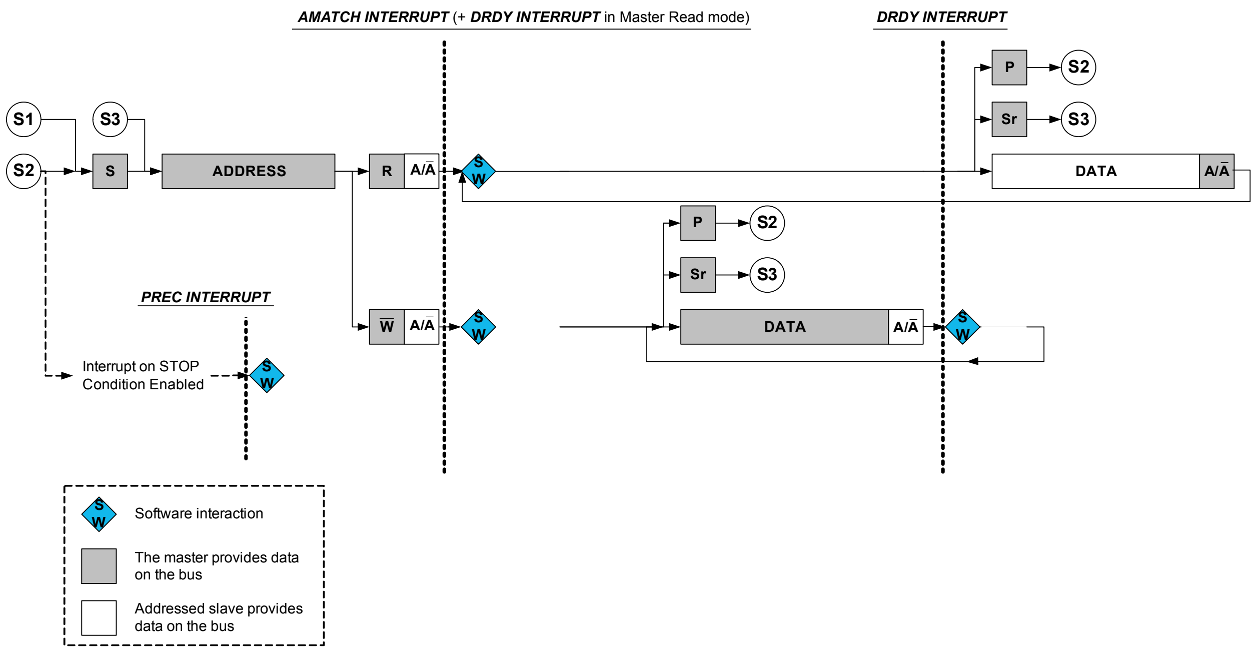


In the second strategy (CTRLA.SCLSM=1), interrupts only occur after the ACK bit is sent as shown in Slave Behavioral Diagram (SCLSM=1). This strategy can be used when it is not necessary to check

DATA before acknowledging. For master reads, an address and data interrupt will be issued simultaneously after the address acknowledge. However, for master writes, the first data interrupt will be seen after the first data byte has been received by the slave and the acknowledge bit has been sent to the master.

Note: For I²C High-speed mode (*Hs*), SCLSM=1 is required.

Figure 33-11. I²C Slave Behavioral Diagram (SCLSM=1)



Receiving Address Packets (SCLSM=0)

When CTRLA.SCLSM=0, the I2C slave stretches the SCL line according to Figure 33-10. When the I²C slave is properly configured, it will wait for a start condition.

When a start condition is detected, the successive address packet will be received and checked by the address match logic. If the received address is not a match, the packet will be rejected, and the I²C slave will wait for a new start condition. If the received address is a match, the Address Match bit in the Interrupt Flag register (INTFLAG.AMATCH) will be set.

SCL will be stretched until the I²C slave clears INTFLAG.AMATCH. As the I²C slave holds the clock by forcing SCL low, the software has unlimited time to respond.

The direction of a transaction is determined by reading the Read / Write Direction bit in the Status register (STATUS.DIR). This bit will be updated only when a valid address packet is received.

