



Welcome to [E-XFL.COM](https://www.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.

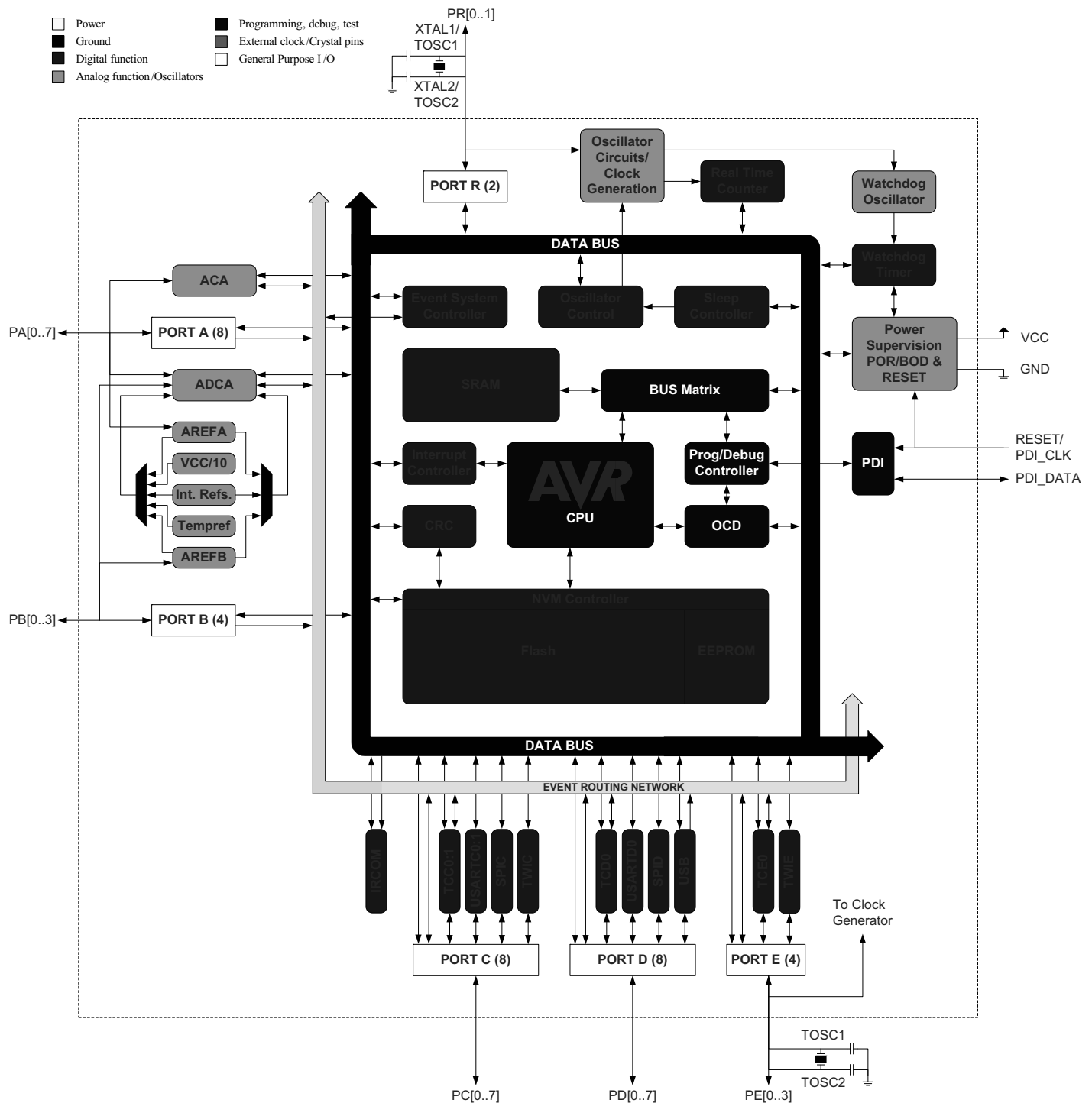
Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Active
Core Processor	AVR
Core Size	8/16-Bit
Speed	32MHz
Connectivity	I ² C, IrDA, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	34
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.6V ~ 3.6V
Data Converters	A/D 16x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VFQFN Exposed Pad
Supplier Device Package	44-VQFN (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atxmega16c4-mn

3.1 Block Diagram

Figure 3-1. XMEGA C4 Block Diagram



device-dependent. These two sections have separate lock bits, and can have different levels of protection. The store program memory (SPM) instruction, which is used to write to the flash from the application software, will only operate when executed from the boot loader section.

The application section contains an application table section with separate lock settings. This enables safe storage of nonvolatile data in the program memory.

Figure 7-1. Flash Program Memory (hexadecimal address)

Word address			
ATxmega32C4		ATxmega16C4	
0		0	Application Section (32K/16K)
			...
37FF	/	17FF	Application Table Section (4K/4K)
3800	/	1800	
3FFF	/	1FFF	
4000	/	2000	Boot Section (4K/4K)
47FF	/	27FF	

7.3.1 Application Section

The Application section is the section of the flash that is used for storing the executable application code. The protection level for the application section can be selected by the boot lock bits for this section. The application section can not store any boot loader code since the SPM instruction cannot be executed from the application section.

7.3.2 Application Table Section

The application table section is a part of the application section of the flash memory that can be used for storing data. The size is identical to the boot loader section. The protection level for the application table section can be selected by the boot lock bits for this section. The possibilities for different protection levels on the application section and the application table section enable safe parameter storage in the program memory. If this section is not used for data, application code can reside here.

7.3.3 Boot Loader Section

While the application section is used for storing the application code, the boot loader software must be located in the boot loader section because the SPM instruction can only initiate programming when executing from this section. The SPM instruction can access the entire flash, including the boot loader section itself. The protection level for the boot loader section can be selected by the boot loader lock bits. If this section is not used for boot loader software, application code can be stored here.

7.3.4 Production Signature Row

The production signature row is a separate memory section for factory programmed data. It contains calibration data for functions such as oscillators and analog modules. Some of the calibration values will be automatically loaded to the corresponding module or peripheral unit during reset. Other values must be loaded from the signature row and written to the corresponding peripheral registers from software. For details on calibration conditions, refer to “Electrical Characteristics” on page 65.

8. Event System

8.1 Features

- System for direct peripheral-to-peripheral communication and signaling
- Peripherals can directly send, receive, and react to peripheral events
 - CPU independent operation
 - 100% predictable signal timing
 - Short and guaranteed response time
- Four event channels for up to four different and parallel signal routing configurations
- Events can be sent and/or used by most peripherals, clock system, and software
- Additional functions include
 - Quadrature decoders
 - Digital filtering of I/O pin state
- Works in active mode and idle sleep mode

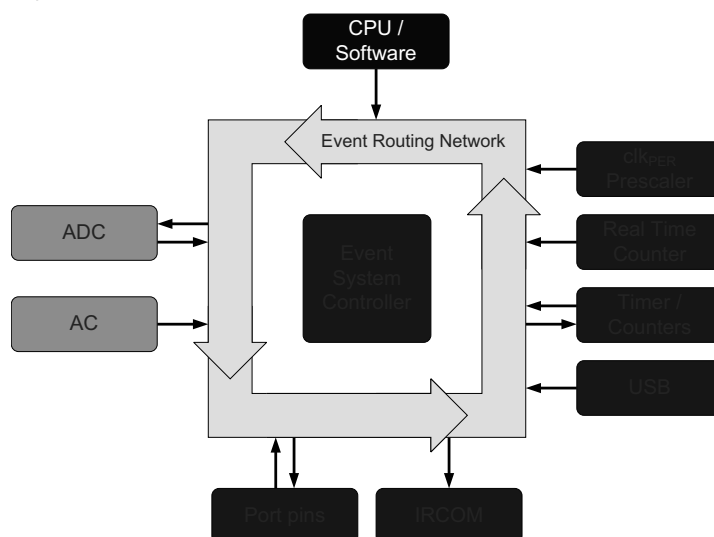
8.2 Overview

The event system enables direct peripheral-to-peripheral communication and signaling. It allows a change in one peripheral's state to automatically trigger actions in other peripherals. It is designed to provide a predictable system for short and predictable response times between peripherals. It allows for autonomous peripheral control and interaction without the use of interrupts, and CPU, and is thus a powerful tool for reducing the complexity, size and execution time of application code. It also allows for synchronized timing of actions in several peripheral modules.

