8/16-bit Atmel AVR XMEGA Microcontrollers

ATxmega32E5 / ATxmega16E5 / ATxmega8E5

Preliminary

Features

- High-performance, low-power Atmel® AVR® XMEGA® 8/16-bit Microcontroller
- Nonvolatile program and data memories
  - 8K –32KBytes of in-system self-programmable flash
  - 2K – 4KBytes boot section
  - 512Bytes – 1KBytes EEPROM
  - 1K – 4KBytes internal SRAM
- Peripheral features
  - Four-channel enhanced DMA controller with 8/16-bit address match
  - Eight-channel event system
    - Asynchronous and synchronous signal routing
    - Quadrature encoder with rotary filter
- Three 16-bit timer/counters
  - One timer/counter with 4 output compare or input capture channels
  - Two timer/counter with 2 output compare or input capture channels
  - High resolution extension enabling down to 4ns PWM resolution
  - Waveform extension for control of motor, LED, lighting, H-bridge, high drives and more
  - Fault extension for safe and deterministic handling and/or shut-down of external driver
- CRC-16 (CRC-CCITT) and CRC-32 (IEEE 802.3) generator
- XMEGA Custom Logic (XCL) module with timer, counter and logic functions
  - Two 8-bit timer/counters with capture/compare and 16-bit cascade mode
  - Connected to one USART to support custom data frame length
  - Connected to I/O pins and event system to do programmable logic functions
    - MUX, AND, NAND, OR, NOR, XOR, XNOR, NOT, D-Flip-Flop, D Latch, RS Latch
- Two USARTs with full-duplex and single wire half-duplex configuration
  - Master SPI mode
  - Support custom protocols with configurable data frame length up to 256-bit
  - System wake-up from deep sleep modes when used with internal 8MHz oscillator
- One two-wire interface with dual address match (I^2C and SMBus compatible)
  - Bridge configuration for simultaneous master and slave operation
  - Up to 1MHz bus speed support
- One serial peripheral interface (SPI)
- 16-bit real time counter with separate oscillator and digital correction
- One sixteen-channel, 12-bit, 300ksps Analog to Digital Converter with:
  - Offset and gain correction
  - Averaging
  - Over-sampling and decimation
- One two-channel, 12-bit, 1Msps Digital to Analog Converter
- Two Analog Comparators with window compare function and current sources
- External interrupts on all general purpose I/O pins
- Programmable watchdog timer with separate on-chip ultra low power oscillator
- QTouch® library support
  - Capacitive touch buttons, sliders and wheels
- Special microcontroller features
  - Power-on reset and programmable brown-out detection
  - Internal and external clock options with PLL
  - Programmable multilevel interrupt controller
  - Five sleep modes
  - Programming and debug interface
    - PDI (Program and Debug Interface)
- I/O and Packages
  - 26 programmable I/O pins
  - 7x7mm 32-lead TQFP
  - 5x5mm 32-lead VQFN
  - 4x4mm 32-lead UQFN
- Operating Voltage
  - 1.6 – 3.6V
- Operating frequency
  - 0 – 12MHz from 1.6V
  - 0 – 32MHz from 2.7V

1. Ordering Information

Ordering Code	Package[1][2][3]	Flash (Bytes)	EEPROM (Bytes)	SRAM (Bytes)	Speed (MHz)	Power supply	Temp.
ATxmega8E5-AU	32A (7x7mm TQFP)	8K + 2K	512B	1K	32	1.6 – 3.6V	-40°C – 85°C
ATxmega8E5-AUR[4]							
ATxmega8E5-MU	32Z (5x5mm VQFN)						
ATxmega8E5-MUR[4]							
ATxmega8E5-M4U	32MA (4x4mm UQFN)						
ATxmega8E5-M4UR[4]							
ATxmega16E5-AU	32A (7x7mm TQFP)	16K + 4K	512B	2K	32	1.6 – 3.6V	-40°C – 85°C
ATxmega16E5-AUR[4]							
ATxmega16E5-MU	32Z (5x5mm VQFN)						
ATxmega16E5-MUR[4]							
ATxmega16E5-M4U	32MA (4x4mm UQFN)						
ATxmega16E5-M4UR[4]							
ATxmega32E5-AU	32A (7x7mm TQFP)	32K + 4K	1K	4K	32	1.6 – 3.6V	-40°C – 85°C
ATxmega32E5AUR[4]							
ATxmega32E5-MU	32Z (5x5mm VQFN)						
ATxmega32E5-MUR[4]							
ATxmega32E5-M4U	32MA (4x4mm UQFN)						
ATxmega32E5-M4UR[4]							