If the Transmit Collision bit in the Status register (STATUS.COLL) is set, this indicates that the last packet addressed to the I²C slave had a packet collision. A collision causes the SDA and SCL lines to be released without any notification to software. Therefore, the next AMATCH interrupt is the first indication of the previous packet's collision. Collisions are intended to follow the SMBus Address Resolution Protocol (ARP).

After the address packet has been received from the I²C master, one of two cases will arise based on transfer direction.

Case 1: Address packet accepted – Read flag set

The STATUS.DIR bit is '1', indicating an I²C master read operation. The SCL line is forced low, stretching the bus clock. If an ACK is sent, I²C slave hardware will set the Data Ready bit in the Interrupt Flag

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

35.7.1.3 Control B Set

This register allows the user to set bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Clear register (CTRLBCLR).

Name: CTRLBSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection, Read-synchronized, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0]: Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command has been executed, the CMD bit group will be read back as zero.

Writing 0x0 to these bits has no effect.

Writing a value different from 0x0 to these bits will issue a command for execution.

Value	Name	Description
0x0	NONE	No action
0x1	RETRIGGER	Force a start, restart or retrigger
0x2	STOP	Force a stop
0x3	UPDATE	Force update of double buffered registers
0x4	READSYNC	Force a read synchronization of COUNT

Bit 2 – ONESHOT: One-Shot on Counter
This bit controls one-shot operation of the TC.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will enable one-shot operation.

Value	Description
0	The TC will wrap around and continue counting on an overflow/underflow condition.
1	The TC will wrap around and stop on the next underflow/overflow condition.

Bit 1 – LUPD: Lock Update
This bit controls the update operation of the TC buffered registers.

When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the LUPD bit.

This bit has no effect when input capture operation is enabled.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition.

Bit 0 – DIR: Counter Direction
This bit is used to change the direction of the counter.

Writing a '0' to this bit has no effect

Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

35.7.1.4 Event Control

Name: EVCTRL
Offset: 0x06
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
			MCEOx	MCEOx				OVFEO
Access			R/W	R/W				R/W
Reset			0	0				0

Bit	7	6	5	4	3	2	1	0
			TCEI	TCINV		EVACT[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bits 13,12 – MCEOx: Match or Capture Channel x Event Output Enable [x = 1..0]
These bits enable the generation of an event for every match or capture on channel x.

Value	Description
0	Match/Capture event on channel x is disabled and will not be generated.
1	Match/Capture event on channel x is enabled and will be generated for every compare/ capture.

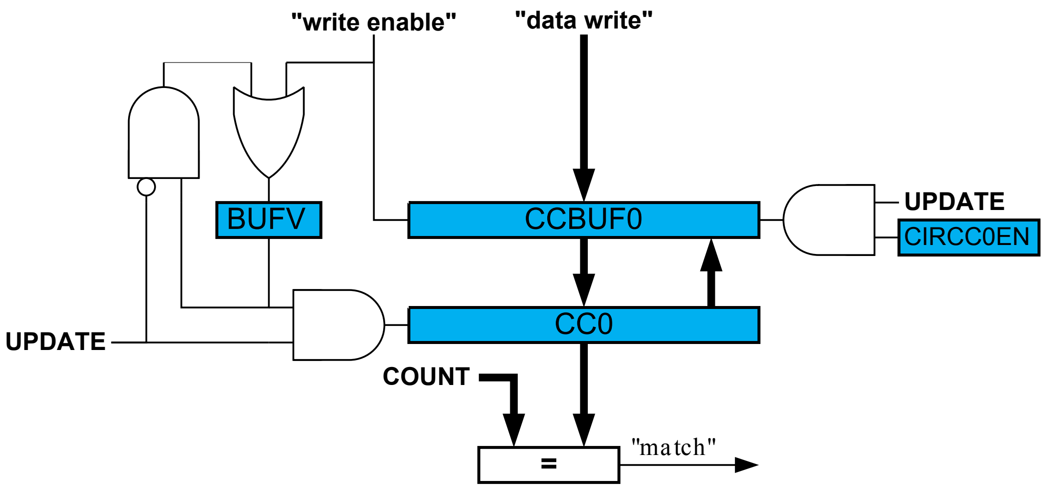
Bit 8 – OVFEO: Overflow/Underflow Event Output Enable
This bit enables the Overflow/Underflow event. When enabled, an event will be generated when the counter overflows/underflows.