A change in a peripheral's state is referred to as an event, and usually corresponds to the peripheral's interrupt conditions. Events can be directly passed to other peripherals using a dedicated routing network called the event routing network. How events are routed and used by the peripherals is configured in software.

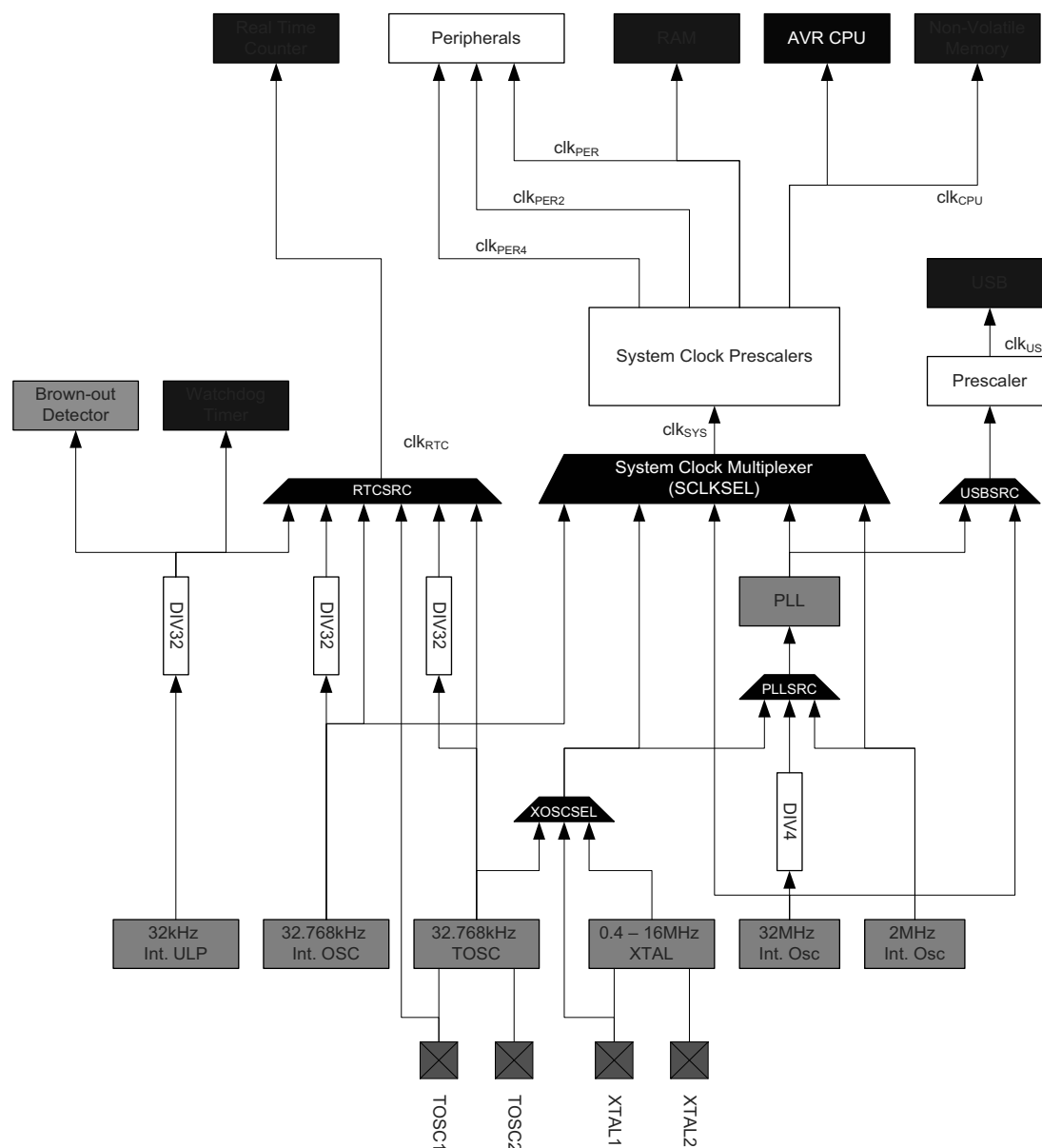
Figure 8-1 shows a basic diagram of all connected peripherals. The event system can directly connect together analog to digital converter, analog comparators, I/O port pins, the real-time counter, timer/counters, IR communication module (IRCOM), and USB interface. Events can also be generated from software and the peripheral clock.

Figure 8-1. Event System Overview and Connected Peripherals



The event routing network consists of four software-configurable multiplexers that control how events are routed and used. These are called event channels, and allow for up to four parallel event routing configurations. The maximum routing latency is two peripheral clock cycles. The event system works in both active mode and idle sleep mode.

Figure 9-1. The Clock System, Clock Sources, and Clock Distribution



9.3 Clock Sources

The clock sources are divided in two main groups: internal oscillators and external clock sources. Most of the clock sources can be directly enabled and disabled from software, while others are automatically enabled or disabled, depending on peripheral settings. After reset, the device starts up running from the 2MHz internal oscillator. The other clock sources (DFLLs and PLL) are turned off by default.

The internal oscillators do not require any external components to run. For details on characteristics and accuracy of the internal oscillators, refer to the device datasheet.

9.3.1 32kHz Ultra Low Power Internal Oscillator

This oscillator provides an approximate 32kHz clock. The 32kHz ultra low power (ULP) internal oscillator is a very low power clock source, and it is not designed for high accuracy. The oscillator employs a built-in prescaler that provides a 1kHz output. The oscillator is automatically enabled/disabled when it is used as clock source for any part of the device. This oscillator can be selected as the clock source for the RTC.

19. RTC – 16-bit Real-Time Counter

19.1 Features

- 16-bit resolution
- Selectable clock source
 - 32.768kHz external crystal
 - External clock
 - 32.768kHz internal oscillator
 - 32kHz internal ULP oscillator
- Programmable 10-bit clock prescaling
- One compare register
- One period register
- Clear counter on period overflow
- Optional interrupt/event on overflow and compare match

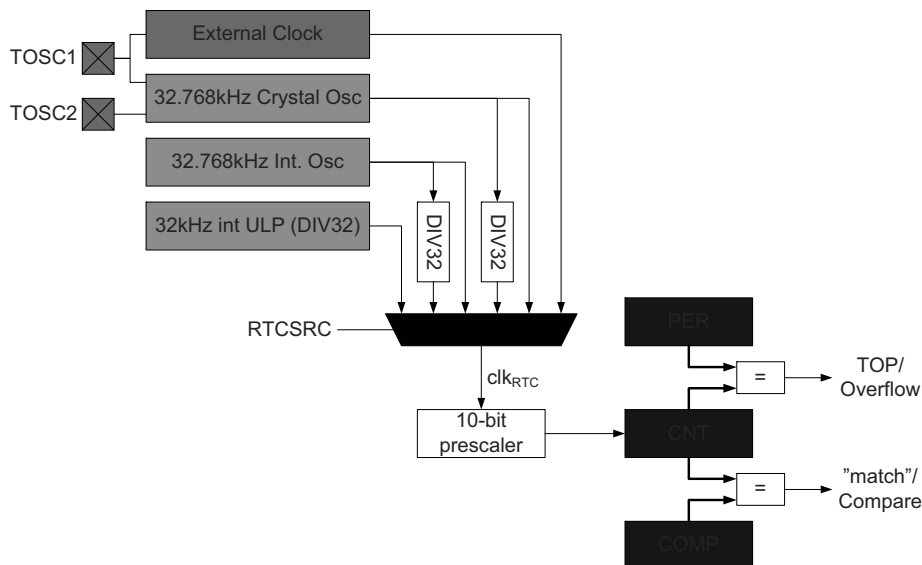
19.2 Overview

The 16-bit real-time counter (RTC) is a counter that typically runs continuously, including in low-power sleep modes, to keep track of time. It can wake up the device from sleep modes and/or interrupt the device at regular intervals.

