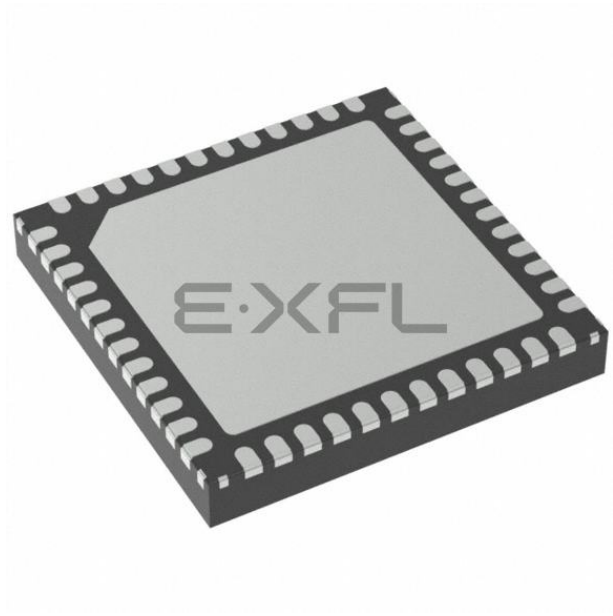


Welcome to E-XFL.COM

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	Obsolete
Core Processor	ARM® Cortex®-M0+
Core Size	32-Bit Single-Core
Speed	48MHz
Connectivity	I²C, LINbus SBC, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	32
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	0V ~ 3.8V
Data Converters	A/D 13x12b; D/A 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TC)
Mounting Type	Surface Mount, Wettable Flank
Package / Case	48-VFQFN Exposed Pad
Supplier Device Package	48-VQFN (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsamha1g15a-mbt

All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. Refer to “Nested Vector Interrupt Controller” on page 36 for details. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated. Refer to “Nested Vector Interrupt Controller” on page 36 for details.

19.6.14 Synchronization

Due to the multiple clock domains, values in the DFLL48M control registers need to be synchronized to other clock domains. The status of this synchronization can be read from the Power and Clocks Status register (PCLKSR). Before writing to any of the DFLL48M control registers, the user must check that the DFLL Ready bit (PCLKSR.DFLLRDY) in PCLKSR is set to one. When this bit is set, the DFLL48M can be configured and CLK_DFLL48M is ready to be used. Any write to any of the DFLL48M control registers while DFLLRDY is zero will be ignored. An interrupt is generated on a zero-to-one transition of DFLLRDY if the DFLLRDY bit (INTENSET.DFLLDY) in the Interrupt Enable Set register is set.

In order to read from any of the DFLL48M configuration registers, the user must request a read synchronization by writing a one to DFLLSYNC.READREQ. The registers can be read only when PCLKSR.DFLLRDY is set. If DFLLSYNC.READREQ is not written before a read, a synchronization will be started, and the bus will be halted until the synchronization is complete. Reading the DFLL48M registers when the DFLL48M is disabled will not halt the bus.

The prescaler counter used to trigger one-shot brown-out detections also operates asynchronously from the peripheral bus. As a consequence, the prescaler registers require synchronization when written or read. The synchronization results in a delay from when the initialization of the write or read operation begins until the operation is complete.

The write-synchronization is triggered by a write to the BOD33 control register. The Synchronization Ready bit (PCLKSR.B33SRDY) in the PCLKSR register will be cleared when the write-synchronization starts and set when the write-synchronization is complete. When the write-synchronization is ongoing (PCLKSR.B33SRDYis zero), an attempt to do any of the following will cause the peripheral bus to stall until the synchronization is complete:

- Writing to the BOD33control register
- Reading the BOD33 control register that was written

The user can either poll PCLKSR.B33SRDY or use the INTENSET.B33SRDY interrupts to check when the synchronization is complete. It is also possible to perform the next read/write operation and wait, as this next operation will be completed after the ongoing read/write operation is synchronized.

21.8.26 Alarm 0 Value - MODE2

Name: ALARM0

Offset: 0x18

Reset: 0x00000000

Property: Write-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	YEAR[5:0]						MONTH[3:2]	
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	23	22	21	20	19	18	17	16
	MONTH[1:0]		DAY[4:0]					HOUR[4]
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	14	13	12	11	10	9	8
	HOUR[3:0]				MINUTE[5:2]			
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	7	6	5	4	3	2	1	0
	MINUTE[1:0]		SECOND[5:0]					
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

The 32-bit value of ALARM0 is continuously compared with the 32-bit CLOCK value, based on the masking set by MASKn.SEL. When a match occurs, the Alarm 0 interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.ALARMn) is set on the next counter cycle, and the counter is cleared if CTRL.MATCHCLR is one.

