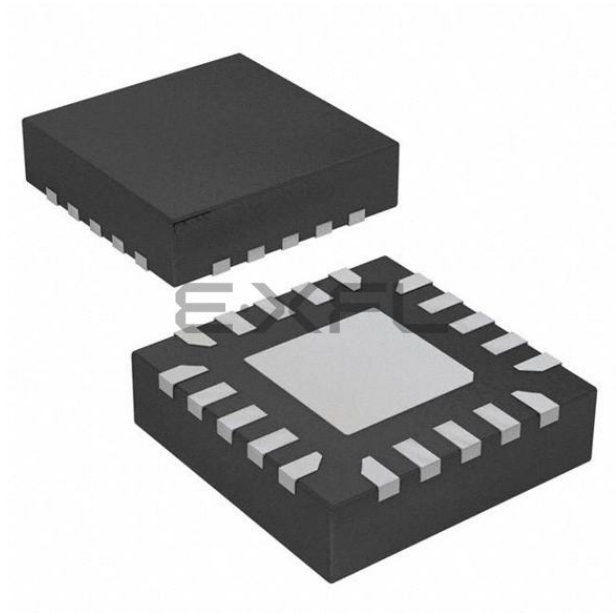


Welcome to E-XFL.COM

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	AVR
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	18
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 20x10b; D/A 3x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	20-VFQFN Exposed Pad
Supplier Device Package	20-VQFN (3x3)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/attiny1616-mnr

5. I/O Multiplexing and Considerations

5.1 Multiplexed Signals

Table 5-1. PORT Function Multiplexing

VQFN 20-pin	SOIC 20-pin	Pin Name (1,2)	Other/Special	ADC0	ADC1	PTC(3)	AC0	AC1	AC2	DAC0	USART0	SPI0	TWI0	TCA0	TCBn	TCD0	CCL
19	16	PA0	$\overline{\text{RESET}}$ UPDI	AIN0													LUT0-IN0
20	17	PA1		AIN1							TXD	MOSI	SDA				LUT0-IN1
1	18	PA2	EVOUT0	AIN2							RxD	MISO	SCL				LUT0-IN2
2	19	PA3	EXTCLK	AIN3							XCK	SCK		WO3	TCB1 WO		
3	20	GND															
4	1	VDD															
5	2	PA4		AIN4	AIN0	X0/Y0					XDIR	$\overline{\text{SS}}$		WO4		WOA	LUT0-OUT
6	3	PA5	VREFA	AIN5	AIN1	X1/Y1	OUT	AINN0						WO5	TCB0 WO	WOB	
7	4	PA6		AIN6	AIN2	X2/Y2	AINN0	AINP1	AINP0	OUT							
8	5	PA7		AIN7	AIN3	X3/Y3	AINP0	AINP0	AINN0								LUT1-OUT
9	6	PB5	CLKOUT	AIN8			AINP1		AINP2					WO2			
10	7	PB4		AIN9			AINN1	AINP3						WO1			LUT0-OUT
11	8	PB3	TOSC1					OUT			RxD			WO0			
12	9	PB2	TOSC2, EVOUT1						OUT		TxD			WO2			
13	10	PB1		AIN10		X4/Y4	AINP2				XCK		SDA	WO1			
14	11	PB0		AIN11		X5/Y5		AINP2	AINP1		XDIR		SCL	WO0			
15	12	PC0			AIN6							SCK			TCB0 WO	WOC	
16	13	PC1			AIN7							MISO				WOD	LUT1-OUT
17	14	PC2	EVOUT2		AIN8							MOSI					
18	15	PC3			AIN9							$\overline{\text{SS}}$		WO3			LUT1-IN0

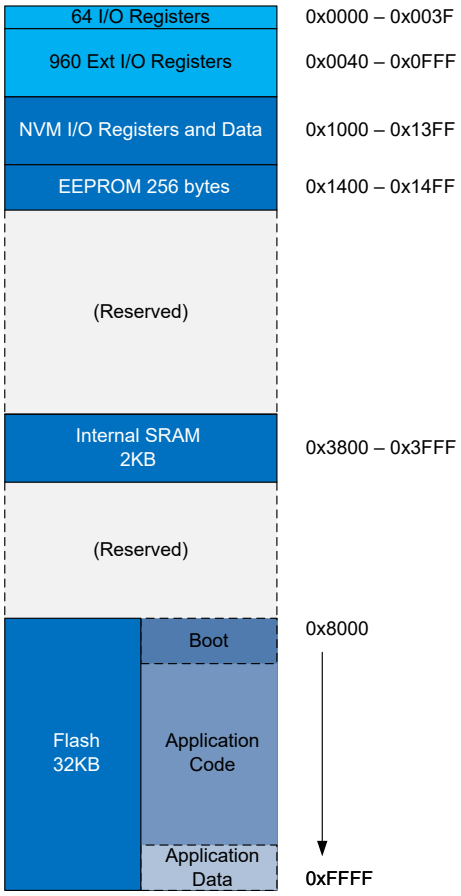
Note:

1. Pin names are of type P*xn*, with *x* being the PORT instance (A, B) and *n* the pin number. Notation for signals is PORT*x*_PIN*n*. All pins can be used as event input.
2. All pins can be used for external interrupt, where pins P*x*2 and P*x*6 of each port have full asynchronous detection.
3. Every PTC line can be configured as X- or Y-line.

