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, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	52
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 20x12b; D/A 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsamd21j15b-af

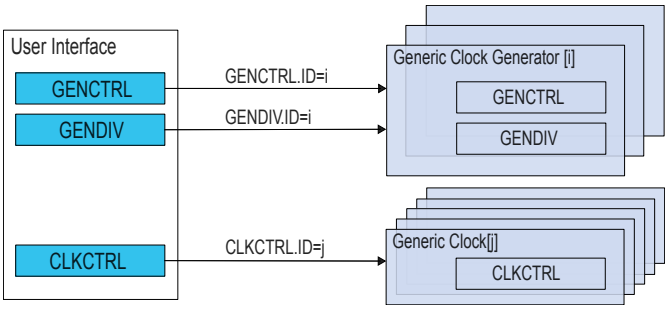
There is one exception concerning the GCLKGEN[0]. As it is used as GCLK_MAIN, it can not be locked. It is reset by any Reset and will start up in a known configuration. The software reset (CTRL.SWRST) can not unlock the registers.

15.6.4 Additional Features

15.6.4.1 Indirect Access

The Generic Clock Generator Control and Division registers (GENCTRL and GENDIV) and the Generic Clock Control register (CLKCTRL) are indirectly addressed as shown in the next figure.

Figure 15-5. GCLK Indirect Access



Writing these registers is done by setting the corresponding ID bit group. To read a register, the user must write the ID of the channel, i, in the corresponding register. The value of the register for the corresponding ID is available in the user interface by a read access.

For example, the sequence to read the GENCTRL register of generic clock generator i is:

1. Do an 8-bit write of the i value to GENCTRL.ID
2. Read the value of GENCTRL

15.6.4.2 Generic Clock Enable after Reset

The Generic Clock Controller must be able to provide a generic clock to some specific peripherals after a reset. That means that the configuration of the Generators and generic clocks after Reset is device-dependent.

Refer to *GENCTRL.ID* for details on GENCTRL reset.

Refer to *GENDIV.ID* for details on GENDIV reset.

Refer to *CLKCTRL.ID* for details on CLKCTRL reset.

Related Links

GENDIV

GENCTRL

CLKCTRL

15.6.5 Sleep Mode Operation

15.6.5.1 Sleep Walking

The GCLK module supports the Sleep Walking feature. If the system is in a sleep mode where the Generic Clocks are stopped, a peripheral that needs its clock in order to execute a process must request it from the Generic Clock Controller.

The Generic Clock Controller receives this request, determines which Generic Clock Generator is involved and which clock source needs to be awakened. It then wakes up the respective clock source, enables the Generator and generic clock stages successively, and delivers the clock to the peripheral.

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.

17.5.7 Analog Connections

When used, the 32.768kHz crystal must be connected between the XIN32 and XOUT32 pins, and the 0.4-32MHz crystal must be connected between the XIN and XOUT pins, along with any required load capacitors. For details on recommended oscillator characteristics and capacitor load, refer to the *Electrical Characteristics* for details.

Related Links

Electrical Characteristics

17.6 Functional Description

17.6.1 Principle of Operation

XOSC, XOSC32K, OSC32K, OSCULP32K, OSC8M, DFLL48M, FDPLL96M, BOD33, BOD12, and VREF are configured via SYSCTRL control registers. Through this interface, the sub-peripherals are enabled, disabled or have their calibration values updated.

The Power and Clocks Status register gathers different status signals coming from the sub-peripherals controlled by the SYSCTRL. The status signals can be used to generate system interrupts, and in some cases wake up the system from standby mode, provided the corresponding interrupt is enabled.

The oscillator must be enabled to run. The oscillator is enabled by writing a one to the ENABLE bit in the respective oscillator control register, and disabled by writing a zero to the oscillator control register. In idle mode, the default operation of the oscillator is to run only when requested by a peripheral. In standby mode, the default operation of the oscillator is to stop. This behavior can be changed by the user, see below for details.

The behavior of the oscillators in the different sleep modes is shown in the table below.

Table 17-1. Behavior of the Oscillators

Oscillator	Idle 0, 1, 2	Standby
XOSC	Run on request	Stop
XOSC32K	Run on request	Stop
OSC32K	Run on request	Stop
OSCULP32K	Run	Run
OSC8M	Run on request	Stop
DFLL48M	Run on request	Stop
FDPLL96M	Run on request	Stop

To force an oscillator to always run in idle mode, and not only when requested by a peripheral, the oscillator ONDEMAND bit must be written to zero. The default value of this bit is one, and thus the default operation in idle mode is to run only when requested by a peripheral.

When suspended, the Channel Suspend Interrupt flag in the Channel Interrupt Status and Clear register is set (CHINTFLAG.SUSP=1) and the optional suspend interrupt is generated.

By configuring the block action to suspend by writing Block Action bit group in the Block Transfer Control register (BTCTRL.BLOCKACT is 0x2 or 0x3), the DMA channel will be suspended after it has completed a block transfer. The DMA channel will be kept enabled and will be able to receive transfer triggers, but it will be removed from the arbitration scheme.

If an invalid transfer descriptor (BTCTRL.VALID=0) is fetched from SRAM, the DMA channel will be suspended, and the Channel Fetch Error bit in the Channel Status register(CHASTATUS.FERR) will be set.