- **Bits 31:26 – YEAR[5:0]: Year**
  The alarm year. Years are only matched if MASKn.SEL is 6.
- **Bits 25:22 – MONTH[3:0]: Month**
  The alarm month. Months are matched only if MASKn.SEL is greater than 4.
- **Bits 21:17 – DAY[4:0]: Day**
  The alarm day. Days are matched only if MASKn.SEL is greater than 3.
- **Bits 16:12 – HOUR[4:0]: Hour**
  The alarm hour. Hours are matched only if MASKn.SEL is greater than 2.
- **Bits 11:6 – MINUTE[5:0]: Minute**
  The alarm minute. Minutes are matched only if MASKn.SEL is greater than 1.
- **Bits 5:0 – SECOND[5:0]: Second**
  The alarm second. Seconds are matched only if MASKn.SEL is greater than 0.

22.8.1.2 CRC Control

Name: CRCCTRL

Offset: 0x02

Reset: 0x0000

Property: Enable-Protected, Write-Protected

Bit	15	14	13	12	11	10	9	8
			CRCSRC[5:0]					
Access	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					CRCPOLY[1:0]		CRCBEATSIZE[1:0]	
Access	R	R	R	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

- **Bits 15:14 – Reserved**
  These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written. These bits will always return zero when read.

- **Bits 13:8 – CRCSRC[5:0]: CRC Input Source**
  These bits select the input source for generating the CRC, as shown in Table 22-3. The selected source is locked until either the CRC generation is completed or the CRC module is disabled. This means the CRCSRC cannot be modified when the CRC operation is ongoing. The lock is signaled by the CRCBUSY status bit. CRC generation complete is generated and signaled from the selected source when used with the DMA channel.

Table 22-3. CRC Input Source

CRCSRC[5:0]	Name	Description
0x0	NOACT	No action
0x1	IO	I/O interface
0x2-0x1F		Reserved
0x20-0x3F	CHN	DMA channel n

- **Bits 7:4 – Reserved**
  These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written. These bits will always return zero when read.

- **Bits 3:2 – CRCPOLY[1:0]: CRC Polynomial Type**
  These bits select the CRC polynomial type, as shown in Table 22-4.

Table 22-15. Event Output Selection

EVOSEL[1:0]	Name	Description
0x0	DISABLE	Event generation disabled
0x1	BLOCK	Event strobe when block transfer complete
0x2		Reserved
0x3	BEAT	Event strobe when beat transfer complete

- **Bit 0 – VALID: Descriptor Valid**
  0: The descriptor is not valid.
  1: The descriptor is valid.
  Writing a zero to this bit in the Descriptor or Write-Back memory will suspend the DMA channel operation when fetching the corresponding descriptor.
  The bit is automatically cleared in the Write-Back memory section when channel is aborted, when an error is detected during the block transfer, or when the block transfer is completed.

If the Generic Clocks Request bit in the Control register (CTRL.GCLKREQ) is zero, the channel operates in SleepWalking mode and request the configured generic clock only when an event is to be propagated through the channel. If CTRL.GCLKREQ is one, the generic clock will always be on for the configured channel.

Resynchronized Path

The resynchronized path should be used when the event generator and the event channel do not share the same generic clock generator. When the resynchronized path is used, resynchronization of the event from the event generator is done in the channel.

When the resynchronized path is used, the channel is capable of generating interrupts. The channel status bits in the Channel Status register (CHSTATUS) are also updated and available for use.

If the Generic Clocks Request bit in the Control register (CTRL.GCLKREQ) is zero, the channel operates in SleepWalking mode and request the configured generic clock only when an event is to be propagated through the channel. If CTRL.GCLKREQ is one, the generic clock will always be on for the configured channel.