Tip: Signals on alternative pin locations are in `typewriter` font.

6.2 Memory Map

Figure 6-1. Memory Map ATtiny3216



6.10.2.5 OSC16 Error at 5V

Name: OSC16ERR5V
Offset: 0x23
Reset: [Oscillator frequency error value]
Property: -

Bit	7	6	5	4	3	2	1	0
	OSC16ERR5V[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bits 7:0 – OSC16ERR5V[7:0] OSC16 Error at 5V
These registers contain the signed oscillator frequency error value when running at internal 16 MHz at 5V, as measured during production.

10.5.1 Main Clock Control A

Name: MCLKCTRLA
Offset: 0x00
Reset: 0x00
Property: Configuration Change Protection

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

Bit 7 – CLKOUT System Clock Out
When this bit is written to '1', the system clock is output to CLKOUT pin.

When the device is in a Sleep mode, there is no clock output unless a peripheral is using the system clock.

Bits 1:0 – CLKSEL[1:0] Clock Select
This bit field selects the source for the Main Clock (CLK_MAIN).

Value	Name	Description
0x0	OSC20M	16/20 MHz internal oscillator
0x1	OSCULP32K	32 KHz internal ultra low-power oscillator
0x2	XOSC32K	32.768 kHz external crystal oscillator
0x3	EXTCLK	External clock

15.2 Register Summary - PORTMUX

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0			LUT1	LUT0		EVOUT2	EVOUT1	EVOUT0
0x01	CTRLB	7:0						SPI0		USART0
0x02	CTRLC	7:0			TCA05	TCA04	TCA03	TCA02	TCA01	TCA00
0x03	CTRLD	7:0							TCB1	TCB0

15.3 Register Description

18. VREF - Voltage Reference

18.1 Features

- Programmable Voltage Reference Sources:
  - One for each ADC peripheral
  - One for each AC and DAC peripheral
- Each Reference Source Supports Five Different Voltages:
  - 0.55V
  - 1.1V
  - 1.5V
  - 2.5V
  - 4.3V

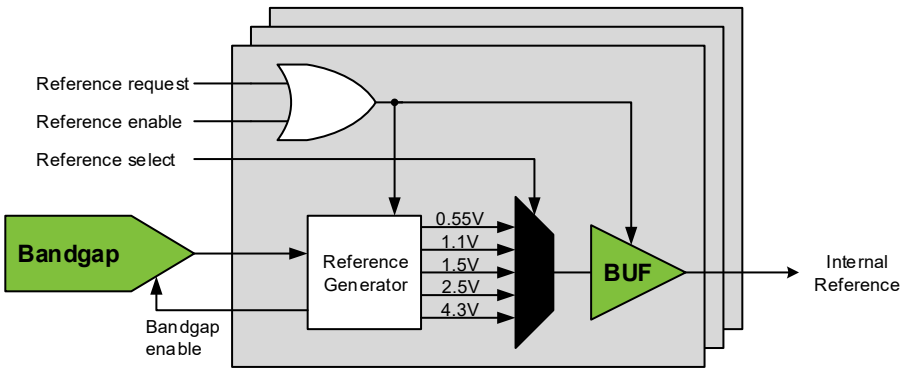
18.2 Overview

The Voltage Reference (VREF) peripheral provides control registers for the voltage reference sources used by several peripherals. The user can select the reference voltages for the ADC0 by writing to the ADC0 Reference Select bit field (ADC0REFSEL) in the Control A register (VREF.CTRLA), and for both AC0 and DAC0 by writing to the DAC0 and AC0 Reference Select bit field DAC0REFSEL in VREF.CTRLA.

A voltage reference source is enabled automatically when requested by a peripheral. The user can enable the reference voltage sources (and thus, override the automatic disabling of unused sources) by writing to the respective Force Enable bit (ADC0REFEN, DAC0REFEN) in the Control B register (VREF.CTRLB). This may be desirable to decrease start-up time, at the cost of increased power consumption.