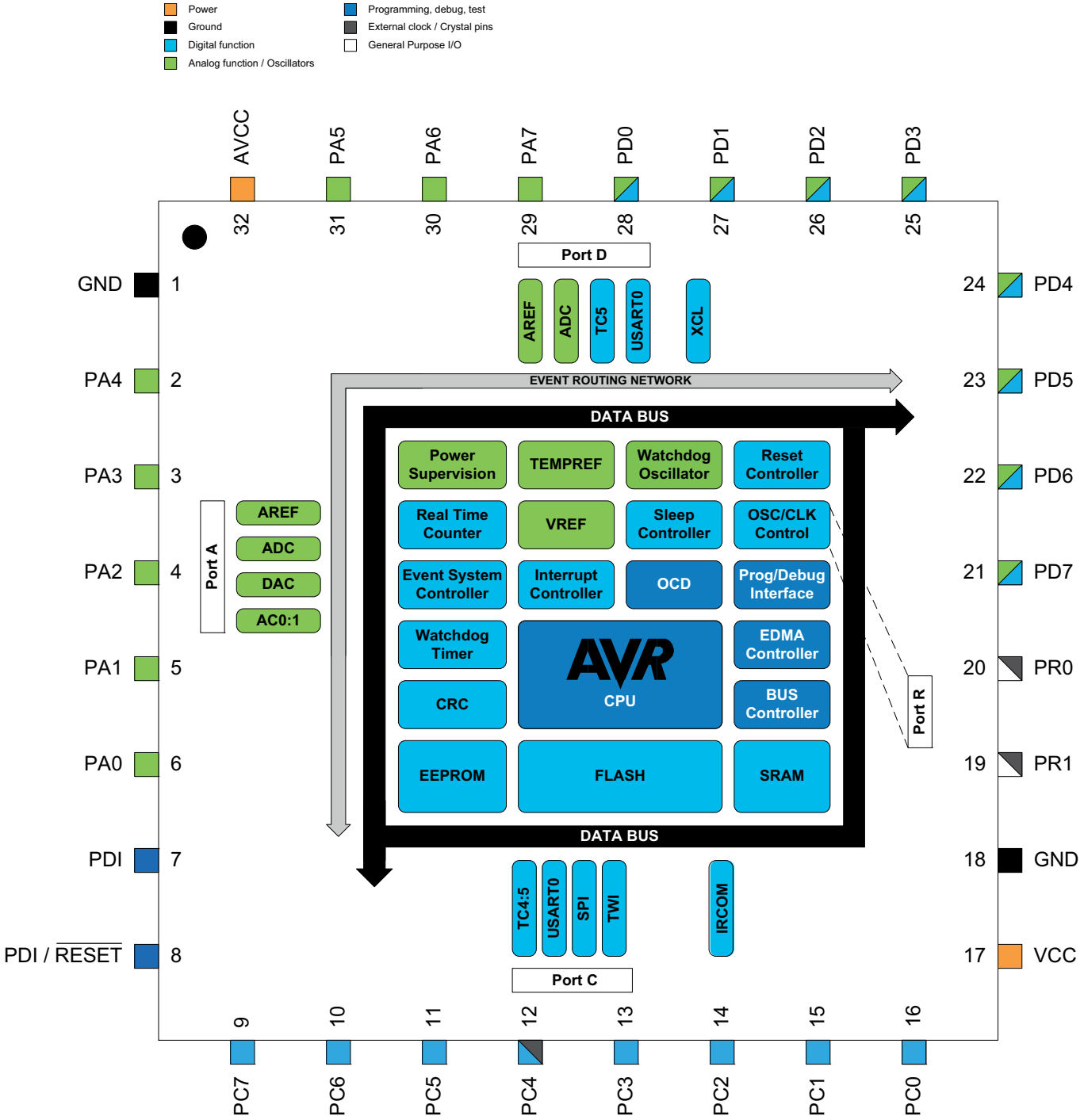
Notes:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information.
2. Pb-free packaging, complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
3. For packaging information, see "Packaging information" on page 68.
4. Tape and Reel

Package Type	
32A	32-lead, 7x7mm body size, 1.0mm body thickness, 0.8mm lead pitch, thin profile plastic quad flat package (TQFP)
32Z	32-lead, 0.5mm pitch, 5x5mm Very Thin quad Flat No Lead Package (VQFN) Sawn
32MA	32-lead, 0.4mm pitch, 4x4x0.60mm Ultra Thin Quad No Lead (UQFN) Package

2. Typical Applications

Board controller	Sensor control	Motor control
User interface	Industrial control	Ballast control, Inverters
Communication bridges	Battery charger	Utility metering
Appliances		

3. Pinout and Block Diagram



Note: 1. For full details on pinout and alternate pin functions refer to "Pinout and Pin Functions" on page 57.