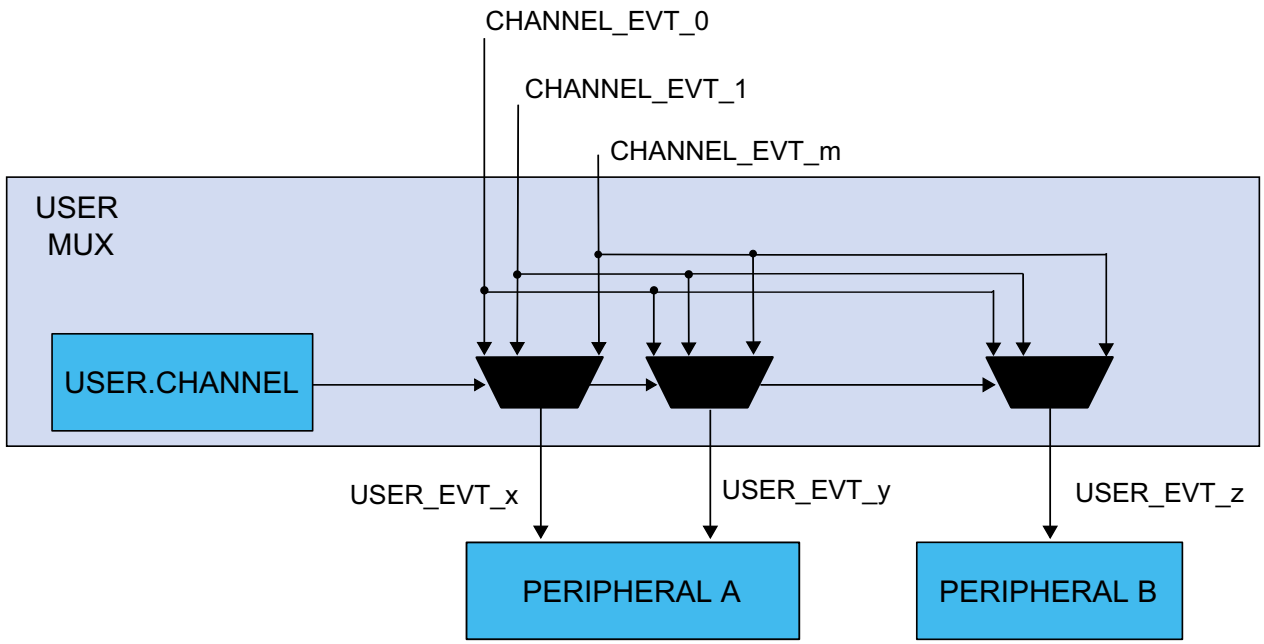
26.6.2.4 User Multiplexer Setup

Each user multiplexer is dedicated to one event user. A user multiplexer receives all event channel outputs and must be configured to select one of these channels. The user must always be configured before the channel is configured. A full list of selectable users can be found in the User Multiplexer register (USER) description. Refer to Table 26-6 for details.

To configure a user multiplexer, the USER register must be written in a single 16-bit write.

It is possible to read out the configuration of a user by first selecting the user by writing to USER.USER using an 8-bit write and then performing a read of the 16-bit USER register.

Figure 26-3. User MUX



26.6.2.5 Channel Setup

The channel to be used with an event user must be configured with an event generator. The path of the channel should be configured, and when using a synchronous path or resynchronized path, the edge selection should be configured. All these configurations are available in the Channel register (CHANNEL).

Table 26-4. Event Generator Selection

Value	Event Generator	Description
0x00	NONE	No event generator selected
0x01	RTC CMP0	Compare 0 (mode 0 and 1) or Alarm 0 (mode 2)
0x02	RTC CMP1	Compare 1
0x03	RTC OVF	Overflow
0x04	RTC PER0	Period 0
0x05	RTC PER1	Period 1
0x06	RTC PER2	Period 2
0x07	RTC PER3	Period 3
0x08	RTC PER4	Period 4
0x09	RTC PER5	Period 5
0x0A	RTC PER6	Period 6
0x0B	RTC PER7	Period 7
0x0C	EIC EXTINT0	External Interrupt 0
0x0D	EIC EXTINT1	External Interrupt 1
0x0E	EIC EXTINT2	External Interrupt 2
0x0F	EIC EXTINT3	External Interrupt 3
0x10	EIC EXTINT4	External Interrupt 4
0x11	EIC EXTINT5	External Interrupt 5
0x12	EIC EXTINT6	External Interrupt 6
0x13	EIC EXTINT7	External Interrupt 7
0x14	EIC EXTINT8	External Interrupt 8
0x15	EIC EXTINT9	External Interrupt 9
0x16	EIC EXTINT10	External Interrupt 10
0x17	EIC EXTINT11	External Interrupt 11
0x18	EIC EXTINT12	External Interrupt 12
0x19	EIC EXTINT13	External Interrupt 13
0x1A	EIC EXTINT14	External Interrupt 14
0x1B	EIC EXTINT15	External Interrupt 15
0x1C	Reserved	
0x1D	Reserved	
0x1E	DMAC CH0	Channel 0
0x1F	DMAC CH1	Channel 1