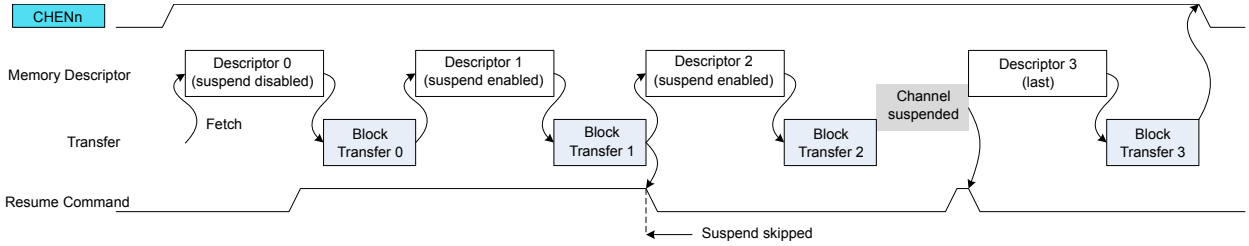
Note: Only enabled DMA channels can be suspended. If a channel is disabled when it is attempted to be suspended, the internal suspend command will be ignored.

For more details on transfer descriptors, refer to section Transfer Descriptors.

20.6.3.3 Channel Resume and Next Suspend Skip

A channel operation can be resumed by software by setting the Resume command in the Command bit field of the Channel Control B register (CHCTRLB.CMD). If the channel is already suspended, the channel operation resumes from where it previously stopped when the Resume command is detected. When the Resume command is issued before the channel is suspended, the next suspend action is skipped and the channel continues the normal operation.

Figure 20-10. Channel Suspend/Resume Operation



20.6.3.4 Event Input Actions

The event input actions are available only on the least significant DMA channels. For details on channels with event input support, refer to the in the Event system documentation.

Before using event input actions, the event controller must be configured first according to the following table, and the Channel Event Input Enable bit in the Channel Control B register (CHCTRLB.EVIE) must be written to '1'. Refer also to Events.

Table 20-1. Event Input Action

Action	CHCTRLB.EVACT	CHCTRLB.TRGSRC
None	NOACT	-
Normal Transfer	TRIG	DISABLE

Bit	7	6	5	4	3	2	1	0
	WRBADDR[7:0]							
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

Bits 31:0 – WRBADDR[31:0]: Write-Back Memory Base Address
These bits store the Write-Back memory base address. The value must be 128-bit aligned.

20.8.17 Channel ID

Name: CHID
Offset: 0x3F
Reset: 0x00
Property: -

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

Bits 3:0 – ID[3:0]: Channel ID
These bits define the channel number that will be affected by the channel registers (CH*). Before reading or writing a channel register, the channel ID bit group must be written first.

20.8.18 Channel Control A

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Name: CHCTRLA
Offset: 0x40
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

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

Bit 1 – ENABLE: Channel Enable
Writing a '0' to this bit during an ongoing transfer, the bit will not be cleared until the internal data transfer buffer is empty and the DMA transfer is aborted. The internal data transfer buffer will be empty once the ongoing burst transfer is completed.

Writing a '1' to this bit will enable the DMA channel.

This bit is not enable-protected.

Value	Description
0	DMA channel is disabled.
1	DMA channel is enabled.

Bit 0 – SWRST: Channel Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets the channel registers to their initial state. The bit can be set when the channel is disabled (ENABLE=0). Writing a '1' to this bit will be ignored as long as ENABLE=1. This bit is automatically cleared when the reset is completed.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

20.8.19 Channel Control B

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Name: CHCTRLB
Offset: 0x44
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
							CMD[1:0]	
Access							R/W	R/W
Reset							0	0

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

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

Bit	7	6	5	4	3	2	1	0
		LVL[1:0]		EVOE	EVIE	EVACT[2:0]		
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 25:24 – CMD[1:0]: Software Command
These bits define the software commands. Refer to Channel Suspend and Channel Resume and Next Suspend Skip.

These bits are not enable-protected.

CMD[1:0]	Name	Description
0x0	NOACT	No action
0x1	SUSPEND	Channel suspend operation

24.4 Signal Description

Not applicable.

24.5 Product Dependencies

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

24.5.1 I/O Lines

Not applicable.

24.5.2 Power Management

The EVSYS can be used to wake up the CPU from all sleep modes, even if the clock used by the EVSYS channel and the EVSYS bus clock are disabled. Refer to the *PM – Power Manager* for details on the different sleep modes.

In all sleep modes, although the clock for the EVSYS is stopped, the device still can wake up the EVSYS clock. Some event generators can generate an event when their clocks are stopped.

Related Links

PM – Power Manager

24.5.3 Clocks

The EVSYS bus clock (CLK_EVSYS_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_EVSYS_APB can be found in *Peripheral Clock Masking*.

Each EVSYS channel has a dedicated generic clock (GCLK_EVSYS_CHANNEL_n). These are used for event detection and propagation for each channel. These clocks must be configured and enabled in the generic clock controller before using the EVSYS. Refer to *GCLK - Generic Clock Controller* for details.

Related Links

Peripheral Clock Masking

GCLK - Generic Clock Controller

24.5.4 DMA

Not applicable.

24.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the EVSYS interrupts requires the interrupt controller to be configured first. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller

24.5.6 Events

Not applicable.

24.5.7 Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging.