4. Overview

The Atmel AVR XMEGA is a family of low power, high performance, and peripheral rich 8/16-bit microcontrollers based on the AVR enhanced RISC architecture. By executing instructions in a single clock cycle, the AVR XMEGA devices achieve CPU throughput approaching one million instructions per second (MIPS) per megahertz, allowing the system designer to optimize power consumption versus processing speed.

The AVR CPU combines a rich instruction set with 32 general purpose working registers. All 32 registers are directly connected to the arithmetic logic unit (ALU), allowing two independent registers to be accessed in a single instruction, executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs many times faster than conventional single-accumulator or CISC based microcontrollers.

The AVR XMEGA E5 devices provide the following features: in-system programmable flash with read-while-write capabilities; internal EEPROM and SRAM; four-channel enhanced DMA (EDMA) controller; eight-channel event system with asynchronous event support; programmable multilevel interrupt controller; 26 general purpose I/O lines; CRC-16 (CRC-CCITT) and CRC-32 (IEEE 802.3) generators; one XMEGA Custom Logic module with timer, counter and logic functions (XCL); 16-bit real-time counter (RTC) with digital correction; three flexible, 16-bit timer/counters with compare and PWM channels; two USARTs; one two-wire serial interface (TWI) allowing simultaneous master and slave; one serial peripheral interface (SPI); one sixteen-channel, 12-bit ADC with programmable gain, offset and gain correction, averaging, over-sampling and decimation; one 2-channel 12-bit DAC; two analog comparators (ACs) with window mode and current sources; programmable watchdog timer with separate internal oscillator; accurate internal oscillators with PLL and prescaler; and programmable brown-out detection.

The program and debug interface (PDI), a fast, two-pin interface for programming and debugging, is available.

The AVR XMEGA E5 devices have five software selectable power saving modes. The idle mode stops the CPU while allowing the SRAM, EDMA controller, event system, interrupt controller, and all peripherals to continue functioning. The power-down mode saves the SRAM and register contents, but stops the oscillators, disabling all other functions until the next TWI, or pin-change interrupt, or reset. In power-save mode, the asynchronous real-time counter continues to run, allowing the application to maintain a timer base while the rest of the device is sleeping. In standby mode, the external crystal oscillator keeps running while the rest of the device is sleeping. This allows very fast startup from the external crystal, combined with low power consumption. In extended standby mode, both the main oscillator and the asynchronous timer continue to run. In each power save, standby or extended standby mode, the low power mode of the internal 8MHz oscillator allows very fast startup time combined with very low power consumption.

To further reduce power consumption, the peripheral clock to each individual peripheral can optionally be stopped in active mode and idle sleep mode and low power mode of the internal 8MHz oscillator can be enabled.

Atmel offers a free QTouch library for embedding capacitive touch buttons, sliders and wheels functionality into AVR microcontrollers. The devices are manufactured using Atmel high-density, nonvolatile memory technology. The program flash memory can be reprogrammed in-system through the PDI. A boot loader running in the device can use any interface to download the application program to the flash memory. The boot loader software in the boot flash section can continue to run. By combining an 8/16-bit RISC CPU with in-system, self-programmable flash, the AVR XMEGA is a powerful microcontroller family that provides a highly flexible and cost effective solution for many embedded applications.

All Atmel AVR XMEGA devices are supported with a full suite of program and system development tools, including C compilers, macro assemblers, program debugger/simulators, programmers, and evaluation kits.

5. Resources

A comprehensive set of development tools, application notes and datasheets are available for download on http://www.atmel.com/avr.

5.1 Recommended reading

- XMEGA® E Manual
- XMEGA Application Notes

This device data sheet only contains part specific information with a short description of each peripheral and module. The XMEGA E Manual describes the modules and peripherals in depth. The XMEGA application notes contain example code and show applied use of the modules and peripherals.