28.8.1 Control A

Name: CTRLA

Offset: 0x00

Reset: 0x00000000

Property: Enable-Protected, Write-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
		DORD	CPOL	CMODE	FORM[3:0]			
Access	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SAMPA[1:0]		RXPO[1:0]				TXPO[1:0]	
Access	R/W	R/W	R/W	R/W	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SAMPR[2:0]							IBON
Access	R/W	R/W	R/W	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W	R	R	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

- **Bit 31 – Reserved**
  This bit is unused and reserved for future use. For compatibility with future devices, always write this bit to zero when this register is written. This bit will always return zero when read.
- **Bit 30 – DORD: Data Order**
  This bit indicates the data order when a character is shifted out from the Data register.
  0: MSB is transmitted first.
  1: LSB is transmitted first.
  This bit is not synchronized.
- **Bit 29 – CPOL: Clock Polarity**
  This bit indicates the relationship between data output change and data input sampling in synchronous mode.
  This bit is not synchronized.

This bit is cleared by writing a one to the bit or by disabling the receiver.
This bit is set if the received character had a frame error, i.e., when the first stop bit is zero.
Writing a zero to this bit has no effect.
Writing a one to this bit will clear it.

- **Bit 0 – PERR: Parity Error**
  Reading this bit before reading the Data register will indicate the error status of the next character to be read.
  This bit is cleared by writing a one to the bit or by disabling the receiver.
  This bit is set if parity checking is enabled (CTRLA.FORM is 0x1 or 0x5) and a parity error is detected.
  Writing a zero to this bit has no effect.
  Writing a one to this bit will clear it.

30.6.6 Synchronization

Due to the asynchronicity between CLK_SERCOMx_APB and GCLK_SERCOMx_CORE, some registers must be synchronized when accessed. A register can require:

- Synchronization when written
- Synchronization when read
- Synchronization when written and read
- No synchronization

When executing an operation that requires synchronization, the corresponding Synchronization Busy bit in the Synchronization Busy register (SYNCBUSY) will be set immediately, and cleared when synchronization is complete.

If an operation that requires synchronization is executed while the corresponding SYNCBUSY bit is one, a peripheral bus error is generated.

The following bits need synchronization when written:

- Software Reset bit in the Control A register (CTRLA.SWRST). SYNCBUSY.SWRST is set to one while synchronization is in progress.
- Enable bit in the Control A register (CTRLA.ENABLE). SYNCBUSY.ENABLE is set to one while synchronization is in progress.
- Write to Bus State bits in the Status register (STATUS.BUSSTATE). SYNCBUSY.SYSOP is set to one while synchronization is in progress.
- Address bits in the Address register (ADDR.ADDR) when in master operation. SYNCBUSY.SYSOP is set to one while synchronization is in progress.
- Data (DATA) when in master operation. SYNCBUSY.SYSOP is set to one while synchronization is in progress.

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

Table 30-4. Register Summary – Master Mode

Offset	Name	Bit Pos								
0x00	CTRLA	7:0	RUNSTDBY			MODE[2:0]=101			ENABLE	SWRST
0x01		15:8								
0x02		23:16	SEXTTOEN	MEXTTOEN	SDAHOLD[1:0]					PINOUT
0x03		31:24		LOWTOUT	INACTOUT[1:0]		SCLSM		SPEED[1:0]	
0x04	CTRLB	7:0								
0x05		15:8							QCEN	SMEN
0x06		23:16						ACKACT	CMD[1:0]	
0x07		31:24								
0x08	Reserved									
0x09	Reserved									
0x0A	Reserved									
0x0B	Reserved									
0x0C	BAUD	7:0	BAUD[7:0]							
0x0D		15:8	BAUDLOW[7:0]							
0x0E		23:16	HSBAUD[7:0]							
0x0F		31:24	HSBAUDLOW[7:0]							
0x10	Reserved									
0x11	Reserved									
0x12	Reserved									
0x13	Reserved									
0x14	INTENCLR	7:0	ERROR						SB	MB

30.8.1.2 Control B

Name: CTRLB

Offset: 0x04

Reset: 0x00000000

Property: Write-Protected, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
						ACKACT	CMD[1:0]	
Access	R	R	R	R	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	AMODE[1:0]					AACKEN	GCMD	SMEN
Access	R/W	R/W	R	R	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

- **Bits 31:19 – Reserved**
  These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written. These bits will always return zero when read.