Value	Event Generator	Description
0x20	DMAC CH2	Channel 2
0x21	DMAC CH3	Channel 3
0x22	TCC0 OVF	Overflow
0x23	TCC0 TRG	Trig
0x24	TCC0 CNT	Counter
0x25	TCC0_MCX0	Match/Capture 0
0x26	TCC0_MCX1	Match/Capture 1
0x27	TCC0_MCX2	Match/Capture 2
0x28	TCC0_MCX3	Match/Capture 3
0x29	TCC1 OVF	Overflow
0x2A	TCC1 TRG	Trig
0x2B	TCC1 CNT	Counter
0x2C	TCC1_MCX0	Match/Capture 0
0x2D	TCC1_MCX1	Match/Capture 1
0x2E	TCC2 OVF	Overflow
0x2F	TCC2 TRG	Trig
0x30	TCC2 CNT	Counter
0x31	TCC2_MCX0	Match/Capture 0
0x32	TCC2_MCX1	Match/Capture 1
0x33	TC0 OVF	Overflow/Underflow
0x34	TC0 MC0	Match/Capture 0
0x35	TC0 MC1	Match/Capture 1
0x36	TC1 OVF	Overflow/Underflow
0x37	TC1 MC0	Match/Capture 0
0x38	TC1 MC1	Match/Capture 1
0x39	TC2 OVF	Overflow/Underflow
0x3A	TC2 MC0	Match/Capture 0
0x3B	TC2 MC1	Match/Capture 1
0x3C	TC3 OVF	Overflow/Underflow
0x3D	TC3 MC0	Match/Capture 0
0x3E	TC3 MC1	Match/Capture 1
0x3F	TC4 OVF	Overflow/Underflow

Table 25-1. 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

25.6.2.2 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.

25.6.2.3 Clock Generation – Baud-Rate Generator

The baud-rate generator, as shown in Figure 25-3, 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.

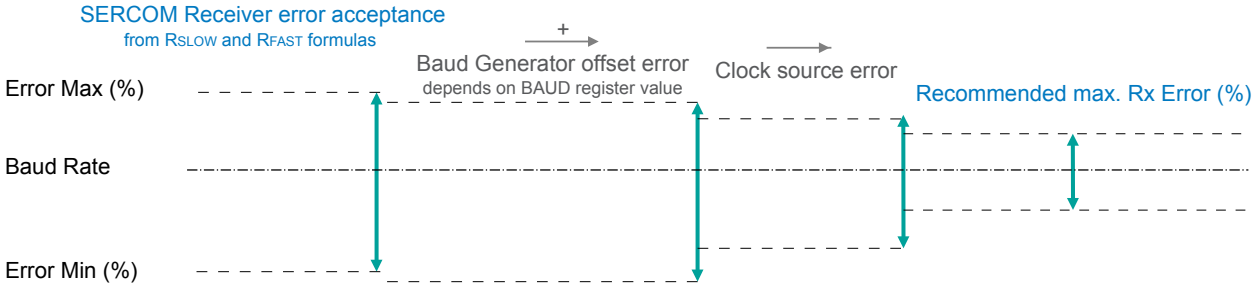
The following equations calculate the ratio of the incoming data rate and internal receiver baud rate:

$$R_{SLOW} = \frac{(D + 1)S}{S - 1 + D \cdot S + S_F} \quad , \quad R_{FAST} = \frac{(D + 2)S}{(D + 1)S + S_M}$$

- R_{SLOW} is the ratio of the slowest incoming data rate that can be accepted in relation to the receiver baud rate
- R_{FAST} is the ratio of the fastest incoming data rate that can be accepted in relation to the receiver baud rate
- D is the sum of character size and parity size (D = 5 to 10 bits)
- S is the number of samples per bit (S = 16, 8 or 3)
- S_F is the first sample number used for majority voting (S_F = 7, 3, or 2) when CTRLA.SAMPA=0.
- S_M is the middle sample number used for majority voting (S_M = 8, 4, or 2) when CTRLA.SAMPA=0.

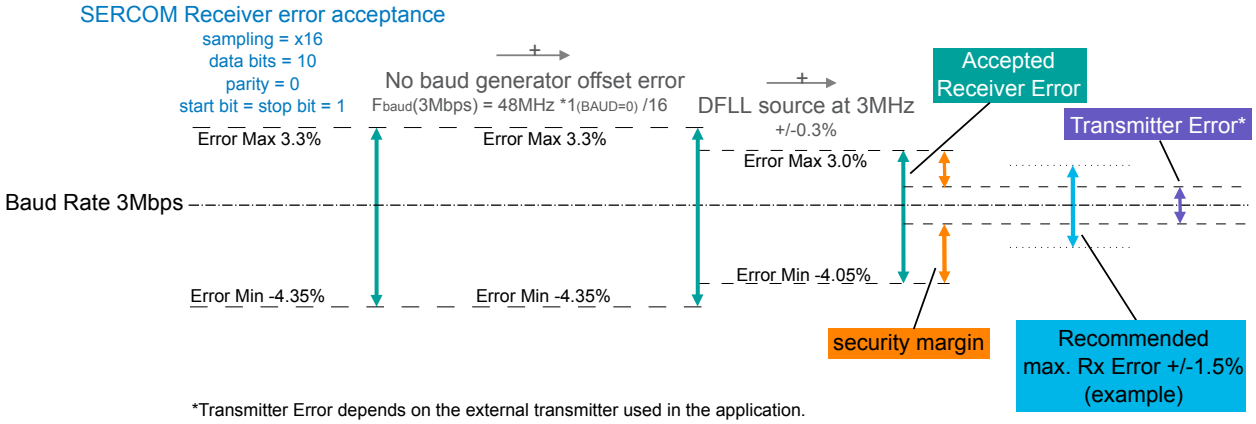
The recommended maximum Rx Error assumes that the receiver and transmitter equally divide the maximum total error. Its connection to the SERCOM Receiver error acceptance is depicted in this figure:

Figure 26-5. USART Rx Error Calculation



The recommendation values in the table above accommodate errors of the clock source and the baud generator. The following figure gives an example for a baud rate of 3Mbps:

Figure 26-6. USART Rx Error Calculation Example



*Transmitter Error depends on the external transmitter used in the application.
It is advised that it is within the Recommended max. Rx Error (+/-1.5% in this example).
Larger Transmitter Errors are acceptable but must lie within the Accepted Receiver Error.