The reference clock is typically the 1.024kHz output from a high-accuracy crystal of 32.768kHz, and this is the configuration most optimized for low power consumption. The faster 32.768kHz output can be selected if the RTC needs a resolution higher than 1ms. The RTC can also be clocked from an external clock signal, the 32.768kHz internal oscillator or the 32kHz internal ULP oscillator.

The RTC includes a 10-bit programmable prescaler that can scale down the reference clock before it reaches the counter. A wide range of resolutions and time-out periods can be configured. With a 32.768kHz clock source, the maximum resolution is 30.5 μ s, and time-out periods can range up to 2000 seconds. With a resolution of 1s, the maximum timeout period is more than 18 hours (65536 seconds). The RTC can give a compare interrupt and/or event when the counter equals the compare register value, and an overflow interrupt and/or event when it equals the period register value.

Figure 19-1. Real-time Counter Overview



20. USB – Universal Serial Bus Interface

20.1 Features

- One USB 2.0 full speed (12Mbps) and low speed (1.5Mbps) device compliant interface
- Integrated on-chip USB transceiver, no external components needed
- 16 endpoint addresses with full endpoint flexibility for up to 31 endpoints
 - One input endpoint per endpoint address
 - One output endpoint per endpoint address
- Endpoint address transfer type selectable to
 - Control transfers
 - Interrupt transfers
 - Bulk transfers
 - Isochronous transfers
- Configurable data payload size per endpoint, up to 1023 bytes
- Endpoint configuration and data buffers located in internal SRAM
 - Configurable location for endpoint configuration data
 - Configurable location for each endpoint's data buffer
- Built-in direct memory access (DMA) to internal SRAM for:
 - Endpoint configurations
 - Reading and writing endpoint data
- Ping-pong operation for higher throughput and double buffered operation
 - Input and output endpoint data buffers used in a single direction
 - CPU can update data buffer during transfer
- Multipacket transfer for reduced interrupt load and software intervention
 - Data payload exceeding maximum packet size is transferred in one continuous transfer
 - No interrupts or software interaction on packet transaction level
- Transaction complete FIFO for workflow management when using multiple endpoints
 - Tracks all completed transactions in a first-come, first-served work queue
- Clock selection independent of system clock source and selection
- Minimum 1.5MHz CPU clock required for low speed USB operation
- Minimum 12MHz CPU clock required for full speed operation
- Connection to event system
- On chip debug possibilities during USB transactions

20.2 Overview

The USB module is a USB 2.0 full speed (12Mbps) and low speed (1.5Mbps) device compliant interface.

The USB supports 16 endpoint addresses. All endpoint addresses have one input and one output endpoint, for a total of 31 configurable endpoints and one control endpoint. Each endpoint address is fully configurable and can be configured for any of the four transfer types; control, interrupt, bulk, or isochronous. The data payload size is also selectable, and it supports data payloads up to 1023 bytes.

No dedicated memory is allocated for or included in the USB module. Internal SRAM is used to keep the configuration for each endpoint address and the data buffer for each endpoint. The memory locations used for endpoint configurations and data buffers are fully configurable. The amount of memory allocated is fully dynamic, according to the number of endpoints in use and the configuration of these. The USB module has built-in direct memory access (DMA), and will read/write data from/to the SRAM when a USB transaction takes place.

To maximize throughput, an endpoint address can be configured for ping-pong operation. When done, the input and output endpoints are both used in the same direction. The CPU can then read/write one data buffer while the USB module writes/reads the others, and vice versa. This gives double buffered communication.

21. TWI – Two-Wire Interface

21.1 Features

- Two Identical two-wire interface peripherals
- Bidirectional, two-wire communication interface
 - Phillips I²C compatible
 - System Management Bus (SMBus) compatible
- Bus master and slave operation supported
 - Slave operation
 - Single bus master operation
 - Bus master in multi-master bus environment
 - Multi-master arbitration
- Flexible slave address match functions
 - 7-bit and general call address recognition in hardware
 - 10-bit addressing supported
 - Address mask register for dual address match or address range masking
 - Optional software address recognition for unlimited number of addresses
- Slave can operate in all sleep modes, including power-down
- Slave address match can wake device from all sleep modes
- 100kHz and 400kHz bus frequency support
- Slew-rate limited output drivers
- Input filter for bus noise and spike suppression
- Support arbitration between start/repeated start and data bit (SMBus)
- Slave arbitration allows support for address resolve protocol (ARP) (SMBus)

21.2 Overview

The two-wire interface (TWI) is a bidirectional, two-wire communication interface. It is I²C and System Management Bus (SMBus) compatible. The only external hardware needed to implement the bus is one pull-up resistor on each bus line.

A device connected to the bus must act as a master or a slave. The master initiates a data transaction by addressing a slave on the bus and telling whether it wants to transmit or receive data. One bus can have many slaves and one or several masters that can take control of the bus. An arbitration process handles priority if more than one master tries to transmit data at the same time. Mechanisms for resolving bus contention are inherent in the protocol.

The TWI module supports master and slave functionality. The master and slave functionality are separated from each other, and can be enabled and configured separately. The master module supports multi-master bus operation and arbitration. It contains the baud rate generator. Both 100kHz and 400kHz bus frequency is supported. Quick command and smart mode can be enabled to auto-trigger operations and reduce software complexity.

The slave module implements 7-bit address match and general address call recognition in hardware. 10-bit addressing is also supported. A dedicated address mask register can act as a second address match register or as a register for address range masking. The slave continues to operate in all sleep modes, including power-down mode. This enables the slave to wake up the device from all sleep modes on TWI address match. It is possible to disable the address matching to let this be handled in software instead.

The TWI module will detect START and STOP conditions, bus collisions, and bus errors. Arbitration lost, errors, collision, and clock hold on the bus are also detected and indicated in separate status flags available in both master and slave modes.

It is possible to disable the TWI drivers in the device, and enable a four-wire digital interface for connecting to an external TWI bus driver. This can be used for applications where the device operates from a different V_{CC} voltage than used by the TWI bus.

PORTC and PORTE each has one TWI. Notation of these peripherals are TWIC and TWIE.

24. IRCOM – IR Communication Module

24.1 Features

- Pulse modulation/demodulation for infrared communication
- IrDA compatible for baud rates up to 115.2Kbps
- Selectable pulse modulation scheme
 - 3/16 of the baud rate period
 - Fixed pulse period, 8-bit programmable
 - Pulse modulation disabled
- Built-in filtering
- Can be connected to and used by any USART

24.2 Overview

Atmel AVR XMEGA devices contain an infrared communication module (IRCOM) that is IrDA compatible for baud rates up to 115.2Kbps. It can be connected to any USART to enable infrared pulse encoding/decoding for that USART.

33.1.3 Current Consumption

Table 33-4. Current Consumption for Active Mode and Sleep Modes

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
I_{CC}	Active power consumption ⁽¹⁾	32kHz, Ext. Clk	$V_{CC} = 1.8V$	40		μA
			$V_{CC} = 3.0V$	80		
		1MHz, Ext. Clk	$V_{CC} = 1.8V$	200		
			$V_{CC} = 3.0V$	410		
		2MHz, Ext. Clk	$V_{CC} = 1.8V$	350	600	mA
			$V_{CC} = 3.0V$	0.75	1.4	
	Idle power consumption ⁽¹⁾	32kHz, Ext. Clk	$V_{CC} = 1.8V$	2.0		μA
			$V_{CC} = 3.0V$	2.8		
		1MHz, Ext. Clk	$V_{CC} = 1.8V$	42		
			$V_{CC} = 3.0V$	85		
		2MHz, Ext. Clk	$V_{CC} = 1.8V$	85	225	mA
			$V_{CC} = 3.0V$	170	350	
	Power-down power consumption	T = 25°C	$V_{CC} = 3.0V$	0.1	1.0	μA
				2.0	4.5	
		T = 105°C		0.1	7.0	
		WDT and sampled BOD enabled, T = 25°C	$V_{CC} = 3.0V$	1.4	3.0	
		WDT and sampled BOD enabled, T = 85°C		3.0	6.0	
		WDT and sampled BOD enabled, T = 105°C		1.4	10	
	Power-save power consumption ⁽²⁾	RTC from ULP clock, WDT and sampled BOD enabled, T = 25°C	$V_{CC} = 1.8V$	1.5		
			$V_{CC} = 3.0V$	1.5		
		RTC from 1.024kHz low power 32.768kHz TOSC, T = 25°C	$V_{CC} = 1.8V$	0.6	2.0	
			$V_{CC} = 3.0V$	0.7	2.0	
		RTC from low power 32.768kHz TOSC, T = 25°C	$V_{CC} = 1.8V$	0.8	3.0	
			$V_{CC} = 3.0V$	1.0	3.0	
	Reset power consumption	Current through \overline{RESET} pin subtracted	$V_{CC} = 3.0V$	300		

- Notes:
1. All Power Reduction Registers set.
 2. Maximum limits are based on characterization, and not tested in production.