- **Bit 18 – ACKACT: Acknowledge Action**
  0: Send ACK
  1: Send NACK
  The Acknowledge Action (ACKACT) bit defines the slave's acknowledge behavior after an address or data byte is received from the master. The acknowledge action is executed when a command is written to the CMD bits. If smart mode is enabled (CTRLB.SMEN is one), the acknowledge action is performed when the DATA register is read.
  This bit is not enable-protected.

- **Bits 17:16 – CMD[1:0]: Command**
  Writing the Command bits (CMD) triggers the slave operation as defined in Table 30-7. The CMD bits are strobe bits, and always read as zero. The operation is dependent on the slave interrupt flags, INTFLAG.DRDY and INTFLAG.AMATCH, in addition to STATUS.DIR (See Table 30-7).

1: High-speed transfer enabled.

- **Bit 13 – LENEN: Transfer Length Enable**
  This bit enables automatic transfer length.
  0: Automatic transfer length disabled.
  1: Automatic transfer length enabled.

- **Bits 12:11 – Reserved**
  These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written. These bits will always return zero when read.

- **Bits 10:0 – ADDR[10:0]: Address**
  When ADDR is written, the consecutive operation will depend on the bus state:
  Unknown: INTFLAG.MB and STATUS.BUSERR are set, and the operation is terminated.
  Busy: The I^2C master will await further operation until the bus becomes idle.
  Idle: The I^2C master will issue a start condition followed by the address written in ADDR. If the address is acknowledged, SCL is forced and held low, and STATUS.CLKHOLD and INTFLAG.MB are set.
  Owner: A repeated start sequence will be performed. If the previous transaction was a read, the acknowledge action is sent before the repeated start bus condition is issued on the bus. Writing ADDR to issue a repeated start is performed while INTFLAG.MB or INTFLAG.SB is set.
  Regardless of winning or loosing arbitration, the entire address will be sent. If arbitration is lost, only ones are transmitted from the point of loosing arbitration and the rest of the address length.
  STATUS.BUSERR, STATUS.ARBLOST, INTFLAG.MB and INTFLAG.SB will be cleared when ADDR is written.
  The ADDR register can be read at any time without interfering with ongoing bus activity, as a read access does not trigger the master logic to perform any bus protocol related operations.
  The I^2C master control logic uses bit 0 of ADDR as the bus protocol's read/write flag (R/W); 0 for write and 1 for read.

- **Bits 31:24 – Reserved**
  These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written. These bits will always return zero when read.

- **Bits 23:16 – LEN[7:0]: Transaction Length**
  For DMA operation, this field represents the data length of the transaction from 0 to 255 bytes. The transaction length enable (ADDR.LENEN) must be written to 1 for automatic transaction length to be used. After ADDR.LEN bytes have been transmitted or received, a NACK (for master reads) and STOP are automatically generated.

- **Bit 15 – TENBITEN: Ten Bit Addressing Enable**
  This bit enables 10-bit addressing. This bit can be written simultaneously with ADDR to indicate a 10-bit or 7-bit address transmission.
  0: 10-bit addressing disabled.
  1: 10-bit addressing enabled.

- **Bit 14 – HS: High Speed**
  This bit enables High-speed mode for the current transfer from repeated START to STOP. This bit can be written simultaneously with ADDR for a high speed transfer.
  0: .High-speed transfer disabled.
  1: High-speed transfer enabled.

- **Bit 13 – LENEN: Transfer Length Enable**
  This bit enables automatic transfer length.
  0: Automatic transfer length disabled.
  1: Automatic transfer length enabled.

Table 31-6. Prescaler

Value	Name	Description
0x0	DIV1	Prescaler: GCLK_TC
0x1	DIV2	Prescaler: GCLK_TC/2
0x2	DIV4	Prescaler: GCLK_TC/4
0x3	DIV8	Prescaler: GCLK_TC/8
0x4	DIV16	Prescaler: GCLK_TC/16
0x5	DIV64	Prescaler: GCLK_TC/64
0x6	DIV256	Prescaler: GCLK_TC/256
0x7	DIV1024	Prescaler: GCLK_TC/1024

- **Bit 7 – Reserved**
  This bit is unused and reserved for future use. For compatibility with future devices, always write this bit to zero when this register is written. This bit will always return zero when read.
- **Bits 6:5 – WAVEGEN[1:0]: Waveform Generation Operation**
  These bits select the waveform generation operation. They affect the top value, as shown in “Waveform Output Operations” on page 582. It also controls whether frequency or PWM waveform generation should be used. How these modes differ can also be seen from “Waveform Output Operations” on page 582.
  These bits are not synchronized.