18.2.1 Block Diagram

Figure 18-1. VREF Block Diagram



18.3 Functional Description

21.5.5 Interrupt Flags

Name: INTFLAGS
Offset: 0x06
Reset: 0x00
Property: -

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

Bit 0 – CAPT Interrupt Flag
This bit is set when an interrupt occurs. The interrupt conditions are dependent on the Counter Mode (CNTMODE) in TCBn.CTRLB.

This bit is cleared by writing a '1' to it or when the Capture register is read in Capture mode.

Counter Mode	Interrupt Flag Behavior
Periodic Interrupt mode	Set when the counter reaches TOP
Timeout Check mode	Set when the counter reaches TOP
Input Capture on Event mode	Set when an event occurs and the Capture register is loaded, cleared when Capture is read
Input Capture Frequency Measurement mode	Set on an edge when the Capture register is loaded and count initialized, cleared when Capture is read
Input Capture Pulse-Width Measurement mode	Set on an edge when the Capture register is loaded, the previous edge initialized the count, cleared when Capture is read
Input Capture Frequency and Pulse-Width Measurement mode	Set on second (positive or negative) edge when the counter is stopped, cleared when Capture is read
Single-Shot mode	Set when the counter reaches TOP
8-Bit PWM mode	Set when the counter reaches CCMPL

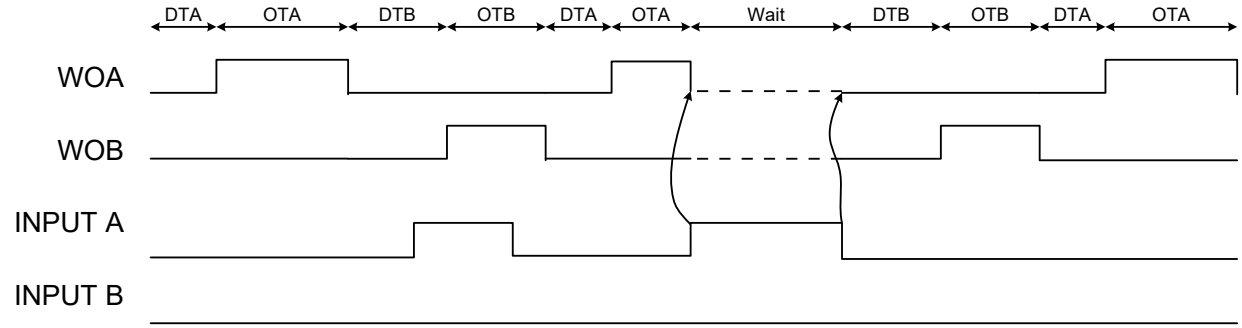
INPUTMODE	One Ramp Mode	Two Ramp Mode	Four Ramp Mode	Dual Slope Mode
0x7	Valid	Valid	Valid	Valid
0x8	Valid	Valid	Valid	Do not use
0x9	Valid	Valid	Valid	Do not use
0xA	Valid	Valid	Valid	Do not use

In the following sections the different Input modes are presented in detail.

Input Mode 1: Stop Output, Jump to Opposite Compare Cycle, and Wait
An input Event in Input mode 1 will stop the output signal, jump to the opposite dead time, and wait until the input event goes low before the TCD counter continues.

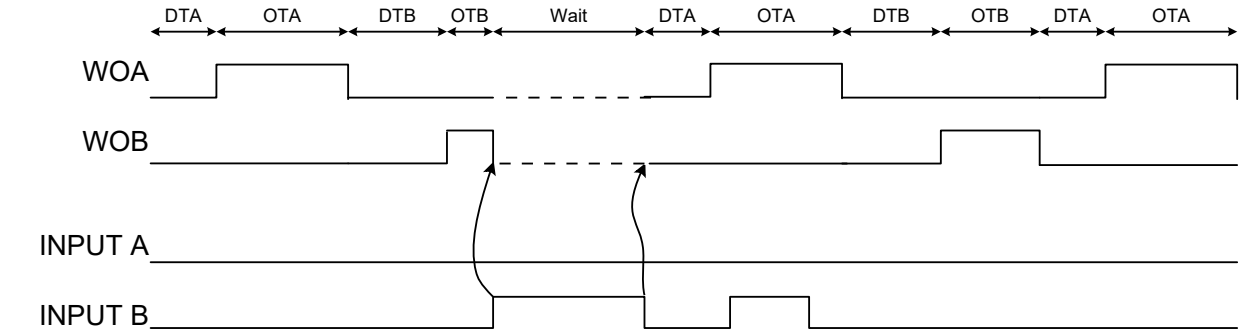
If Input mode 1 is used on input A, an event will only have an effect if the TCD is in Dead-time A or On-time A, and it will only affect the output WOA. When the event is done, the TCD counter starts at Dead-time B.