Table 33-29. Two-wire Interface Characteristics

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
V_{IH}	Input high voltage		$0.7 \cdot V_{CC}$		$V_{CC} + 0.5$	V
V_{IL}	Input low voltage		-0.5		$0.3 \cdot V_{CC}$	
V_{hys}	Hysteresis of Schmitt trigger inputs		$0.05 \cdot V_{CC}^{(1)}$			
V_{OL}	Output low voltage	3mA, sink current	0		0.4	
t_r	Rise time for both SDA and SCL		$20 + 0.1 C_b^{(1)(2)}$		300	ns
t_{of}	Output fall time from V_{IHmin} to V_{ILmax}	$10pF < C_b < 400pF^{(2)}$	$20 + 0.1 C_b^{(1)(2)}$		250	
t_{SP}	Spikes suppressed by input filter		0		50	
I_I	Input current for each I/O Pin	$0.1V_{CC} < V_I < 0.9V_{CC}$	-10		10	μA
C_I	Capacitance for each I/O Pin				10	pF
f_{SCL}	SCL clock frequency	$f_{PER}^{(3)} > \max(10f_{SCL}, 250kHz)$	0		400	kHz
R_P	Value of pull-up resistor	$f_{SCL} \leq 100kHz$	$\frac{V_{CC} - 0.4V}{3mA}$		$\frac{100ns}{C_b}$	Ω
		$f_{SCL} > 100kHz$			$\frac{300ns}{C_b}$	
$t_{HD;STA}$	Hold time (repeated) START condition	$f_{SCL} \leq 100kHz$	4.0			μs
		$f_{SCL} > 100kHz$	0.6			
t_{LOW}	Low period of SCL clock	$f_{SCL} \leq 100kHz$	4.7			
		$f_{SCL} > 100kHz$	1.3			
t_{HIGH}	High period of SCL clock	$f_{SCL} \leq 100kHz$	4.0			
		$f_{SCL} > 100kHz$	0.6			
$t_{SU;STA}$	Set-up time for a repeated START condition	$f_{SCL} \leq 100kHz$	4.7			
		$f_{SCL} > 100kHz$	0.6			
$t_{HD;DAT}$	Data hold time	$f_{SCL} \leq 100kHz$	0		3.45	
		$f_{SCL} > 100kHz$	0		0.9	
$t_{SU;DAT}$	Data setup time	$f_{SCL} \leq 100kHz$	250			
		$f_{SCL} > 100kHz$	100			
$t_{SU;STO}$	Setup time for STOP condition	$f_{SCL} \leq 100kHz$	4.0			
		$f_{SCL} > 100kHz$	0.6			
t_{BUF}	Bus free time between a STOP and START condition	$f_{SCL} \leq 100kHz$	4.7			
		$f_{SCL} > 100kHz$	1.3			

- Notes:
1. Required only for $f_{SCL} > 100kHz$.
 2. C_b = Capacitance of one bus line in pF.
 3. f_{PER} = Peripheral clock frequency.

Figure 34-11. Idle Mode Supply Current vs. V_{CC}

$f_{SYS} = \text{MHz external clock}$

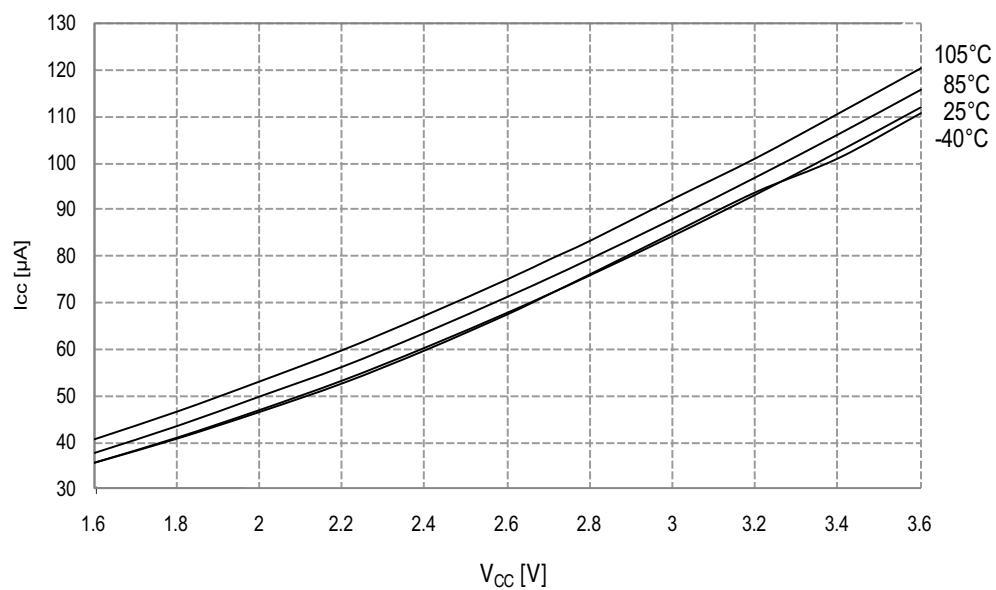


Figure 34-12. Idle Mode Supply Current vs. V_{CC}

$f_{SYS} = 2\text{MHz internal oscillator}$

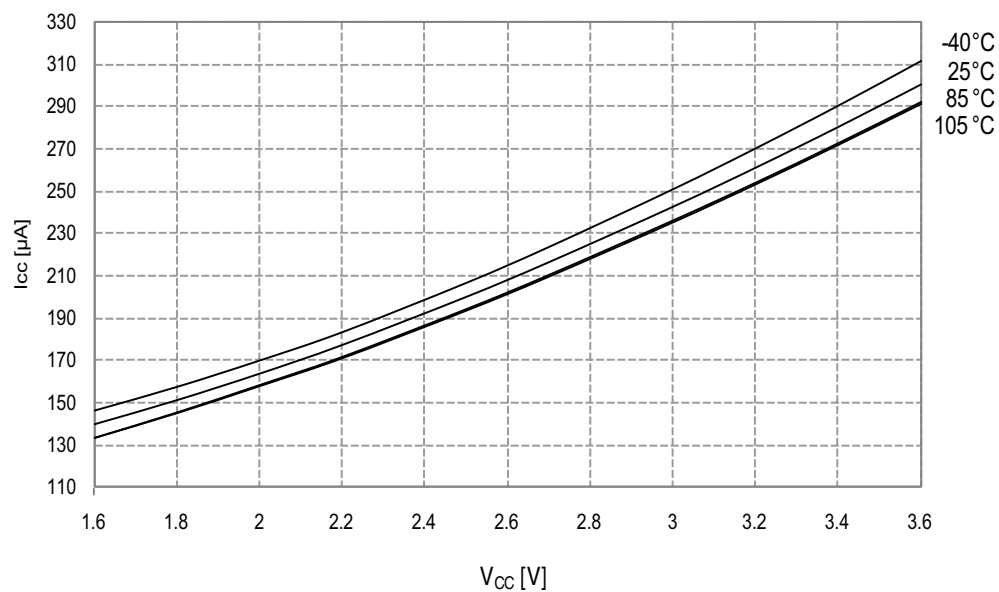
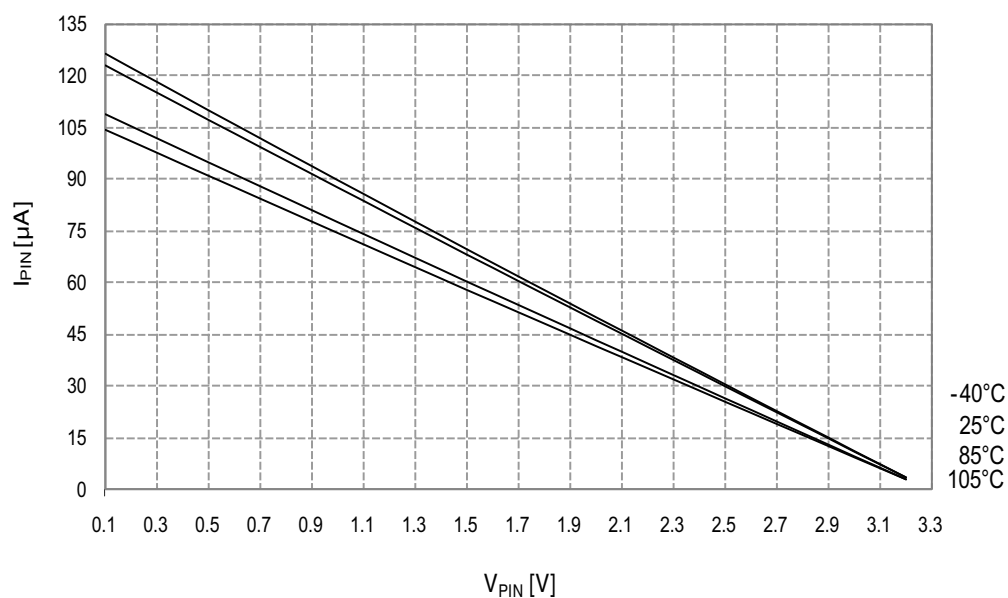