Table 31-7. Waveform Generation Operation

Value	Name	Operation	Top Value	Waveform Output on Match	Waveform Output on Wraparound
0x0	NFRQ	Normal frequency	PER(1)/Max	Toggle	No action
0x1	MFRQ	Match frequency	CC0	Toggle	No action
0x2	NPWM	Normal PWM	PER(1)/Max	Clear when counting up Set when counting down	Set when counting up Clear when counting down
0x3	MPWM	Match PWM	CC0	Clear when counting up Set when counting down	Set when counting up Clear when counting down

Note: 1. This depends on the TC mode. In 8-bit mode, the top value is the Period Value register (PER). In 16- and 32-bit mode it is the maximum value.

- **Bit 4 – Reserved**
  This bit is unused and reserved for future use. For compatibility with future devices, always write this bit to zero when this register is written. This bit will always return zero when read.
- **Bits 3:2 – MODE[1:0]: TC Mode**
  These bits select the TC mode, as shown in Table 31-8.
  These bits are not synchronized.

33.8.3 Average Control

Name: AVGCTRL

Offset: 0x02

Reset: 0x00

Property: Write-Protected

Bit	7	6	5	4	3	2	1	0
		ADJRES[2:0]			SAMPLENUM[3:0]			
Access	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

- **Bit 7 – Reserved**
  This bit is unused and reserved for future use. For compatibility with future devices, always write this bit to zero when this register is written. This bit will always return zero when read.

- **Bits 6:4 – ADJRES[2:0]: Adjusting Result / Division Coefficient**
  These bits define the division coefficient in 2^n steps.

- **Bits 3:0 – SAMPLENUM[3:0]: Number of Samples to be Collected**
  These bits define how many samples should be added together.The result will be available in the Result register (RESULT). Note: if the result width increases, CTRLB.RESSEL must be changed.

Table 33-8. Number of Samples to be Collected

SAMPLENUM[3:0]	Name	Description
0x0	1	1 sample
0x1	2	2 samples
0x2	4	4 samples
0x3	8	8 samples
0x4	16	16 samples
0x5	32	32 samples
0x6	64	64 samples
0x7	128	128 samples
0x8	256	256 samples
0x9	512	512 samples
0xA	1024	1024 samples
0xB-0xF		Reserved

34.9.11 Comparator Control n

The configuration of comparator n is protected while comparator n is enabled (COMPCTRLn.ENABLE=1). Changes to the other bits in COMPCTRLn can only occur when COMPCTRLn.ENABLE is zero.

Name: COMPCTRLn

Offset: 0x10+n*0x4 [n=0..1]

Reset: 0x00000000

Property: Write-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
						FLEN[2:0]		
Access	R	R	R	R	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
					HYST		OUT[1:0]	
Access	R	R	R	R	R/W	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SWAP		MUXPOS[1:0]			MUXNEG[2:0]		
Access	R/W	R	R/W	R/W	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		INTSEL[1:0]			SPEED[1:0]		SINGLE	ENABLE
Access	R	R/W	R/W	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

- **Bits 31:27 – Reserved**
  These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written. These bits will always return zero when read.

- **Bits 26:24 – FLEN[2:0]: Filter Length**
  These bits configure the filtering for comparator n. COMPCTRLn.FLEN can only be written while COMPCTRLn.ENABLE is zero.

  These bits are not synchronized.

Table 34-6. Filter Length

FLEN[2:0]	Name	Description
0x0	OFF	No filtering

35.8.3 Event Control

Name: EVCTRL

Offset: 0x2

Reset: 0x00

Property: Write-Protected

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

- **Bits 7:2 – Reserved**
  These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written. These bits will always return zero when read.

- **Bit 1 – EMPTYEO: Data Buffer Empty Event Output**
  This bit indicates whether or not the Data Buffer Empty event is enabled and will be generated when the Data Buffer register is empty.
  0: Data Buffer Empty event is disabled and will not be generated.
  1: Data Buffer Empty event is enabled and will be generated.

- **Bit 0 – STARTEI: Start Conversion Event Input**
  This bit indicates whether or not the Start Conversion event is enabled and data are loaded from the Data Buffer register to the Data register upon event reception.
  0: A new conversion will not be triggered on an incoming event.
  1: A new conversion will be triggered on an incoming event.