Figure 22-10. Input Mode 1 on Input A



If Input mode 1 is used on input B, an event will only have an effect if the TCD is in Dead-time B or On-time B, and it will only affect the output WOB. When the event is done, the TCD counter starts at Dead-time A.

Figure 22-11. Input Mode 1 on Input B



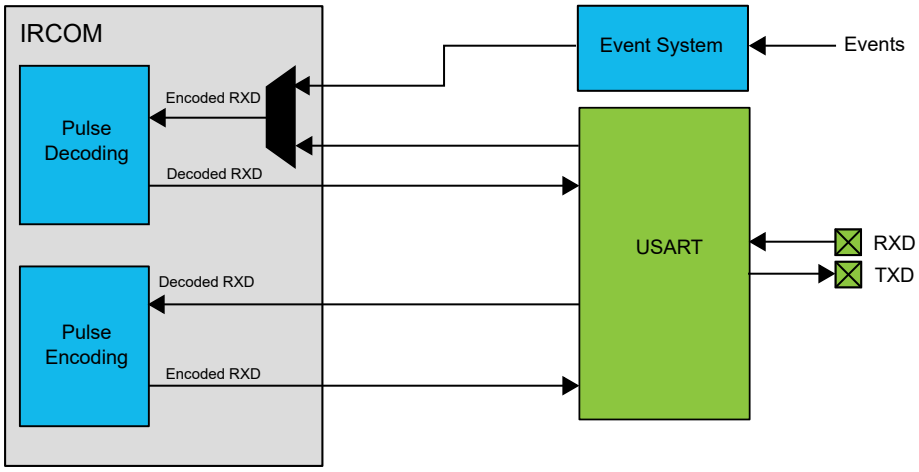
Input Mode 2: Stop Output, Execute Opposite Compare Cycle, and Wait
An input Event in Input mode 2 will stop the output signal, execute to the opposite dead time and on time, then wait until the input event goes low before the TCD counter continues. If the input is done before the opposite dead time and on time have finished, there will be no waiting, but the opposite dead time and on time will continue.

If Input mode 2 is used on input A, an event will only have an effect if the TCD is in Dead-time A or On-time A, and it will only affect the output WOA.

For the reception, a fixed programmable minimum high-level pulse-width for the pulse to be decoded as a logical ‘0’ is used. Shorter pulses will then be discarded, and the bit will be decoded to logical ‘1’ as if no pulse was received.

24.3.2.11.2 Block Diagram

Figure 24-13. Block Diagram



24.3.2.11.3 IRCOM and Event System

The Event System can be used as the receiver input. This enables the IRCOM or USART input from the I/O pins or sources other than the corresponding RX pin. If the Event System input is enabled, input from the USART's RX pin is automatically disabled.

Related Links

14. EVSYS - Event System

24.3.3 Events

The USART can accept the following input events:

- IREI - IrDA Event Input

The event is enabled by writing a '1' to the IrDA Event Input bit (IREI) in the Event Control register (USART.EVCTRL).

Related Links

14. EVSYS - Event System
24.5.12 EVCTRL

24.3.4 Interrupts

Table 24-10. Available Interrupt Vectors and Sources

Offset	Name	Vector Description	Conditions
0x00	RXC	Receive Complete Interrupt	• There are unread data in the receive buffer (RXCIE) • Receive of Start-of-Frame detected (RXSIE)

Bit 0 – DATA[8] Receiver Data Register
When USART receiver is set to LINAUTO mode, this bit indicates if the received data is within the response space of a LIN frame. If the received data is the protected identifier field, this bit will be read as '0'. Otherwise, the bit will be read as '1'. For Receiver mode other than LINAUTO mode, DATA[8] holds the ninth data bit in the received character when operating with serial frames with nine data bits.

26.2.3.3 Interrupts

26.2.3.1 Clocks

This peripheral requires the system clock (CLK_PER). The relationship between CLK_PER and the TWI bus clock (SCL) is explained in the TWI.MBAUD register.

Related Links

10. CLKCTRL - Clock Controller
26.5.6 MBAUD

26.2.3.2 I/O Lines and Connections

Using the I/O lines of the peripheral requires configuration of the I/O pins.

26.2.3.3 Interrupts

Using the interrupts of this peripheral requires the interrupt controller to be configured first.

26.2.3.4 Events

Not applicable.

26.2.3.5 Debug Operation

When run-time debugging, this peripheral will continue normal operation. Halting the CPU in Debugging mode will halt normal operation of the peripheral.

This peripheral can be forced to operate with halted CPU by writing a '1' to the Debug Run bit (DBGRUN) in the Debug Control register of the peripheral (*peripheral*.DBGCTRL).

When the CPU is halted in Debug mode and DBGRUN=1, reading/writing the DATA register will neither trigger a bus operation nor cause transmit and clear flags.