All documentations are available from www.atmel.com/avr.

6. Capacitive touch sensing

The Atmel® QTouch® library provides a simple to use solution to realize touch sensitive interfaces on most Atmel AVR® microcontrollers. The patented charge-transfer signal acquisition offers robust sensing and includes fully debounced reporting of touch keys and includes Adjacent key suppression® (AKS®) technology for unambiguous detection of key events. The QTouch library includes support for the QTouch and QMatrix acquisition methods.

Touch sensing can be added to any application by linking the appropriate Atmel QTouch library for the AVR Microcontroller. This is done by using a simple set of APIs to define the touch channels and sensors, and then calling the touch sensing API's to retrieve the channel information and determine the touch sensor states.

The Atmel QTouch library is FREE and downloadable from the Atmel website at the following location: http://www.atmel.com/tools/QTOUCHLIBRARY.aspx. For implementation details and other information, refer to the Atmel QTouch library user guide - also available for download from the Atmel website.

7. CPU

7.1 Features

- 8/16-bit, high-performance Atmel AVR RISC CPU
- 142 instructions
- Hardware multiplier
- 32x8-bit registers directly connected to the ALU
- Stack in RAM
- Stack pointer accessible in I/O memory space
- Direct addressing of up to 16MB of program memory and 16MB of data memory
- True 16/24-bit access to 16/24-bit I/O registers
- Efficient support for 8-, 16-, and 32-bit arithmetic
- Configuration change protection of system-critical features

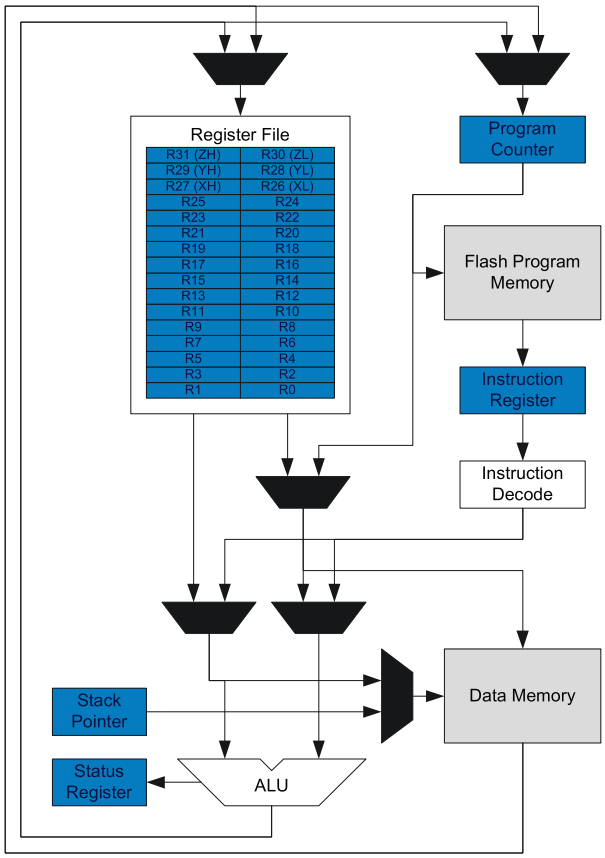
7.2 Overview

All AVR XMEGA devices use the 8/16-bit AVR CPU. The main function of the CPU is to execute the code and perform all calculations. The CPU is able to access memories, perform calculations, control peripherals, and execute the program in the flash memory. Interrupt handling is described in a separate section, refer to “Interrupts and Programmable Multilevel Interrupt Controller” on page 27.

7.3 Architectural Overview

In order to maximize performance and parallelism, the AVR CPU uses a Harvard architecture with separate memories and buses for program and data. Instructions in the program memory are executed with single-level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This enables instructions to be executed on every clock cycle. For details of all AVR instructions, refer to http://www.atmel.com/avr.

Figure 7-1. Block Diagram of the AVR CPU architecture.



The arithmetic logic unit (ALU) supports arithmetic and logic operations between registers or between a constant and a register. Single-register operations can also be executed in the ALU. After an arithmetic operation, the status register is updated to reflect information about the result of the operation.

The ALU is directly connected to the fast-access register file. The 32 x 8-bit general purpose working registers all have single clock cycle access time allowing single-cycle arithmetic logic unit (ALU) operation between registers or between a register and an immediate. Six of the 32 registers can be used as three 16-bit address pointers for program and data space addressing, enabling efficient address calculations.