Related Links

Clock Generation – Baud-Rate Generator

Asynchronous Arithmetic Mode BAUD Value Selection

26.6.3 Additional Features

26.6.3.1 Parity

Even or odd parity can be selected for error checking by writing 0x1 to the Frame Format bit field in the Control A register (CTRLA.FORM).

Startup Timing The minimum time between SDA transition and SCL rising edge is 6 APB cycles when the DATA register is written in smart mode. If a greater startup time is required due to long rise times, the time between DATA write and IF clear must be controlled by software.
Note: When timing is controlled by user, the Smart Mode cannot be enabled.

Related Links
Electrical Characteristics

Master Clock Generation (High-Speed Mode)
For I²C *Hs* transfers, there is no SCL synchronization. Instead, the SCL frequency is determined by the GCLK_SERCOMx_CORE frequency (f_{GCLK}) and the High-Speed Baud setting in the Baud register (BAUD.HSBAUD). When BAUD.HSBAUDLOW=0, the HSBAUD value will determine both SCL high and SCL low. In this case the following formula determines the SCL frequency.

$$f_{SCL} = \frac{f_{GCLK}}{2 + 2 \cdot HS\,BAUD}$$

When HSBAUDLOW is non-zero, the following formula determines the SCL frequency.

$$f_{SCL} = \frac{f_{GCLK}}{2 + HS\,BAUD + HSBAUDLOW}$$

Note: The I²C standard *Hs* (High-speed) requires a nominal high to low SCL ratio of 1:2, and HSBAUD should be set accordingly. At a minimum, BAUD.HSBAUD and/or BAUD.HSBAUDLOW must be non-zero.

Transmitting Address Packets
The I²C master starts a bus transaction by writing the I²C slave address to ADDR.ADDR and the direction bit, as described in Principle of Operation. If the bus is busy, the I²C master will wait until the bus becomes idle before continuing the operation. When the bus is idle, the I²C master will issue a start condition on the bus. The I²C master will then transmit an address packet using the address written to ADDR.ADDR. After the address packet has been transmitted by the I²C master, one of four cases will arise according to arbitration and transfer direction.

Case 1: Arbitration lost or bus error during address packet transmission

If arbitration was lost during transmission of the address packet, the Master on Bus bit in the Interrupt Flag Status and Clear register (INTFLAG.MB) and the Arbitration Lost bit in the Status register (STATUS.ARBLOST) are both set. Serial data output to SDA is disabled, and the SCL is released, which disables clock stretching. In effect the I²C master is no longer allowed to execute any operation on the bus until the bus is idle again. A bus error will behave similarly to the arbitration lost condition. In this case, the MB interrupt flag and Master Bus Error bit in the Status register (STATUS.BUSERR) are both set in addition to STATUS.ARBLOST.

The Master Received Not Acknowledge bit in the Status register (STATUS.RXNACK) will always contain the last successfully received acknowledge or not acknowledge indication.

In this case, software will typically inform the application code of the condition and then clear the interrupt flag before exiting the interrupt routine. No other flags have to be cleared at this moment, because all flags will be cleared automatically the next time the ADDR.ADDR register is written.

Case 2: Address packet transmit complete – No ACK received

If there is no I²C slave device responding to the address packet, then the INTFLAG.MB interrupt flag and STATUS.RXNACK will be set. The clock hold is active at this point, preventing further activity on the bus.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

28.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
		LOWTOUT			SCLSM		SPEED[1:0]	
Access		R/W			R/W		R/W	R/W
Reset		0			0		0	0

Bit	23	22	21	20	19	18	17	16
	SEXTTOEN		SDAHOLD[1:0]					PINOUT
Access	R/W		R/W	R/W				R/W
Reset	0		0	0				0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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

Bit 30 – LOWTOUT: SCL Low Time-Out
This bit enables the SCL low time-out. If SCL is held low for 25ms-35ms, the slave will release its clock hold, if enabled, and reset the internal state machine. Any interrupt flags set at the time of time-out will remain set.

Value	Description
0	Time-out disabled.
1	Time-out enabled.

Bit 27 – SCLSM: SCL Clock Stretch Mode
This bit controls when SCL will be stretched for software interaction.

This bit is not synchronized.

Value	Description
0	SCL stretch according to Figure 28-9
1	SCL stretch only after ACK bit according to Figure 28-10

Bits 25:24 – SPEED[1:0]: Transfer Speed
These bits define bus speed.

These bits are not synchronized.

Value	Name	Description
0x0	MASK	The slave responds to the address written in ADDR.ADDR masked by the value in ADDR.ADDRMASK. See *SERCOM – Serial Communication Interface* for additional information.
0x1	2_ADDRS	The slave responds to the two unique addresses in ADDR.ADDR and ADDR.ADDRMASK.
0x2	RANGE	The slave responds to the range of addresses between and including ADDR.ADDR and ADDR.ADDRMASK. ADDR.ADDR is the upper limit.
0x3	-	Reserved.

Bit 10 – AACKEN: Automatic Acknowledge Enable
This bit enables the address to be automatically acknowledged if there is an address match.

This bit is not write-synchronized.

Value	Description
0	Automatic acknowledge is disabled.
1	Automatic acknowledge is enabled.