If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during halted debugging.

Related Links

33. UPDI - Unified Program and Debug Interface

26.3 Functional Description

26.3.1 Initialization

Before enabling the master or the slave unit, ensure that the correct settings for SDASETUP, SDAHOLD, and, if used, Fast-mode plus (FMPEN) are stored in TWI.CTRLA. If alternate pins are to be used for the slave, this must be specified in the TWIn.DUALCTRL register as well. Note that for dual mode the master enables the primary SCL/SDA pins, while the ENABLE bit in TWIn.DUALCTRL enables the secondary pins.

Master Operation

It is recommended to write to the Master Baud Rate register (TWIn.BAUD) before enabling the TWI master since TIMEOUT is dependent on the baud rate setting. To start the TWI master, write a ‘1’ to the ENABLE bit and configure an appropriate TIMEOUT if using the TWI in an SMBus environment. The ENABLE and TIMEOUT bits are all located in the Master Control A register (TWIn.MCTRLA). If no TIMEOUT value is set, which is the case for I²C operation, the bus state must be manually set to IDLE by writing 0x1 to BUSSTATE in TWIn.MSTATUS at a safe point in time. Note that unlike the SMBus specification, the I²C specification does not specify when it is safe to assume that the bus is idle in a

3. Writing to the TWIn.MDATA register.
4. Reading the TWIn.DATA register while the ACKACT control bits in TWIn.MCTRLB are set to either send ACK or NACK.
5. Writing a valid command to the TWIn.MCTRLB register.

Bit 4 – RXACK Received Acknowledge
This bit is read-only and contains the most recently received Acknowledge bit from the slave. When read as '0', the most recent acknowledge bit from the slave was ACK. When read as '1', the most recent acknowledge bit was NACK.

Bit 3 – ARBLOST Arbitration Lost
If read as '1' this bit indicates that the master has lost arbitration while transmitting a high data or NACK bit, or while issuing a Start or repeated Start condition (S/Sr) on the bus.

Writing a '1' to it will clear the ARBLOST flag. However, normal use of the TWI does not require the flag to be cleared by this method. However, as for the CLKHOLD flag, clearing the ARBLOST flag is not required during normal use of the TWI.

Clearing the ARBLOST bit will follow the same software interaction as the CLKHOLD flag.

Given the condition where the bus ownership is lost to another master, the software must either abort operation or resend the data packet. Either way, the next required software interaction is in both cases to write to the TWIn.MADDR register. A write access to the TWIn.MADDR register will then clear the ARBLOST flag.

Bit 2 – BUSERR Bus Error
The BUSERR flag indicates that an illegal bus condition has occurred. An illegal bus condition is detected if a protocol violating Start (S), repeated Start (Sr), or Stop (P) is detected on the TWI bus lines. A Start condition directly followed by a Stop condition is one example of protocol violation.

Writing a '1' to this bit will clear the BUSERR. However, normal use of the TWI does not require the BUSERR to be cleared by this method.

A robust TWI driver software design will treat the bus error flag similarly to the ARBLOST flag, assuming the bus ownership is lost when the bus error flag is set. As for the ARBLOST flag, the next software operation of writing the TWIn.MADDR register will consequently clear the BUSERR flag. For bus error to be detected, the bus state logic must be enabled and the system frequency must be 4x the SCL frequency.

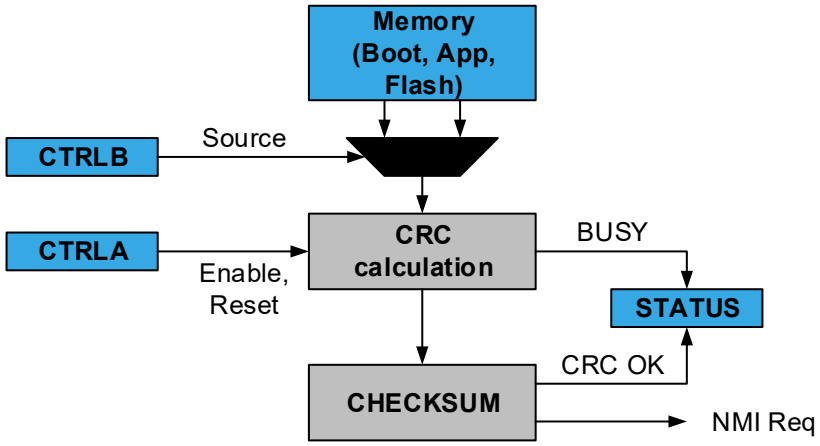
Bits 1:0 – BUSSTATE[1:0] Bus State
These bits indicate the current TWI bus state as defined in the table below. After a System Reset or re-enabling, the TWI master bus state will be unknown. The change of bus state is dependent on the bus activity.