The memory spaces are linear. The data memory space and the program memory space are two different memory spaces.

The data memory space is divided into I/O registers, SRAM, and memory mapped EEPROM.

All I/O status and control registers reside in the lowest 4KB addresses of the data memory. This is referred to as the I/O memory space. The lowest 64 addresses can be accessed directly, or as the data space locations from 0x00 to 0x3F. The rest is the extended I/O memory space, ranging from 0x0040 to 0x0FFF. I/O registers here must be accessed as data space locations using load (LD/LDS/LDD) and store (ST/STS/STD) instructions.

The SRAM holds data. Code execution from SRAM is not supported. It can easily be accessed through the five different addressing modes supported in the AVR architecture. The first SRAM address is 0x2000.

Data addresses 0x1000 to 0x1FFF are reserved for EEPROM.

The program memory is divided in two sections, the application program section and the boot program section. Both sections have dedicated lock bits for write and read/write protection. The SPM instruction that is used for self-programming of the application flash memory must reside in the boot program section. The application section contains an application table section with separate lock bits for write and read/write protection. The application table section can be used for save storing of nonvolatile data in the program memory.

7.4 ALU - Arithmetic Logic Unit

The arithmetic logic unit (ALU) supports arithmetic and logic operations between registers or between a constant and a register. Single-register operations can also be executed. The ALU operates in direct connection with all 32 general purpose registers. In a single clock cycle, arithmetic operations between general purpose registers or between a register and an immediate are executed and the result is stored in the register file. After an arithmetic or logic operation, the status register is updated to reflect information about the result of the operation.

ALU operations are divided into three main categories – arithmetic, logical, and bit functions. Both 8- and 16-bit arithmetic is supported, and the instruction set allows for efficient implementation of 32-bit arithmetic. The hardware multiplier supports signed and unsigned multiplication and fractional format.

7.4.1 Hardware Multiplier

The multiplier is capable of multiplying two 8-bit numbers into a 16-bit result. The hardware multiplier supports different variations of signed and unsigned integer and fractional numbers:

- Multiplication of unsigned integers
- Multiplication of signed integers
- Multiplication of a signed integer with an unsigned integer
- Multiplication of unsigned fractional numbers
- Multiplication of signed fractional numbers
- Multiplication of a signed fractional number with an unsigned one

A multiplication takes two CPU clock cycles.

7.5 Program Flow

After reset, the CPU starts to execute instructions from the lowest address in the flash program memory '0.' The program counter (PC) addresses the next instruction to be fetched.

Program flow is provided by conditional and unconditional jump and call instructions capable of addressing the whole address space directly. Most AVR instructions use a 16-bit word format, while a limited number use a 32-bit format.

During interrupts and subroutine calls, the return address PC is stored on the stack. The stack is allocated in the general data SRAM, and consequently the stack size is only limited by the total SRAM size and the usage of the SRAM. After reset, the stack pointer (SP) points to the highest address in the internal SRAM. The SP is read/write accessible in the I/O memory space, enabling easy implementation of multiple stacks or stack areas. The data SRAM can easily be accessed through the five different addressing modes supported in the AVR CPU.

7.6 Status Register

The status register (SREG) contains information about the result of the most recently executed arithmetic or logic instruction. This information can be used for altering program flow in order to perform conditional operations. Note that the status register is updated after all ALU operations, as specified in the instruction set reference. This will in many cases remove the need for using the dedicated compare instructions, resulting in faster and more compact code.

The status register is not automatically stored when entering an interrupt routine nor restored when returning from an interrupt. This must be handled by software.

The status register is accessible in the I/O memory space.

7.7 Stack and Stack Pointer

The stack is used for storing return addresses after interrupts and subroutine calls. It can also be used for storing temporary data. The stack pointer (SP) register always points to the top of the stack. It is implemented as two 8-bit registers that are accessible in the I/O memory space. Data are pushed and popped from the stack using the PUSH and POP instructions. The stack grows from a higher memory location to a lower memory location. This implies that pushing data onto the stack decreases the SP, and popping data off the stack increases the SP. The SP is automatically loaded

after reset, and the initial value is the highest address of the internal SRAM. If the SP is changed, it must be set to point above address 0x2000, and it must be defined before any subroutine calls are executed or before interrupts are enabled.