Bit 9 – GCMD: PMBus Group Command
This bit enables PMBus group command support. When enabled, the Stop Recived interrupt flag (INTFLAG.PREC) will be set when a STOP condition is detected if the slave has been addressed since the last STOP condition on the bus.

This bit is not write-synchronized.

Value	Description
0	Group command is disabled.
1	Group command is enabled.

Bit 8 – SMEN: Smart Mode Enable
When smart mode is enabled, data is acknowledged automatically when DATA.DATA is read.

This bit is not write-synchronized.

Value	Description
0	Smart mode is disabled.
1	Smart mode is enabled.

28.8.3 Interrupt Enable Clear

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

Name: INTENCLR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ERROR					DRDY	AMATCH	PREC
Access	R/W					R/W	R/W	R/W
Reset	0					0	0	0

Value	Description
0	The Recoverable Fault B interrupt is disabled.
1	The Recoverable Fault B interrupt is enabled.

Bit 12 – FAULTA: Recoverable Fault A Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Recoverable Fault A Interrupt Disable/Enable bit, which enables the Recoverable Fault A interrupt.

Value	Description
0	The Recoverable Fault A interrupt is disabled.
1	The Recoverable Fault A interrupt is enabled.

Bit 11 – DFS: Non-Recoverable Debug Fault Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Debug Fault State Interrupt Disable/Enable bit, which enables the Debug Fault State interrupt.

Value	Description
0	The Debug Fault State interrupt is disabled.
1	The Debug Fault State interrupt is enabled.

Bit 3 – ERR: Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Error Interrupt Disable/Enable bit, which enables the Compare interrupt.

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

Bit 2 – CNT: Counter Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Retrigger Interrupt Disable/Enable bit, which enables the Counter interrupt.

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

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

Writing a '1' to this bit will set the Retrigger Interrupt Disable/Enable bit, which enables the Retrigger interrupt.

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

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

Bit 5 – EORSM: End Of Resume Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the End Of Resume interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The End Of Resume interrupt is disabled.
1	The End Of Resume interrupt is enabled.

Bit 4 – WAKEUP: Wake-Up Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the Wake Up interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Wake Up interrupt is disabled.
1	The Wake Up interrupt is enabled.

Bit 3 – EORST: End of Reset Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the End of Reset interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The End of Reset interrupt is disabled.
1	The End of Reset interrupt is enabled.

Bit 2 – SOF: Start-of-Frame Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the Start-of-Frame interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Start-of-Frame interrupt is disabled.
1	The Start-of-Frame interrupt is enabled.

Bit 0 – SUSPEND: Suspend Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the Suspend interrupt Enable bit and enable the corresponding interrupt request.

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

32.8.2.7 Device Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x01C
Reset: 0x0000

Property: Read-Synchronized

Bit	15	14	13	12	11	10	9	8
	RESULT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

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

Bits 15:0 – RESULT[15:0]: Result Conversion Value
These bits will hold up to a 16-bit ADC result, depending on the configuration.

In single conversion mode without averaging, the ADC conversion will produce a 12-bit result, which can be left- or right-shifted, depending on the setting of CTRLB.LEFTADJ.

If the result is left-adjusted (CTRLB.LEFTADJ), the high byte of the result will be in bit position [15:8], while the remaining 4 bits of the result will be placed in bit locations [7:4]. This can be used only if an 8-bit result is required; i.e., one can read only the high byte of the entire 16-bit register.

If the result is not left-adjusted (CTRLB.LEFTADJ) and no oversampling is used, the result will be available in bit locations [11:0], and the result is then 12 bits long.

If oversampling is used, the result will be located in bit locations [15:0], depending on the settings of the Average Control register (AVGCTRL).

33.8.15 Window Monitor Lower Threshold

Name: WINLT
Offset: 0x1C
Reset: 0x0000
Property: Write-Protected, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	WINLT[15:8]							
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

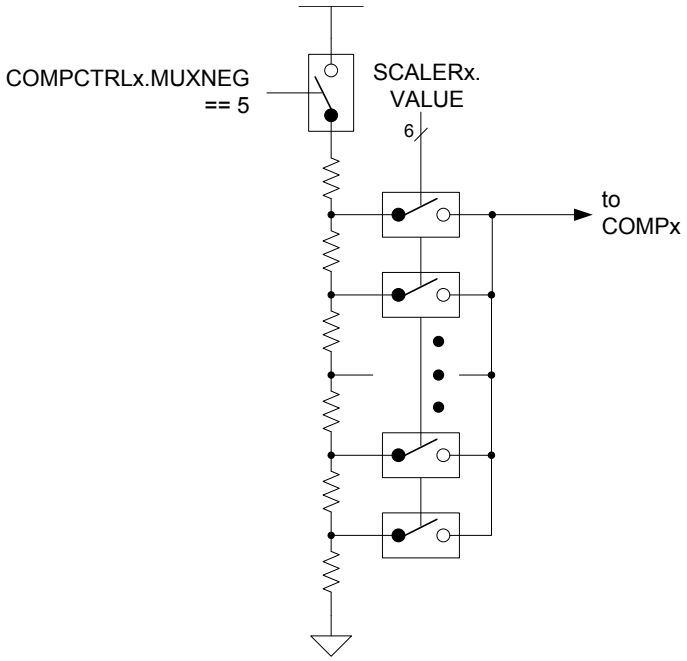
Bit	7	6	5	4	3	2	1	0
	WINLT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – WINLT[15:0]: Window Lower Threshold
If the window monitor is enabled, these bits define the lower threshold value.