Writing 0x1 to the BUSSTATE bits forces the bus state logic into its Idle state. However, the bus state logic cannot be forced into any other state. When the master is disabled, the bus state is 'unknown'.

Value	Name	Description
0x0	UNKNOWN	Unknown bus state
0x1	IDLE	Bus is idle
0x2	OWNER	This TWI controls the bus
0x3	BUSY	The bus is busy

27.2.1 Block Diagram

Figure 27-2. Cyclic Redundancy Check Block Diagram



27.2.2 System Dependencies

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

Table 27-1. System Product Dependencies

Dependency	Applicable	Peripheral
Clocks	Yes	CLKCTRL
I/O Lines and Connections	No	-
Interrupts	Yes	CPUINT
Events	No	-
Debug	Yes	UPDI

Related Links

11.2.2.1 Clocks
27.2.2.3 Interrupts

27.2.2.1 Clocks

This peripheral depends on the peripheral clock.

Related Links

10. CLKCTRL - Clock Controller

27.2.2.2 I/O Lines and Connections

Not applicable.

27.2.2.3 Interrupts

Using the interrupts of this peripheral requires the interrupt controller to be configured first.

Related Links

13. CPUINT - CPU Interrupt Controller
8.7.3 SREG
27.3.3 Interrupts

28.5.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: -

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

Bit 6 – RUNSTDBY Run in Standby
This bit indicates if the peripheral clock (CLK_PER) is kept running in Standby Sleep mode. The setting is ignored for configurations where the CLK_PER is not required.

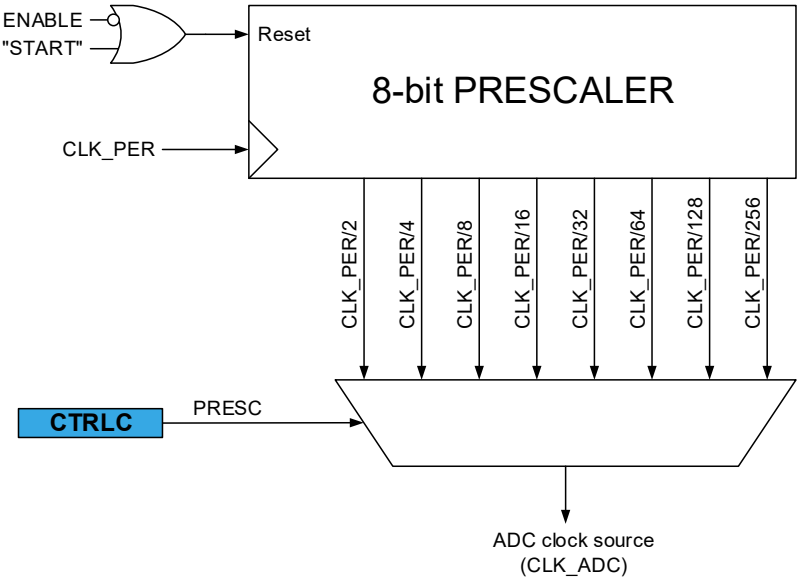
Value	Description
0	System clock is not required in Standby Sleep mode
1	System clock is required in Standby Sleep mode

Bit 0 – ENABLE Enable

Value	Description
0	The peripheral is disabled
1	The peripheral is enabled

30.3.2.2 Clock Generation

Figure 30-6. ADC Prescaler



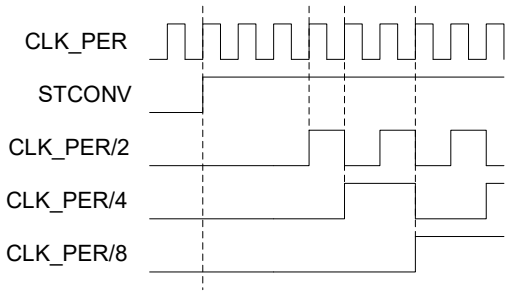
The ADC requires an input clock frequency between 50 kHz and 1.5 MHz for maximum resolution. If a lower resolution than 10 bits is selected, the input clock frequency to the ADC can be higher than 1.5 MHz to get a higher sample rate.

The ADC module contains a prescaler which generates the ADC clock (CLK_ADC) from any CPU clock (CLK_PER) above 100 kHz. The prescaling is selected by writing to the Prescaler bits (PRESC) in the Control C register (ADCn.CTRLC). The prescaler starts counting from the moment the ADC is switched on by writing a '1' to the ENABLE bit in ADCn.CTRLA. The prescaler keeps running as long as the ENABLE bit is '1'. The prescaler counter is reset to zero when the ENABLE bit is '0'.