During interrupts or subroutine calls, the return address is automatically pushed on the stack. The return address can be two or three bytes, depending on program memory size of the device. For devices with 128KB or less of program memory, the return address is two bytes, and hence the stack pointer is decremented/incremented by two. For devices with more than 128KB of program memory, the return address is three bytes, and hence the SP is decremented/incremented by three. The return address is popped off the stack when returning from interrupts using the RETI instruction, and from subroutine calls using the RET instruction.

The SP is decremented by one when data are pushed on the stack with the PUSH instruction, and incremented by one when data is popped off the stack using the POP instruction.

To prevent corruption when updating the stack pointer from software, a write to SPL will automatically disable interrupts for up to four instructions or until the next I/O memory write.

7.8 Register File

The register file consists of 32 x 8-bit general purpose working registers with single clock cycle access time. The register file supports the following input/output schemes:

- One 8-bit output operand and one 8-bit result input
- Two 8-bit output operands and one 8-bit result input
- Two 8-bit output operands and one 16-bit result input
- One 16-bit output operand and one 16-bit result input

Six of the 32 registers can be used as three 16-bit address register pointers for data space addressing, enabling efficient address calculations. One of these address pointers can also be used as an address pointer for lookup tables in flash program memory.

8. Memories

8.1 Features

- Flash program memory
  - One linear address space
  - In-system programmable
  - Self-programming and boot loader support
  - Application section for application code
  - Application table section for application code or data storage
  - Boot section for application code or bootloader code
  - Separate read/write protection lock bits for all sections
  - Built in fast CRC check of a selectable flash program memory section
- Data memory
  - One linear address space
  - Single-cycle access from CPU
  - SRAM
  - EEPROM
    - Byte and page accessible
    - Memory mapped for direct load and store
  - I/O memory
    - Configuration and status registers for all peripherals and modules
    - 4 bit-accessible general purpose registers for global variables or flags
  - Bus arbitration
    - Deterministic handling of priority between CPU, EDMA controller, and other bus masters
  - Separate buses for SRAM, EEPROM and I/O memory
    - Simultaneous bus access for CPU and EDMA controller
- Production signature row memory for factory programmed data
  - ID for each microcontroller device type
  - Serial number for each device
  - Calibration bytes for factory calibrated peripherals
- User signature row
  - One flash page in size
  - Can be read and written from software
  - Content is kept after chip erase

8.2 Overview

The Atmel AVR architecture has two main memory spaces, the program memory and the data memory. Executable code can reside only in the program memory, while data can be stored in the program memory and the data memory. The data memory includes the internal SRAM, and EEPROM for nonvolatile data storage. All memory spaces are linear and require no memory bank switching. Nonvolatile memory (NVM) spaces can be locked for further write and read/write operations. This prevents unrestricted access to the application software.

A separate memory section contains the fuse bytes. These are used for configuring important system functions, and can only be written by an external programmer.

The available memory size configurations are shown in “Ordering Information” on page 2”. In addition, each device has a Flash memory signature row for calibration data, device identification, serial number etc.

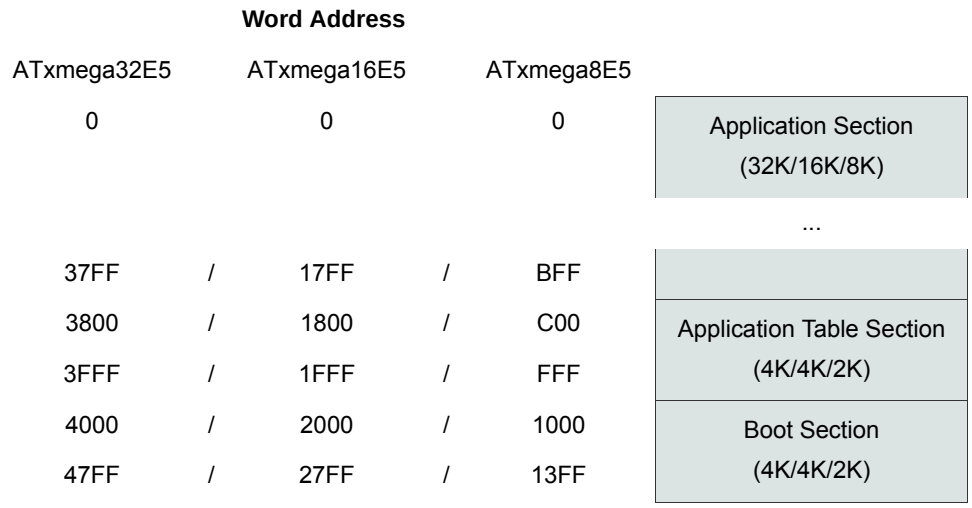
8.3 Flash Program Memory

The Atmel® AVR® XMEGA® devices contain on-chip, in-system reprogrammable flash memory for program storage. The flash memory can be accessed for read and write from an external programmer through the PDI or from application software running in the device.