Figure 34-23. I/O Pin Pull-up Resistor Current vs. Input Voltage

$V_{CC} = 3.3V$



34.1.2.2 Output Voltage vs. Sink/Source Current

Figure 34-24. I/O Pin Output Voltage vs. Source Current

$V_{CC} = 1.8V$

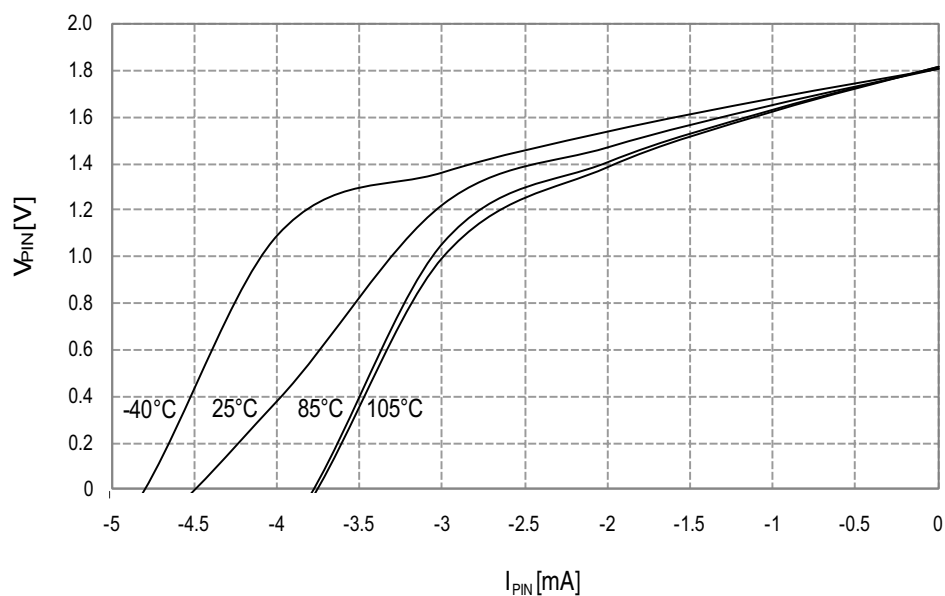


Figure 34-27. I/O Pin Output Voltage vs. Source Current

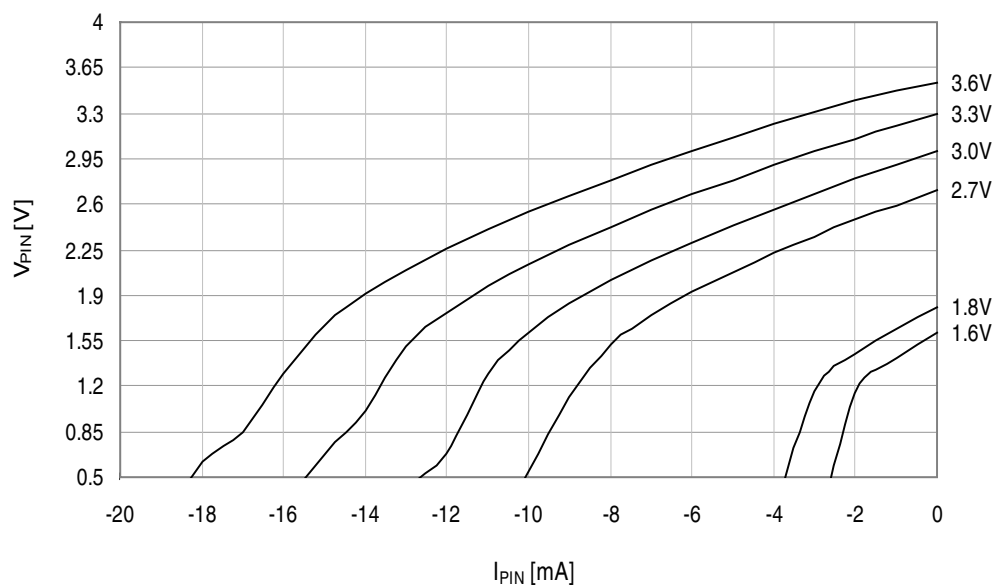


Figure 34-28. I/O Pin Output Voltage vs. Sink Current

$V_{CC} = 1.8V$

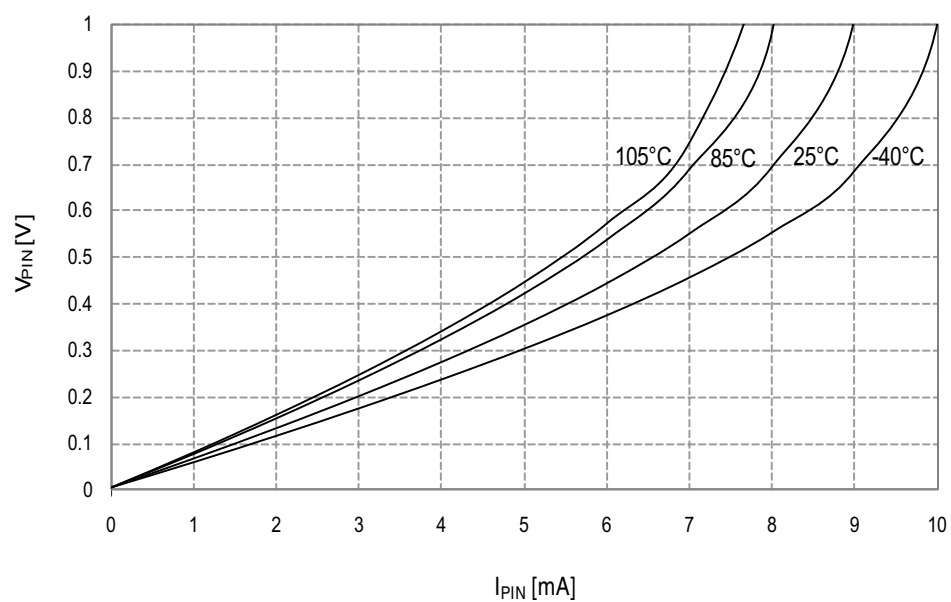
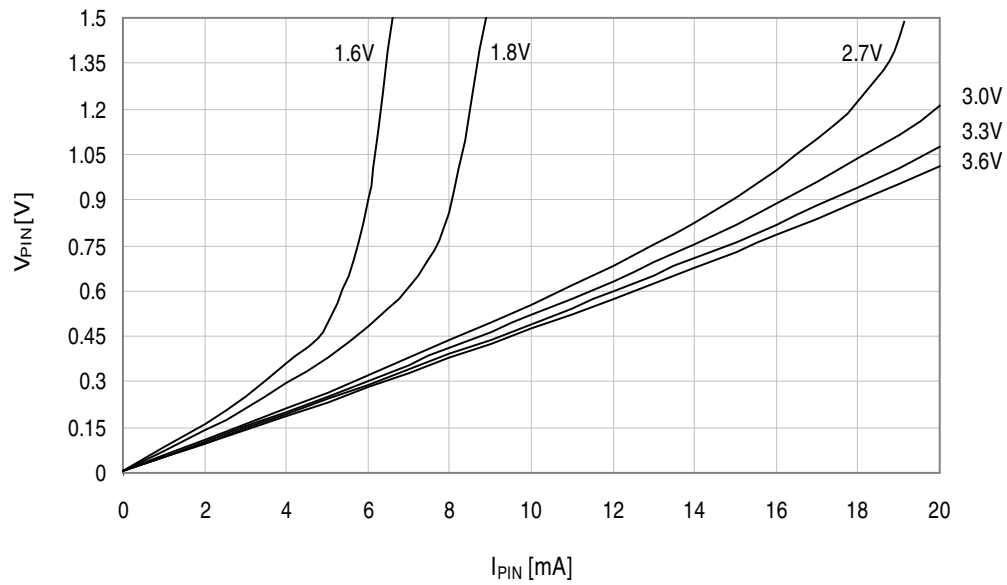