When initiating a conversion by writing a '1' to the Start Conversion bit (STCONV) in the Command register (ADCn.COMMAND) or from an event, the conversion starts at the following rising edge of the CLK_ADC clock cycle. The prescaler is kept reset as long as there is no ongoing conversion. This assures a fixed delay from the trigger to the actual start of a conversion in CLK_PER cycles as:

$$\text{StartDelay} = \frac{\text{PRESC}_{\text{factor}}}{2} + 2$$

Figure 30-7. Start Conversion and Clock Generation



30.3.2.3 Conversion Timing

A normal conversion takes 13 CLK_ADC cycles. The actual sample-and-hold takes place two CLK_ADC cycles after the start of a conversion. Start of conversion is initiated by writing a '1' to the STCONV bit in ADC.COMMAND. When a conversion is complete, the result is available in the Result register (ADC.RES), and the Result Ready interrupt flag is set (RESRDY in ADC.INTFLAG). The interrupt flag will

30.5.11 Interrupt Flags

Name: INTFLAGS
Offset: 0x0B
Reset: 0x00
Property: -

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

Bit 1 – WCOMP Window Comparator Interrupt Flag
This window comparator flag is set when the measurement is complete and if the result matches the selected Window Comparator mode defined by WINCM (ADCn.CTRLE). The comparison is done at the end of the conversion. The flag is cleared by either writing a '1' to the bit position or by reading the Result register (ADCn.RES). Writing a '0' to this bit has no effect.

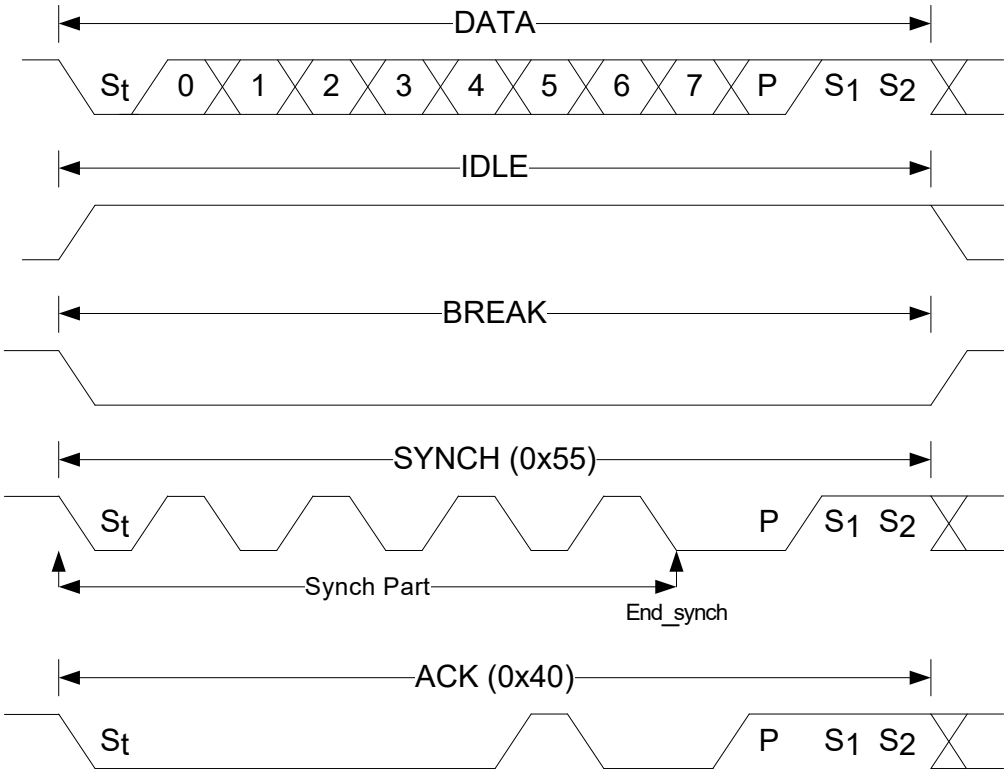
Bit 0 – RESRDY Result Ready Interrupt Flag
The result ready interrupt flag is set when a measurement is complete and a new result is ready. The flag is cleared by either writing a '1' to the bit location or by reading the Result register (ADCn.RES). Writing a '0' to this bit has no effect.

33.3 Functional Description

33.3.1 Principle of Operation

Communication through the UPDI is based on standard UART communication, using a fixed frame format, and automatic baud rate detection for clock and data recovery. In addition to the data frame, there are several control frames which are important to the communication. The supported frame formats are presented in Figure 33-3.

Figure 33-3. Supported UPDI Frame Formats



Data Frame Data frame consists of one Start bit (always low), eight data bits, one parity bit (even parity), and two Stop bits (always high). If the Start bit, parity bit, or Stop bits have an incorrect value, an error will be detected and signalized by the UPDI. The parity bit-check in the UPDI can be disabled by writing the PARD bit in UPDI.CTRLA, in which case the parity generation from the debugger can be ignored.