33.8.16 Window Monitor Upper Threshold

Name: WINUT

Figure 34-5. V_{DDANA} Scaler



34.6.7 Input Hysteresis

Application software can selectively enable/disable hysteresis for the comparison. Applying hysteresis will help prevent constant toggling of the output, which can be caused by noise when the input signals are close to each other.

Hysteresis is enabled for each comparator individually by the Hysteresis Enable bit in the Comparator x Control register (COMPCTRLx.HYSTEN). Hysteresis is available only in continuous mode (COMPCTRLx.SINGLE=0).

34.6.8 Propagation Delay vs. Power Consumption

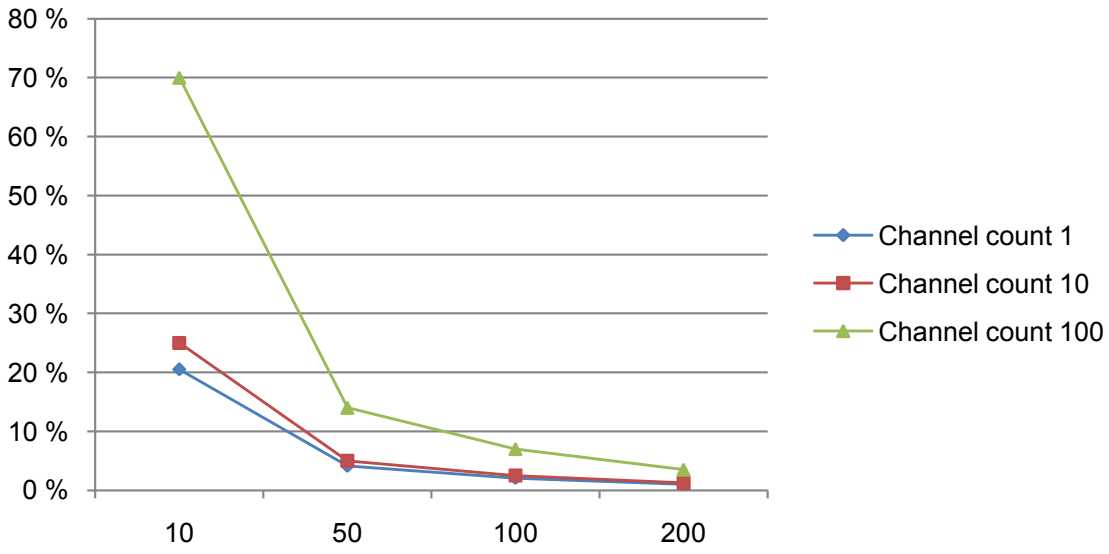
It is possible to trade off comparison speed for power efficiency to get the shortest possible propagation delay or the lowest power consumption. The speed setting is configured for each comparator individually by the Speed bit group in the Comparator x Control register (COMPCTRLx.SPEED). The Speed bits select the amount of bias current provided to the comparator, and as such will also affect the start-up time.

34.6.9 Filtering

The output of the comparators can be filtered digitally to reduce noise. The filtering is determined by the Filter Length bits in the Comparator Control x register (COMPCTRLx.FLEN), and is independent for each comparator. Filtering is selectable from none, 3-bit majority (N=3) or 5-bit majority (N=5) functions. Any change in the comparator output is considered valid only if N/2+1 out of the last N samples agree. The filter sampling rate is the GCLK_AC frequency.

Note that filtering creates an additional delay of N-1 sampling cycles from when a comparison is started until the comparator output is validated. For continuous mode, the first valid output will occur when the required number of filter samples is taken. Subsequent outputs will be generated every cycle based on the current sample plus the previous N-1 samples, as shown in Figure 34-6. For single-shot mode, the comparison completes after the Nth filter sample, as shown in Figure 34-7.