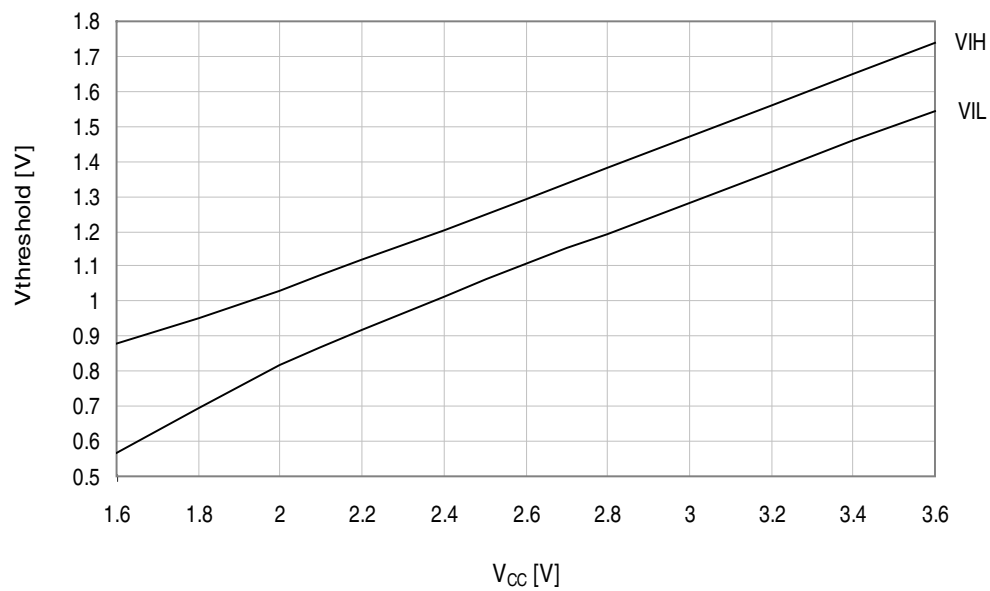
Figure 34-31. I/O Pin Output Voltage vs. Sink Current



34.1.2.3 Thresholds and Hysteresis

Figure 34-32. I/O Pin Input Threshold Voltage vs. V_{CC}

$T = 25^{\circ}\text{C}$



34.1.6 BOD Characteristics

Figure 34-55. BOD Thresholds vs. Temperature
BOD level = 1.6V

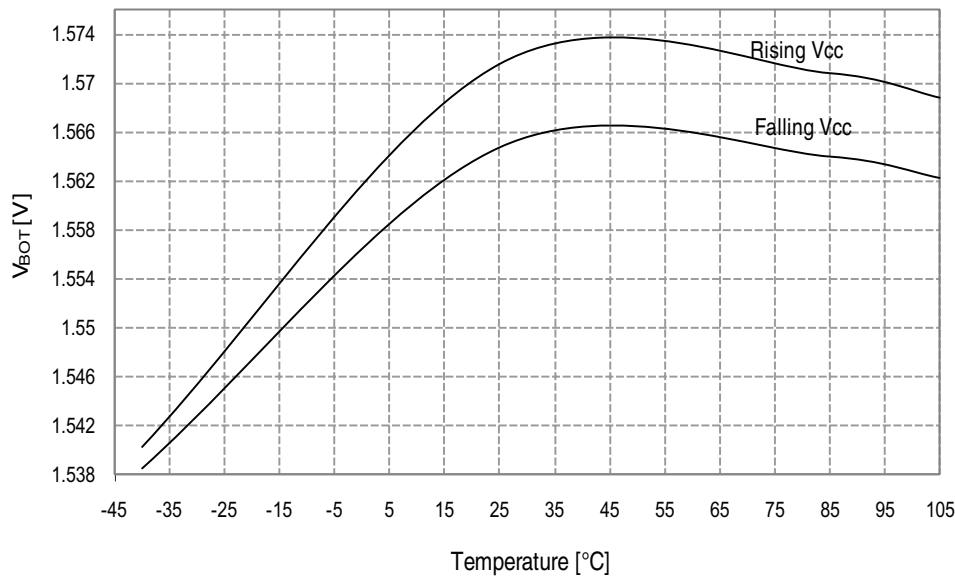
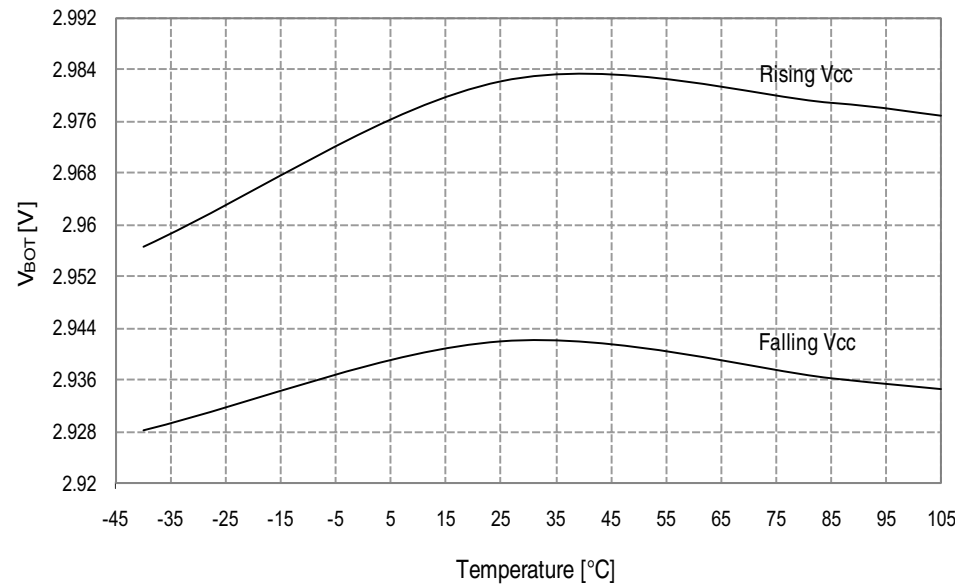