One-shot operation can be enabled by writing a '1' to the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) and disabled by writing a '1' to CTRLBCLR.ONESHOT. When enabled, the TCC will count until an overflow or underflow occurs and stop counting. The one-shot operation can be restarted by a re-trigger software command, a re-trigger event or a start event. When the counter restarts its operation, STATUS.STOP is automatically cleared.

36.6.3.2 Circular Buffer

The Period register (PER) and the compare channels register (CC0 to CC3) support circular buffer operation. When circular buffer operation is enabled, the PER or CCx values are copied into the corresponding buffer registers at each update condition. Circular buffering is dedicated to RAMP2, RAMP2A, and DSBOTH operations.

Figure 36-17. Circular Buffer on Channel 0



36.6.3.3 Dithering Operation

The TCC supports dithering on Pulse-width or Period on a 16, 32 or 64 PWM cycles frame.

Dithering consists in adding some extra clocks cycles in a frame of several PWM cycles, and can improve the accuracy of the *average* output pulse width and period. The extra clock cycles are added on some of the compare match signals, one at a time, through a "blue noise" process that minimizes the flickering on the resulting dither patterns.

Dithering is enabled by writing the corresponding configuration in the Enhanced Resolution bits in CTRLA register (CTRLA.RESOLUTION):

- DITH4 enable dithering every 16 PWM frames
- DITH5 enable dithering every 32 PWM frames
- DITH6 enable dithering every 64 PWM frames

The DITHERCY bits of COUNT, PER and CCx define the number of extra cycles to add into the frame (DITHERCY bits from the respective COUNT, PER or CCx registers). The remaining bits of COUNT, PER, CCx define the compare value itself.

The pseudo code, giving the extra cycles insertion regarding the cycle is:

```
int extra_cycle(resolution, dithercy, cycle){
  int MASK;
  int value
  switch (resolution){
    DITH4: MASK = 0x0f;
    DITH5: MASK = 0x1f;
    DITH6: MASK = 0x3f;
  }
```

Number of Accumulated Samples	AVGCTRL. SAMPLENUM	Number of Automatic Right Shifts	Final Result Precision	Automatic Division Factor
4	0x2	0	14 bits	0
8	0x3	0	15 bits	0
16	0x4	0	16 bits	0
32	0x5	1	16 bits	2
64	0x6	2	16 bits	4
128	0x7	3	16 bits	8
256	0x8	4	16 bits	16
512	0x9	5	16 bits	32
1024	0xA	6	16 bits	64
Reserved	0xB –0xF		12 bits	0

38.6.2.10 Averaging

Averaging is a feature that increases the sample accuracy, at the cost of a reduced sampling rate. This feature is suitable when operating in noisy conditions.

Averaging is done by accumulating m samples, as described in Accumulation, and dividing the result by m. The averaged result is available in the RESULT register. The number of samples to be accumulated is specified by writing to AVGCTRL.SAMPLENUM as shown in Table 38-2.

The division is obtained by a combination of the automatic right shift described above, and an additional right shift that must be specified by writing to the Adjusting Result/Division Coefficient field in AVGCTRL (AVGCTRL.ADJRES), as described in Table 38-2.

Note: To perform the averaging of two or more samples, the Conversion Result Resolution field in the Control C register (CTRLC.RESSEL) must be set.