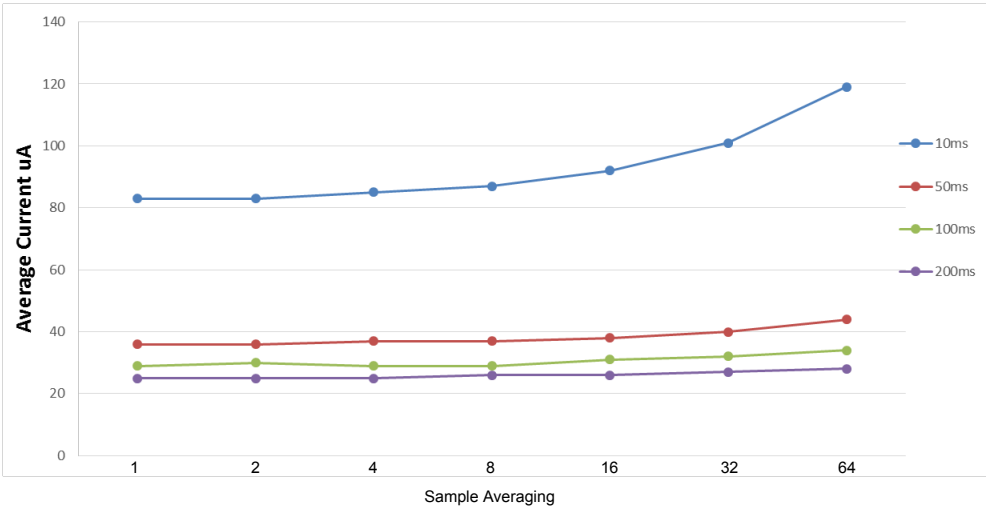
Figure 37-13. CPU Utilization



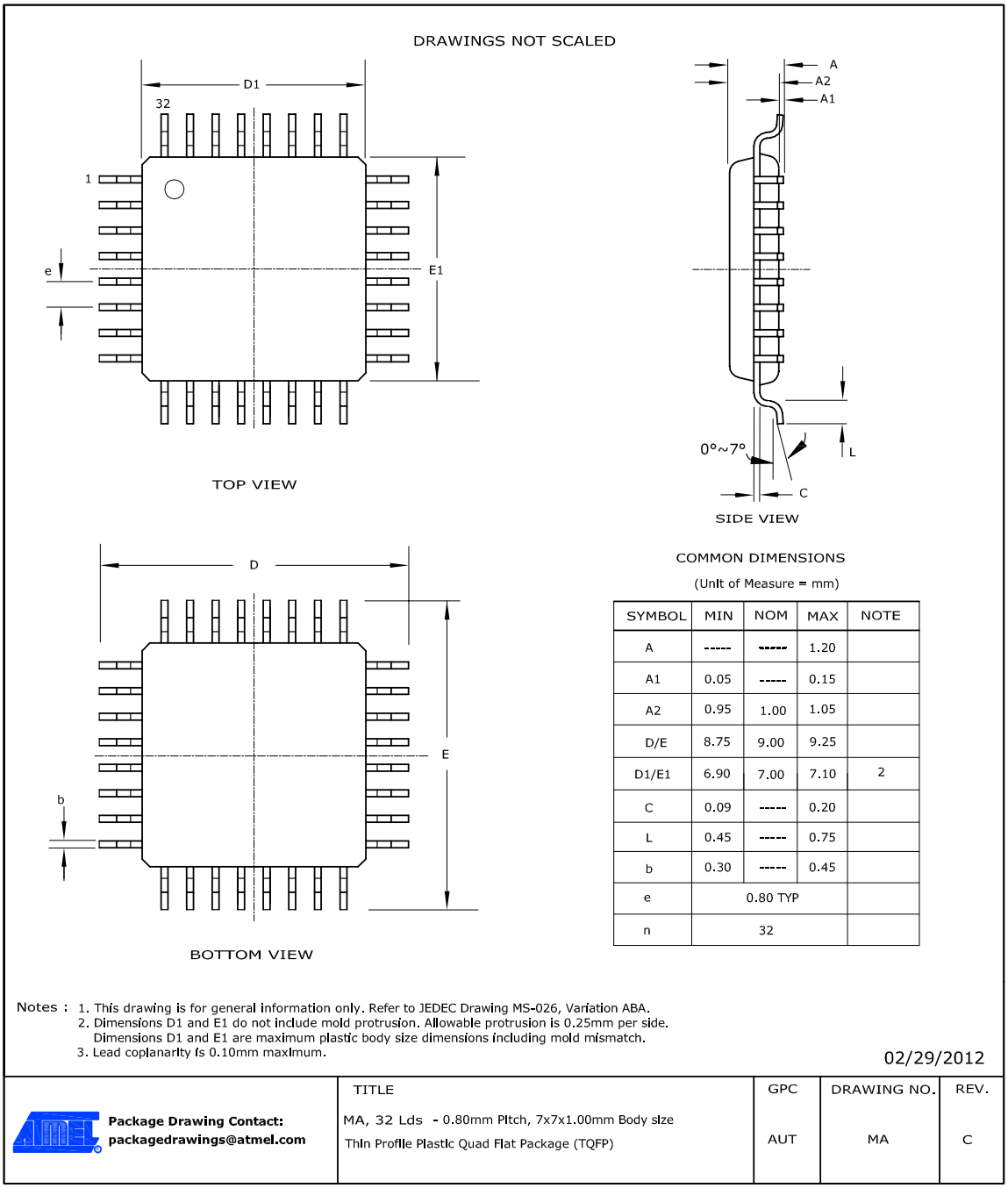
37.13.2 Device Variant B and C

V_{CC} = 3.3C and f_{CPU} = 48MHz for the following PTC measurements.

Figure 37-14. 1 Sensor / PTC_GCLK = 4MHz / FREQ_MODE_NONE



38.2.7 32 pin TQFP



COMMON DIMENSIONS

(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	-----	-----	1.20	
A1	0.05	-----	0.15	
A2	0.95	1.00	1.05	
D/E	8.75	9.00	9.25	
D1/E1	6.90	7.00	7.10	2
C	0.09	-----	0.20	
L	0.45	-----	0.75	
b	0.30	-----	0.45	
e	0.80 TYP			
n	32			

Notes : 1. This drawing is for general information only. Refer to JEDEC Drawing MS-026, Variation ABA.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10mm maximum.

02/29/2012

Package Drawing Contact: packagedrawings@atmel.com	TITLE	GPC	DRAWING NO.	REV.
	MA, 32 Lds - 0.80mm Pitch, 7x7x1.00mm Body size Thin Profile Plastic Quad Flat Package (TQFP)	AUT	MA	C