Table 37-9. Typical Peripheral Current Consumption

Peripheral	Conditions	Typ.	Unit
RTC	f_{GCLK_RTC} = 32kHz, 32bit counter mode	7.4	µA
WDT	f_{GCLK_WDT} = 32kHz, normal mode with EW	5.5	
AC	Both f_{GCLK} = 8MHz, Enable both COMP	31.3	
TCx(1)	f_{GCLK} = 8MHz, Enable + COUNTER in 8bit mode	50	
TCC2	f_{GCLK} = 8MHz, Enable + COUNTER	95.5	
TCC1	f_{GCLK} = 8MHz, Enable + COUNTER	167.5	
TCC0	f_{GCLK} = 8MHz, Enable + COUNTER	180.3	
SERCOMx.I2CM(2)	f_{GCLK} = 8MHz, Enable	69.7	
SERCOMx.I2CS	f_{GCLK} = 8MHz, Enable	29.2	
SERCOMx.SPI	f_{GCLK} = 8MHz, Enable	64.6	
SERCOMx.USART	f_{GCLK} = 8MHz, Enable	65.5	
DMAC(3)	RAM to RAM transfer	399.5	

Notes:
1. All TCs from 4 to 7 share the same power consumption values.
2. All SERCOMs from 0 to 5 share the same power consumption values.
3. The value includes the power consumption of the R/W access to the RAM.

Figure 37-14.CPU Utilization

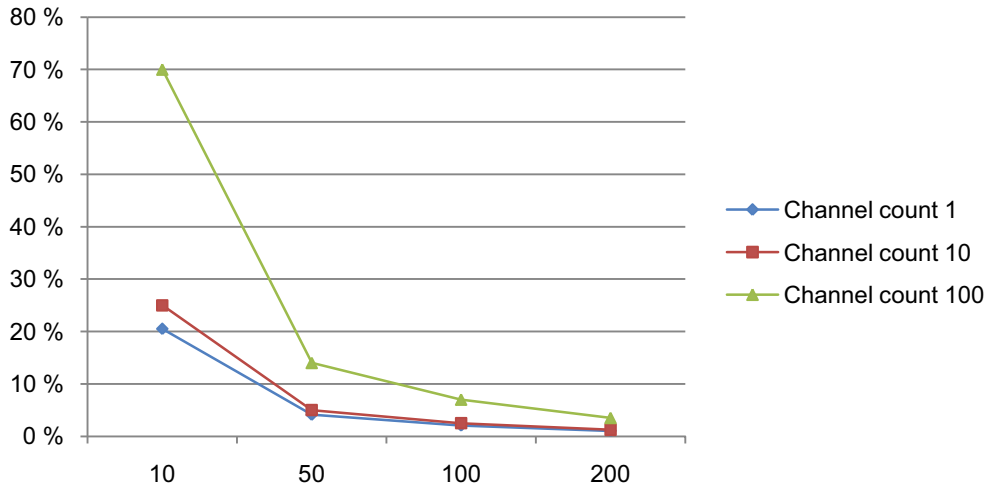


Table 38-8. Equivalent Internal Pin Capacitance

Symbol	Value	Description
C_{XIN32}	3.05pF	Equivalent internal pin capacitance
C_{XOUT32}	3.29pF	Equivalent internal pin capacitance

38.7 Programming and Debug Ports

For programming and/or debugging the SAM HA1 A the device should be connected using the Serial Wire Debug, SWD, interface. Currently the SWD interface is supported by several Atmel and third party programmers and debuggers, like the SAM-ICE, JTAGICE3 ATMEL-ICE or SAM HA1 A Xplained Pro (SAM HA1 A evaluation kit) Embedded Debugger.

Refer to the SAM-ICE, JTAGICE3 ATMEL-ICE or SAM HA1 A Xplained Pro user guides for details on debugging and programming connections and options. For connecting to any other programming or debugging tool, refer to that specific programmer or debugger's user guide.

The SAM HA1 A Xplained Pro evaluation board for the SAM HA1 A supports programming and debugging through the onboard embedded debugger so no external programmer or debugger is needed.

Note that a pull-up resistor on the SWCLK pin is critical for reliable operations. Refer to "Operation in Noisy Environment" on page 845. for more information.