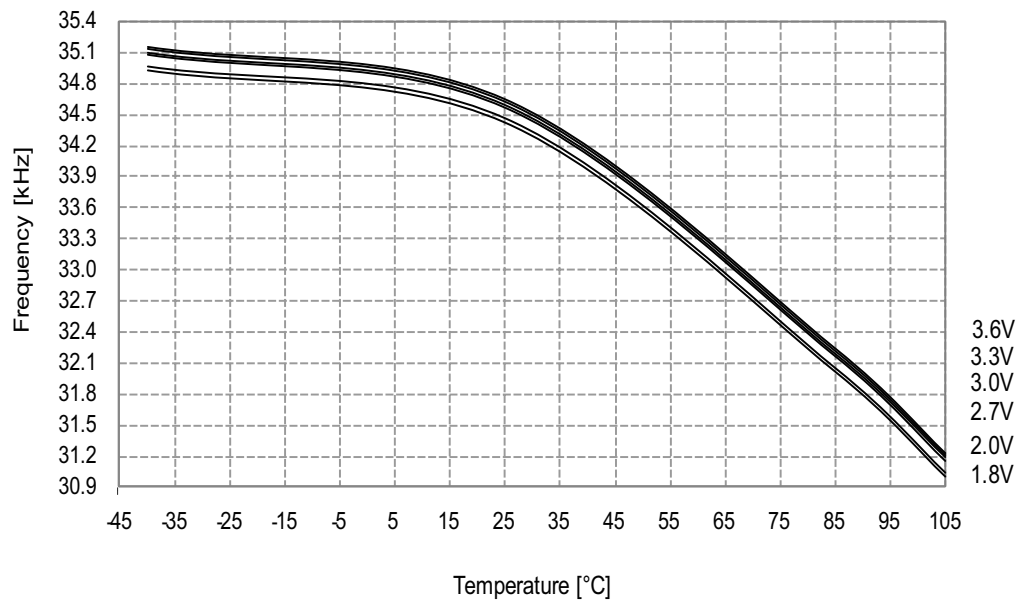
Figure 34-56. BOD Thresholds vs. Temperature
BOD level = 3.0V



34.1.9 Oscillator Characteristics

34.1.9.1 Ultra Low-Power Internal Oscillator

Figure 34-65.Ultra Low-Power Internal Oscillator Frequency vs. Temperature



34.1.9.2 32.768kHz Internal Oscillator

Figure 34-66. 32.768kHz Internal Oscillator Frequency vs. Temperature

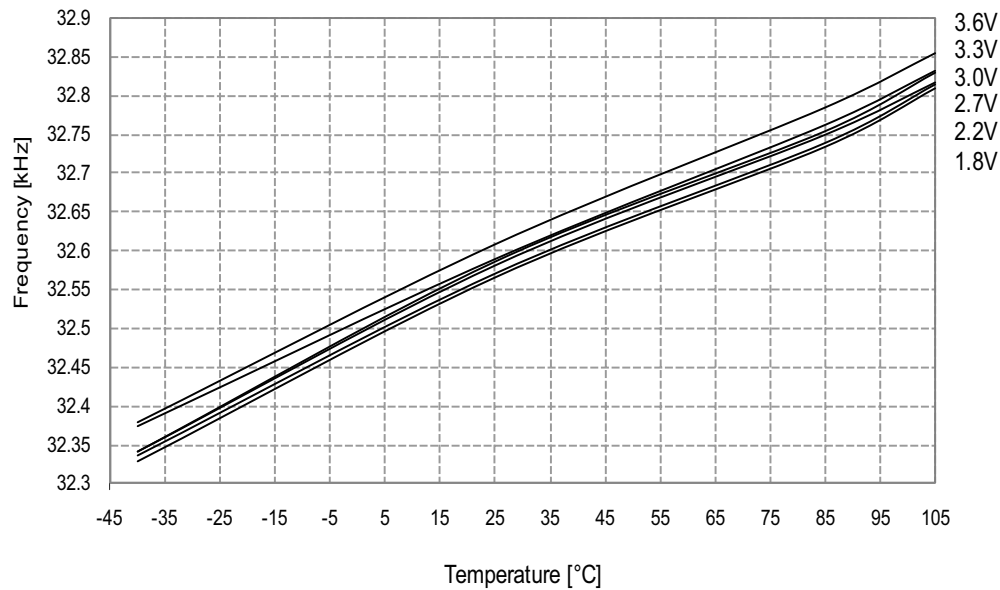


Figure 34-75. 48MHz Internal Oscillator Frequency vs. Temperature
DPLL enabled, from the 32.768kHz internal oscillator

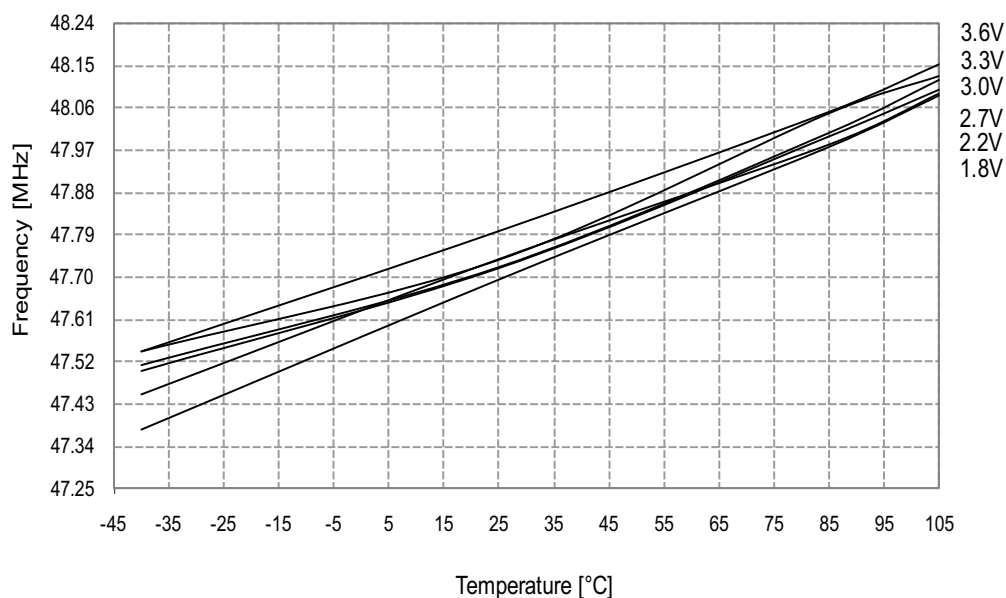


Figure 34-76. 48MHz Internal Oscillator CALA Calibration Step Size
 $V_{CC} = 3.0V$