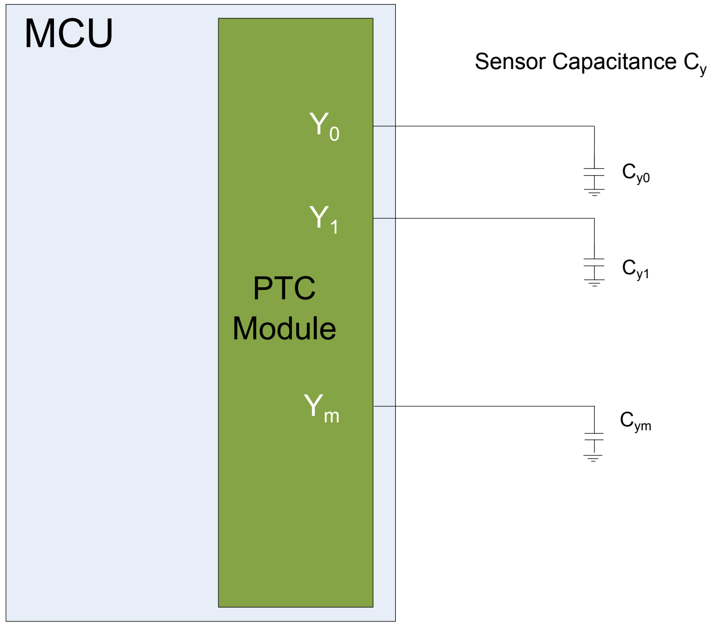
Averaging AVGCTRL.SAMPLENUM samples will reduce the un-averaged sampling rate by a factor $\frac{1}{\mathrm{AVGCTRL.SAMPLENUM}}$.

When the averaged result is available, the INTFLAG.RESRDY bit will be set.

Table 38-2. Averaging

Number of Accumulated Samples	AVGCTRL. SAMPLENUM	Intermediate Result Precision	Number of Automatic Right Shifts	Division Factor	AVGCTRL.ADJRES	Total Number of Right Shifts	Final Result Precision	Automatic Division Factor
1	0x0	12 bits	0	1	0x0		12 bits	0
2	0x1	13	0	2	0x1	1	12 bits	0
4	0x2	14	0	4	0x2	2	12 bits	0
8	0x3	15	0	8	0x3	3	12 bits	0
16	0x4	16	0	16	0x4	4	12 bits	0
32	0x5	17	1	16	0x4	5	12 bits	2

Figure 42-4. Self-capacitance Sensor Arrangement



For more information about designing the touch sensor, refer to Buttons, Sliders and Wheels Touch Sensor Design Guide.

42.5.2 Analog-Digital Converter (ADC)

The PTC is using the ADC for signal conversion and acquisition. The ADC must be enabled and configured appropriatly in order to allow correct behavior of the PTC.

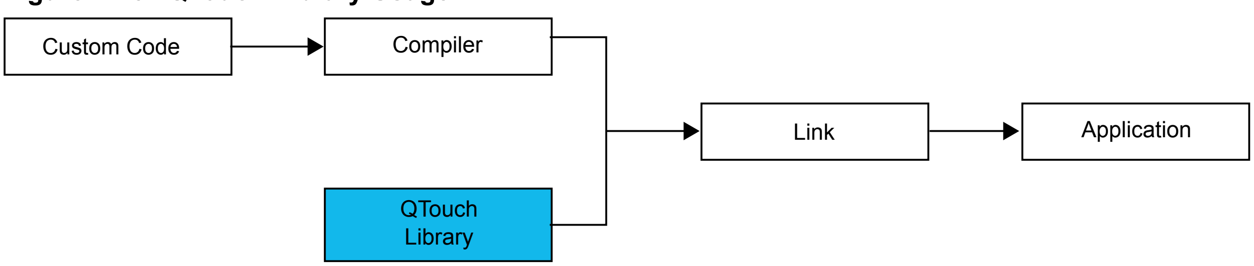
Related Links

ADC – Analog-to-Digital Converter

42.6 Functional Description

In order to access the PTC, the user must use the QTouch Composer tool to configure and link the QTouch Library firmware with the application software. QTouch Library can be used to implement buttons, sliders, wheels in a variety of combinations on a single interface.

Figure 42-5. QTouch Library Usage



For more information about QTouch Library, refer to the QTouch Library Peripheral Touch Controller User Guide.