Figure 38-10.SWCLK Circuit Connections

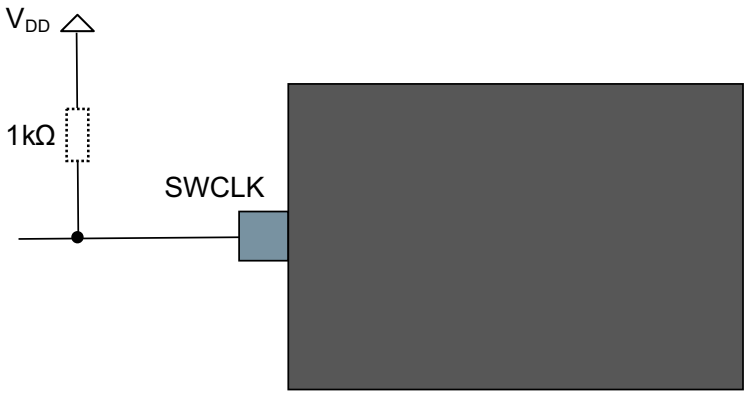


Table 38-9. SWCLK Circuit Connections

Pin Name	Description	Recommended Pin Connection
SWCLK	Serial wire clock pin	Pull-up resistor 1kΩ

38.7.3 20-pin IDC JTAG Connector

For debuggers and/or programmers that support the 20-pin IDC JTAG Connector, e.g. the SAM-ICE, the signals should be connected as shown in Figure 38-13 with details described in Table 38-12.

Figure 38-13.20-pin IDC JTAG Connector

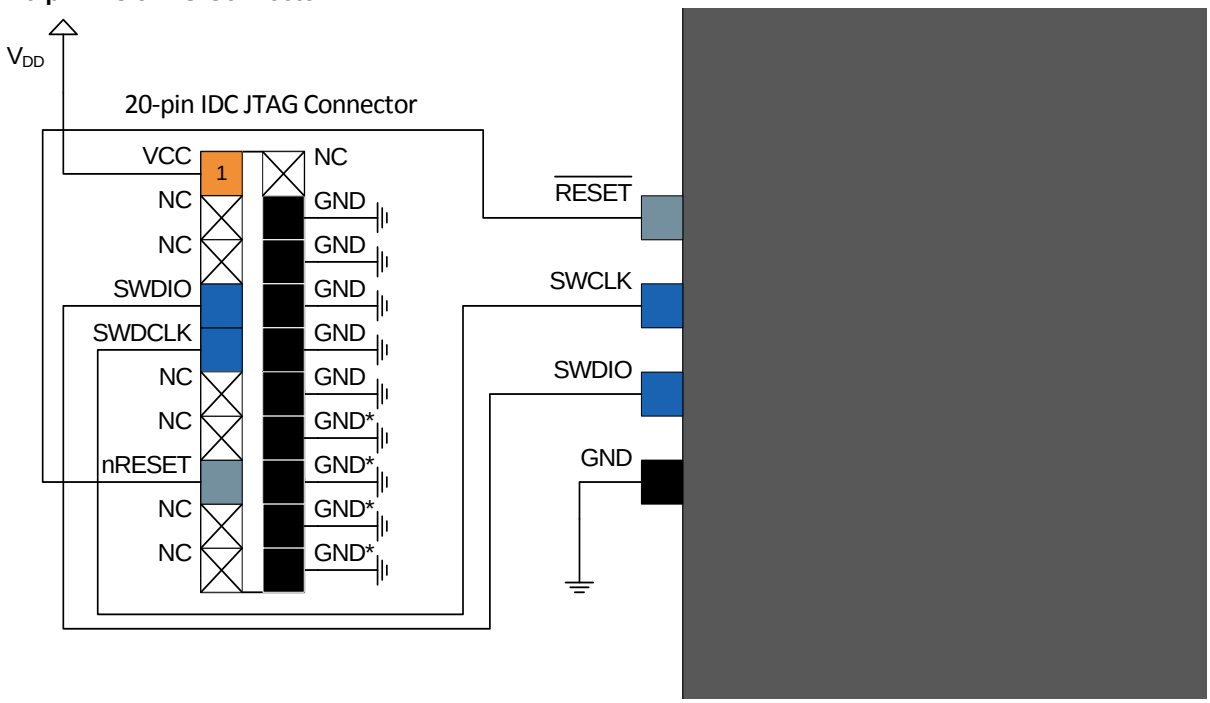


Table 38-12. 20-pin IDC JTAG Connector

Header Signal Name	Description
SWDCLK	Serial wire clock pin
SWDIO	Serial wire bidirectional data pin
nRESET	Target device reset pin, active low
VCC	Target voltage sense, should be connected to the device V_{DD}
GND	Ground
GND*	These pins are reserved for firmware extension purposes. They can be left open or connected to GND in normal debug environment. They are not essential for SWD in general.