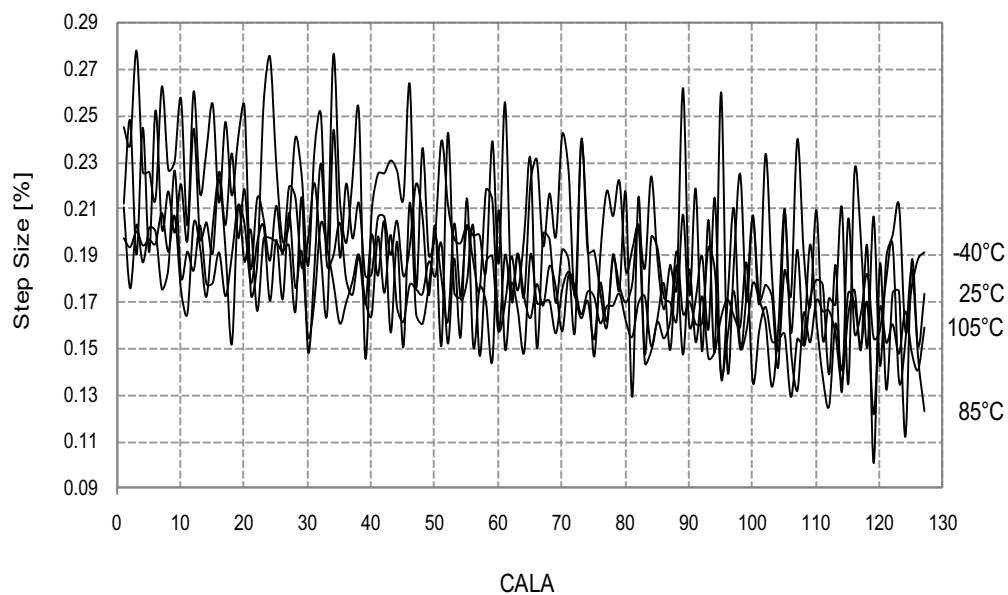


Figure 34-124. Gain Error vs. Temperature

$V_{CC} = 3.0V$, $V_{REF} = \text{external } 2.0V$

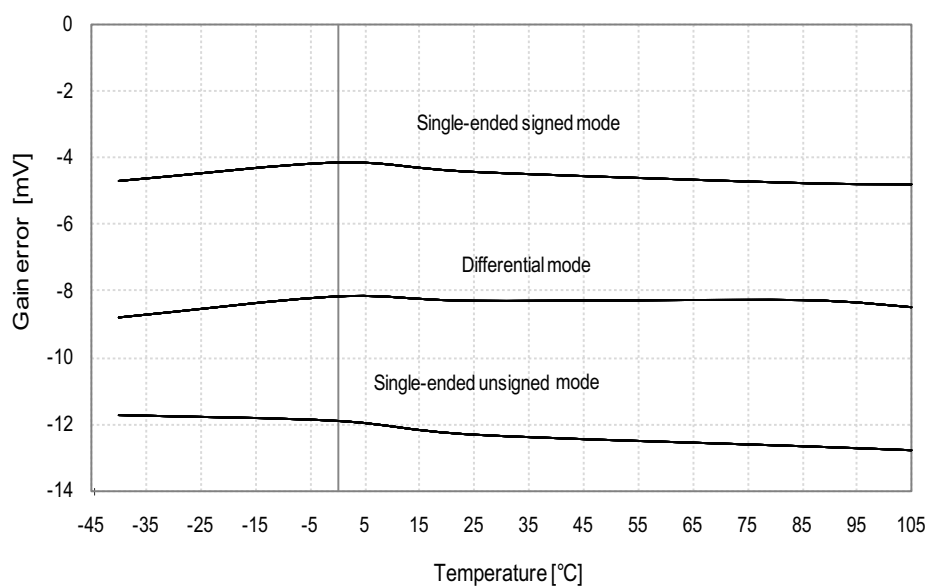


Figure 34-125. Offset Error vs. V_{CC}

$T = 25^{\circ}C$, $V_{REF} = \text{external } 1.0V$, ADC sample rate = 300ksps

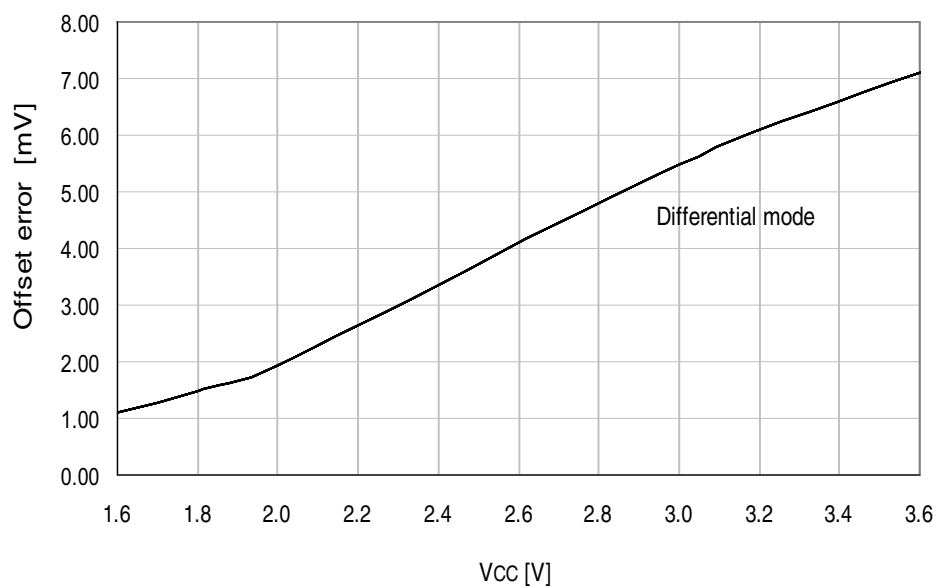


Figure 34-140. Reset Pin Input Threshold Voltage vs. V_{CC}

V_{IH} - Reset pin read as "1"

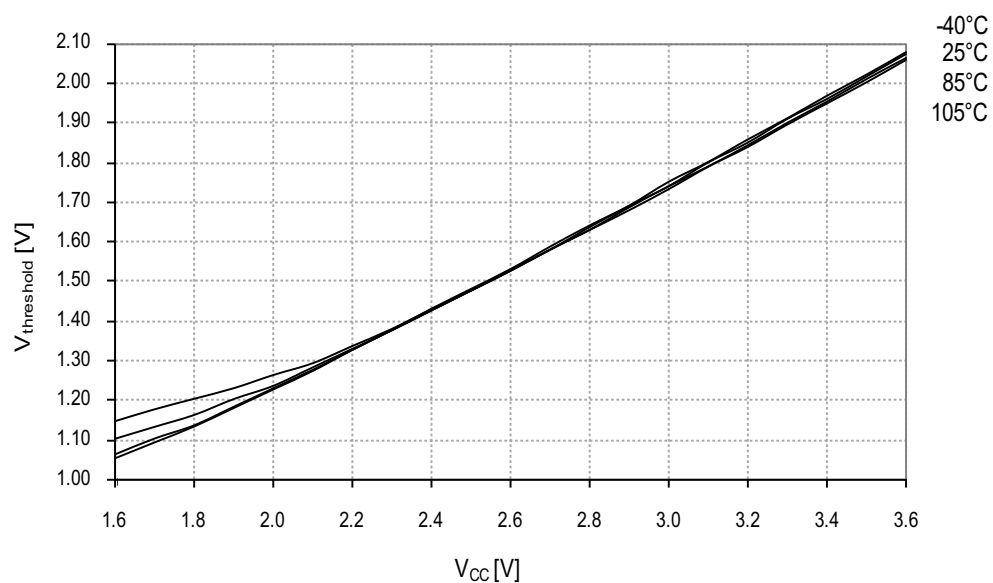
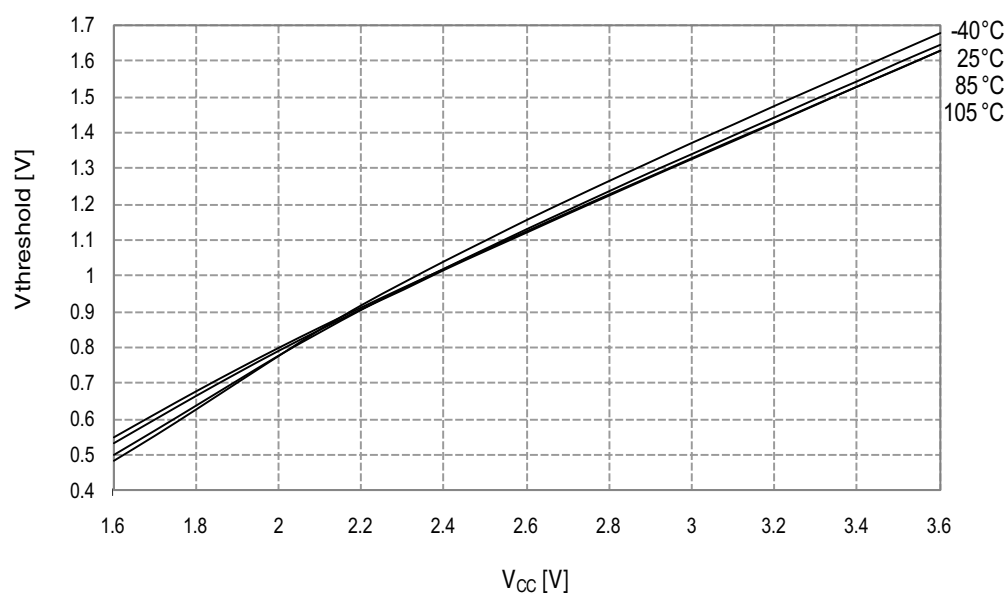


Figure 34-141. Reset Pin Input Threshold Voltage vs. V_{CC}

V_{IL} - Reset pin read as "0"



34.2.11 PDI Characteristics

Figure 34-158. Maximum PDI Frequency vs. V_{CC}