All AVR CPU instructions are 16 or 32 bits wide, and each flash location is 16 bits wide. The flash memory is organized in two main sections, the application section and the boot loader section. The sizes of the different sections are fixed, but 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 8-1. Flash Program Memory (Hexadecimal address).



8.3.1 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.

8.3.2 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.

8.3.3 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. When programming, the CPU is halted, waiting for the flash operation to complete. 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.

8.3.4 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 71.

The production signature row also contains an ID that identifies each microcontroller device type and a serial number for each manufactured device. The serial number consists of the production lot number, wafer number, and wafer coordinates for the device. The device ID for the available devices is shown in Table 8-1.

The production signature row cannot be written or erased, but it can be read from application software and external programmers.

Table 8-1. Device ID bytes for Atmel AVR XMEGA E5 devices.

Device	Device ID bytes		
	Byte 2	Byte 1	Byte 0
ATxmega32E5	4C	95	1E
ATxmega16E5	45	94	1E
ATxmega8E5	41	93	1E

8.3.5 User Signature Row

The user signature row is a separate memory section that is fully accessible (read and write) from application software and external programmers. It is one flash page in size, and is meant for static user parameter storage, such as calibration data, custom serial number, identification numbers, random number seeds, etc. This section is not erased by chip erase commands that erase the flash, and requires a dedicated erase command. This ensures parameter storage during multiple program/erase operations and on-chip debug sessions.

8.4 Fuses and Lock bits

The fuses are used to configure important system functions, and can only be written from an external programmer. The application software can read the fuses. The fuses are used to configure reset sources such as brownout detector and watchdog, startup configuration, etc.

The lock bits are used to set protection levels for the different flash sections (i.e., if read and/or write access should be blocked). Lock bits can be written by external programmers and application software, but only to stricter protection levels. Chip erase is the only way to erase the lock bits. To ensure that flash contents are protected even during chip erase, the lock bits are erased after the rest of the flash memory has been erased.

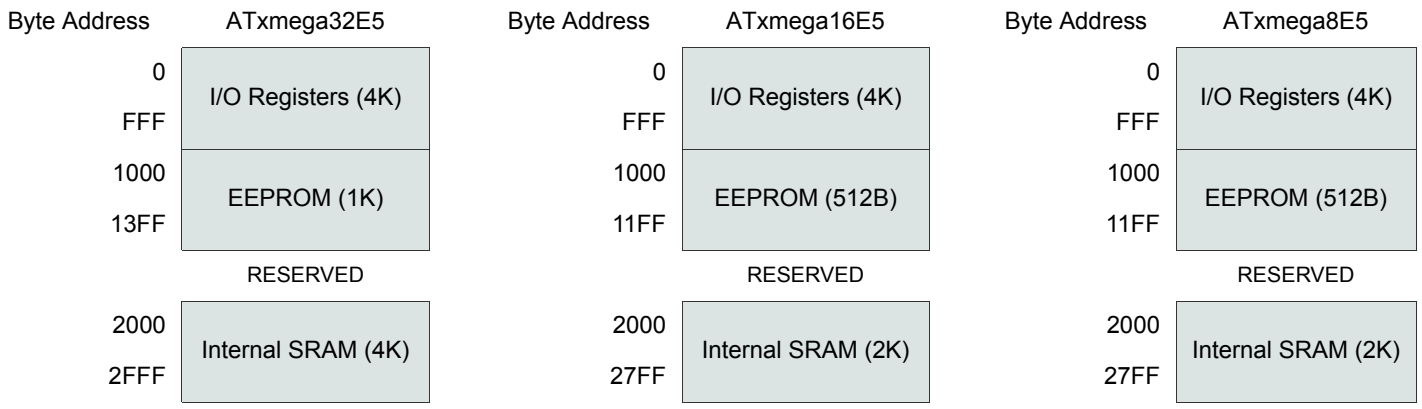
An un-programmed fuse or lock bit will have the value one, while a programmed fuse or lock bit will have the value zero.