IDLE Frame Special frame that consists of 12 high bits. This is the same as keeping the transmission line in an Idle state.

BREAK Special frame that consists of 12 low bits. The BREAK frame is used to reset the UPDI back to its default state and is typically used for error recovery.

SYNCH The SYNCH frame (0x55) is used by the Baud Rate Generator to set the baud rate for the coming transmission. A SYNCH character is always expected by the UPDI in front of every new instruction, and after a successful BREAK has been transmitted.

ACK The Acknowledge (ACK) character is transmitted from the UPDI whenever an ST or STS instruction has successfully crossed the synchronization boundary and have gained bus access. When an ACK is received by the debugger, the next transmission can start.

Symbol	Description	Conditions		Min.	Typ.	Max.	Unit
EABS	Absolute accuracy	REFSEL = INTERNAL V_{REF} = 1.1V	T=[0-105]°C V_{DD} = [1.8V- 3.6V]	-	3	-	LSB
			V_{DD} = [1.8V - 3.6V]	-	3	-	
		REFSEL = V_{DD}		-	2	-	
		REFSEL = INTERNAL		-	3	-	
EGAIN	Gain error	REFSEL = INTERNAL V_{REF} = 1.1V	T=[0 - 105]°C V_{DD} = [1.8V - 3.6V]	-	5	-	LSB
			V_{DD} = [1.8V - 3.6V]	-	5	-	
		REFSEL = V_{DD}		-	2	-	
		REFSEL =INTERNAL		-	5	-	
EOFF	Offset error			-	-0.5	-	LSB

Note:

1. A DNL error of less than or equal to 1 LSB ensures a monotonic transfer function with no missing codes.
2. These values are based on characterization and not covered by production test limits.
3. Reference setting and f_{ADC} must fulfill the specification in "Clock and Timing Characteristics" and "Power supply, Reference, and Input Range" tables.

37.17 DAC

V_{DD}=3V, unless stated otherwise.

Table 37-28. Power Supply, Reference, and Input Range

Symbol	Description	Min.	Typ.	Max.	Unit
V_{DD}	Supply Voltage(1)	1.8	3	5.5	V
R_{Load}	Resistive External Load	5	-	-	kΩ
C_{Load}	Capacitive External Load	-	-	30	pF
V_{OUT}	Output Voltage Range	0.2	-	V_{DD}-0.2	V
I_{OUT}	Output sink/source	-	1	-	mA

Note: 1. Supply voltage must meet the V_{DD} specification for the V_{REF} level used as DAC reference.

Figure 38-53. BOD Threshold vs. Temperature (Level 2.6V)

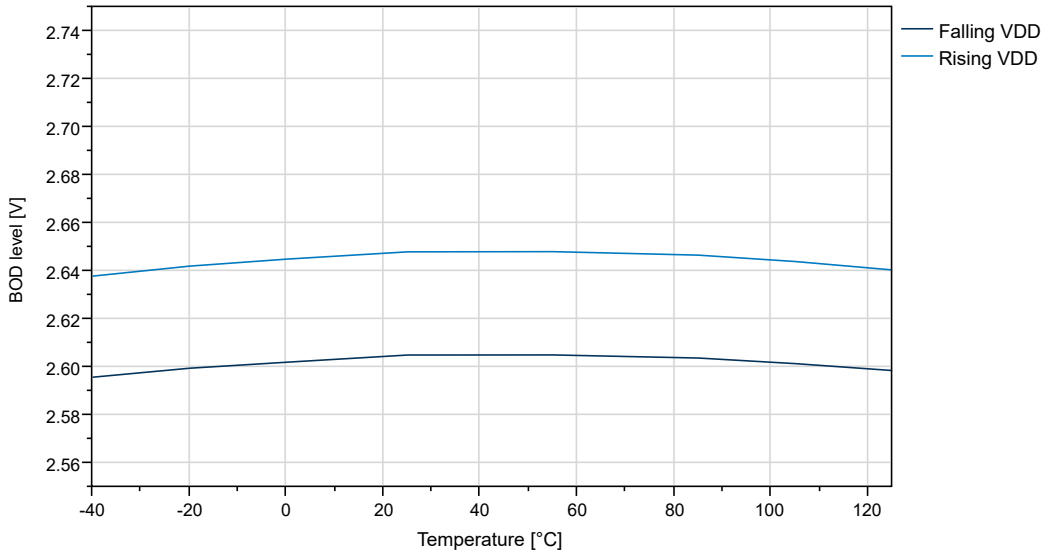


Figure 38-54. BOD Threshold vs. Temperature (Level 4.3V)