Table 38-20. Device and Package Maximum Weight

100	mg

Table 38-21. Package Charateristics

Moisture Sensitivity Level	MSL3

Figure 39-13. 20-pin IDC JTAG Connector

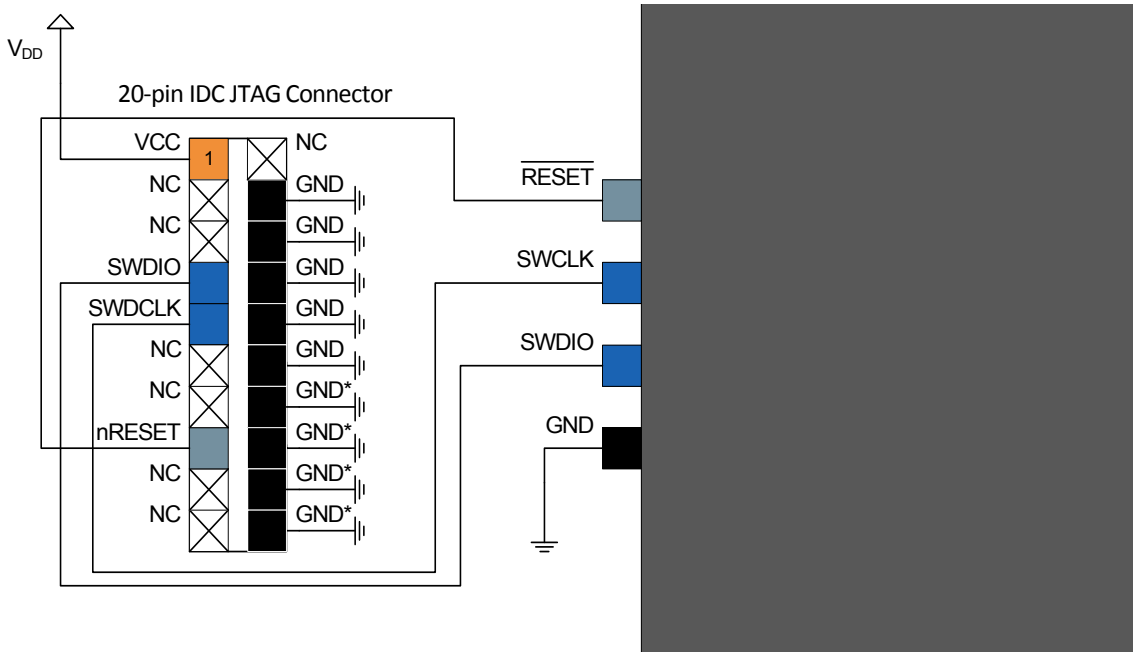


Table 39-12. 20-pin IDC JTAG Connector

Header Signal Name	Description
SWDCLK	Serial wire clock pin
SWDIO	Serial wire bidirectional data pin
$\overline{\text{RESET}}$	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.

39.8 USB Interface

The USB interface consists of a differential data pair (D+/D-) and a power supply (VBUS, GND). Refer to the Electrical Characteristics for operating voltages which will allow USB operation.

Table 39-13. USB Interface Checklist

Signal Name	Recommended Pin Connection	Description
D+	• The impedance of the pair should be matched on the PCB to minimize reflections. • USB differential tracks should be routed with the same characteristics (length, width, number of vias, etc.) • Signals should be routed as parallel as possible, with a minimum number of angles and vias	USB full speed / low speed positive data upstream pin
D-		USB full speed / low speed negative data upstream pin

Symbol	Description	Max.	Units
$f_{GCLK\_DPLL\_32K}$	FDPLL96M 32k Reference clock frequency	32	KHz
f_{GCLK_WDT}	WDT input clock frequency	48	MHz
f_{GCLK_RTC}	RTC input clock frequency	48	MHz
f_{GCLK_EIC}	EIC input clock frequency	48	MHz
f_{GCLK_USB}	USB input clock frequency	48	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_0}$	EVSYS channel 0 input clock frequency	48	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_1}$	EVSYS channel 1 input clock frequency	48	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_2}$	EVSYS channel 2 input clock frequency	48	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_3}$	EVSYS channel 3 input clock frequency	48	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_4}$	EVSYS channel 4 input clock frequency	48	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_5}$	EVSYS channel 5 input clock frequency	48	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_6}$	EVSYS channel 6 input clock frequency	48	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_7}$	EVSYS channel 7 input clock frequency	48	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_8}$	EVSYS channel 8 input clock frequency	48	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_9}$	EVSYS channel 9 input clock frequency	48	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_10}$	EVSYS channel 10 input clock frequency	48	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_11}$	EVSYS channel 11 input clock frequency	48	MHz
$f_{GCLK\_SERCOMx\_SLOW}$	Common SERCOM slow input clock frequency	48	MHz
$f_{GCLK\_SERCOM0\_CORE}$	SERCOM0 input clock frequency	48	MHz
$f_{GCLK\_SERCOM1\_CORE}$	SERCOM1 input clock frequency	48	MHz
$f_{GCLK\_SERCOM2\_CORE}$	SERCOM2 input clock frequency	48	MHz
$f_{GCLK\_SERCOM3\_CORE}$	SERCOM3 input clock frequency	48	MHz
$f_{GCLK\_SERCOM4\_CORE}$	SERCOM4 input clock frequency	48	MHz
$f_{GCLK\_SERCOM5\_CORE}$	SERCOM5 input clock frequency	48	MHz
$f_{GCLK\_TCC0,\ GCLK\_TCC1}$	TCC0, TCC1 input clock frequency	80	MHz
$f_{GCLK\_TCC2,\ GCLK\_TC3}$	TCC2, TC3 input clock frequency	80	MHz
$f_{GCLK\_TC4,\ GCLK\_TC5}$	TC4, TC5 input clock frequency	48	MHz
$f_{GCLK\_TC6,\ GCLK\_TC7}$	TC6, TC7 input clock frequency	48	MHz
f_{GCLK_ADC}	ADC input clock frequency	48	MHz
$f_{GCLK\_AC\_DIG}$	AC digital input clock frequency	48	MHz
$f_{GCLK\_AC\_ANA}$	AC analog input clock frequency	64	KHz
f_{GCLK_DAC}	DAC input clock frequency	350	KHz