43.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

43.5.1 I/O Lines

Not applicable.

43.5.2 Power Management

The TSENS will continue to operate in any sleep mode where the selected source clock is running. The TSENS's interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes. Refer to the Power Manager chapter for details on the different sleep modes.

43.5.3 Clocks

The TSENS bus clock (CLK_TSENS_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_TSENS_APB can be found in the Peripheral Clock Masking section.

A generic clock (GCLK_TSENS) is required to clock the TSENS. This clock must be configured and enabled in the generic clock controller before using the TSENS.

This generic clock is asynchronous to the bus clock (CLK_TSENS_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to Synchronization for details.

Related Links

Peripheral Clock Masking

GCLK - Generic Clock Controller

43.5.4 DMA

The DMA request line is connected to the DMA Controller (DMAC). Using the TSENS Controller DMA request requires the DMA Controller to be configured first.

Related Links

DMAC – Direct Memory Access Controller

43.5.5 Interrupts

The interrupt request lines are connected to the interrupt controller. Using the TSENS interrupts requires the interrupt controller to be configured first.

Related Links

Nested Vector Interrupt Controller

43.5.6 Events

The events are connected to the Event System. Refer to the Event System section for details on how to configure the Event System.

Related Links

EVSYS – Event System

43.5.7 Debug Operation

When the CPU is halted in debug mode the TSENS will halt normal operation. Any on-going measurements will be completed. The TSENS can be forced to continue operation during debugging. Refer to DBGCTRL for details.

44.6.2.3 Measurement

In the Configuration A register, the Number of Reference Clock Cycles field (CFGA.REFNUM) selects the duration of the measurement. The measurement is given in number of GCLK_FREQM_REF periods.

Note: The REFNUM field must be written before the FREQM is enabled.

After the FREQM is enabled, writing a '1' to the START bit in the Control B register (CTRLB.START) starts the measurement. The BUSY bit in Status register (STATUS.BUSY) is set when the measurement starts, and cleared when the measurement is complete.

There is also an interrupt request for Measurement Done: When the Measurement Done bit in Interrupt Enable Set register (INTENSET.DONE) is '1' and a measurement is finished, the Measurement Done bit in the Interrupt Flag Status and Clear register (INTFLAG.DONE) will be set and an interrupt request is generated.

The result of the measurement can be read from the Value register (VALUE.VALUE). The frequency of the measured clock GCLK_FREQM_MSR is then:

$$f_{\mathrm{CLK\_MSR}} = \left(\frac{\mathrm{VALUE}}{\mathrm{REFNUM}}\right) f_{\mathrm{CLK\_REF}}$$

Note: In order to make sure the measurement result (VALUE.VALUE[23:0]) is valid, the overflow status (STATUS.OVF) should be checked.

In case an overflow condition occurred, indicated by the Overflow bit in the STATUS register (STATUS.OVF), either the number of reference clock cycles must be reduced (CFGA.REFNUM), or a faster reference clock must be configured. Once the configuration is adjusted, clear the overflow status by writing a '1' to STATUS.OVF. Then another measurement can be started by writing a '1' to CTRLB.START.

44.6.3 DMA Operation

Not applicable.

44.6.4 Interrupts

The FREQM has one interrupt source:

- DONE: A frequency measurement is done.

The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. The interrupt can be enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the FREQM is reset. See INTFLAG for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAG register to determine which interrupt condition is present.

This interrupt is a synchroneous wake-up source.

Note that interrupts must be globally enabled for interrupt requests to be generated.

44.6.5 Events

Not applicable.

44.6.6 Sleep Mode Operation

The FREQM will continue to operate in idle sleep mode where the selected source clock is running. The FREQM's interrupts can be used to wake up the device from idle sleep mode.

Symbol	Parameters	Conditions	Ta	Typ.	Max	Units
	STANDBY, Mode SAMPL	VDD = 2.7V		0.8	2.1	
		VDD = 5.0V		3.5	4.9	

Note:

1. These values are based on characterization.

Table 46-4. BODVDD Characteristics (see Note 2)

Symbol	Parameters	Conditions	Min	Typ	Max	Unit
VBOD+ (see **Note 1**)	BODVDD high threshold Level	VDD level, BOD setting = 8 (default)	-	2.86	2.98	V
		VDD level, BOD setting = 9	-	2.92	3.01	
		VDD level, BOD setting = 44	-	4.57	4.82	
VBOD- / VBOD (see **Note 1**)	BODVDD low threshold Level	VDD level, BOD setting = 8 (default)	2.71	2.8	2.90	
		VDD level, BOD setting = 9	2.75	2.85	2.96	
		VDD level, Bod setting = 44	4.37	4.51	4.66	
	Step size		-	60	-	mV
VHys (see **Note 1**)	Hysteresis (VBOD+ - VBOD-) BODVDD.LEVEL = 8 to 48	VDD	40	-	75	mV
Tstart (see **Note 3**)	Startup time	Time from enable to RDY	-	3.1	-	μs

Note:

1. These values are based on characterization.
2. BODVDD in Continuous mode.
3. These values are based on simulation, and are not covered by test or characterization.

Related Links

NVM User Row Mapping

NVM User Row Mapping

46.4.2 Analog-to-Digital Converter (ADC) Characteristics

Table 46-5. Power Consumption(1)

Symbol	Parameters	Conditions	Ta	Typ.	Max	Units
IDD VDDANA	Differential mode	fs = 1 Msps / Reference buffer disabled / BIASREFBUF = '111',	Max 105°C Typ 25°C	905	1034	μA

1. These values are only given as a typical example.
2. The capacitors should be placed close to the device for each supply pin pair in the signal group.

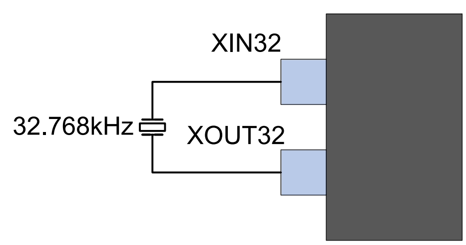
49.7.3 External Real Time Oscillator

The low frequency crystal oscillator is optimized for use with a 32.768kHz watch crystal. When selecting crystals, load capacitance and the crystal's Equivalent Series Resistance (ESR) must be taken into consideration. Both values are specified by the crystal vendor.

SAM C20/C21 oscillator is optimized for very low power consumption, hence close attention should be made when selecting crystals.

The typical parasitic load capacitance values are available in the Electrical Characteristics section. This capacitance and PCB capacitance can allow using a crystal inferior to 12.5pF load capacitance without external capacitors as shown in Figure 49-8.

Figure 49-8. External Real Time Oscillator without Load Capacitor



To improve accuracy and Safety Factor, the crystal datasheet can recommend adding external capacitors as shown in Figure 49-9.

To find suitable load capacitance for a 32.768kHz crystal, consult the crystal datasheet.

Figure 49-9. External Real Time Oscillator with Load Capacitor

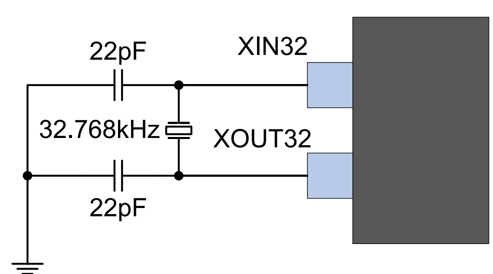


Table 49-6. External Real Time Oscillator Checklist

Signal Name	Recommended Pin Connection	Description
XIN32	Load capacitor 22pF(1)(2)	Timer oscillator input
XOUT32	Load capacitor 22pF(1)(2)	Timer oscillator output

1. These values are only given as typical examples.
2. The capacitors should be placed close to the device for each supply pin pair in the signal group.

Note: In order to minimize the cycle-to-cycle jitter of the external oscillator, keep the neighboring pins as steady as possible. For neighboring pin details, refer to the Oscillator Pinout section.