Both fuses and lock bits are reprogrammable like the flash program memory.

8.5 Data Memory

The data memory contains the I/O memory, internal SRAM and EEPROM. The data memory is organized as one continuous memory section, see Table 8-2 on page 14. To simplify development, I/O Memory, EEPROM and SRAM will always have the same start addresses for all XMEGA devices.

Figure 8-2. Data Memory Map (Hexadecimal Value)



8.6 EEPROM

Atmel AVR XMEGA E5 devices have EEPROM for nonvolatile data storage. It is memory mapped and accessed in normal data space. The EEPROM supports both byte and page access. EEPROM allows highly efficient EEPROM reading and EEPROM buffer loading. When doing this, EEPROM is accessible using load and store instructions. EEPROM will always start at hexadecimal address 0x1000.

8.7 I/O Memory

The status and configuration registers for peripherals and modules, including the CPU, are addressable through I/O memory locations. All I/O locations can be accessed by the load (LD/LDS/LDD) and store (ST/STS/STD) instructions, which are used to transfer data between the 32 registers in the register file and the I/O memory. The IN and OUT instructions can address I/O memory locations in the range of 0x00 to 0x3F directly. In the address range 0x00 - 0x1F, single-cycle instructions for manipulation and checking of individual bits are available.

The I/O memory address for all peripherals and modules in XMEGA E5 is shown in the "Peripheral Module Address Map" on page 61.

8.7.1 General Purpose I/O Registers

The lowest 4 I/O memory addresses are reserved as general purpose I/O registers. These registers can be used for storing global variables and flags, as they are directly bit-accessible using the SBI, CBI, SBIS, and SBIC instructions.

8.8 Data Memory and Bus Arbitration

Since the data memory is organized as three separate sets of memories, the different bus masters (CPU, EDMA controller read and EDMA controller write, etc.) can access different memory sections at the same time.

8.9 Memory Timing

Read and write access to the I/O memory takes one CPU clock cycle. A write to SRAM takes one cycle, and a read from SRAM takes two cycles. For burst read (EDMA), new data are available every cycle. EEPROM page load (write) takes one cycle, and three cycles are required for read. For burst read, new data are available every second cycle. Refer to the instruction summary for more details on instructions and instruction timing.

8.10 Device ID and Revision

Each device has a three-byte device ID. This ID identifies Atmel as the manufacturer of the device and the device type. A separate register contains the revision number of the device.

8.11 I/O Memory Protection

Some features in the device are regarded as critical for safety in some applications. Due to this, it is possible to lock the I/O register related to the clock system, the event system, and the waveform extensions. As long as the lock is enabled, all related I/O registers are locked and they cannot be written from the application software. The lock registers themselves are protected by the configuration change protection mechanism.

8.12 Flash and EEPROM Page Size

The flash program memory and EEPROM data memory are organized in pages. The pages are word accessible for the flash and byte accessible for the EEPROM.

Table 8-2 shows the Flash Program Memory organization and Program Counter (PC) size. Flash write and erase operations are performed on one page at a time, while reading the Flash is done one byte at a time. For Flash access the Z-pointer (Z[m:n]) is used for addressing. The most significant bits in the address (FPAGE) give the page number and the least significant address bits (FWORD) give the word in the page.

Table 8-2. Number of words and pages in the flash.

Devices	PC size	Flash size	Page Size	FWORD	FPAGE	Application		Boot	
	bits	bytes	words			Size	No of pages	Size	No of pages
ATxmega32E5	15	32K+4K	64	Z[6:0]	Z[14:7]	32K	256	4K	32
ATxmega16E5	14	16K+4K	64	Z[6:0]	Z[13:7]	16K	128	4K	32
ATxmega8E5	13	8K+2K	64	Z[6:0]	Z[12:7]	8K	64	2K	16

Table 8-3 shows EEPROM memory organization for the Atmel AVR XMEGA E5 devices. EEPROM write and erase operations can be performed one page or one byte at a time, while reading the EEPROM is done one byte at a time. For EEPROM access the NVM address register (ADDR[m:n]) is used for addressing. The most significant bits in the address (E2PAGE) give the page number and the least significant address bits (E2BYTE) give the byte in the page.

Table 8-3. Number of words and pages in the EEPROM.

Devices	EEPROM	Page Size	E2BYTE	E2PAGE	No of Pages
	Size	bytes			
ATxmega32E5	1K	32	ADDR[4:0]	ADDR[10:5]	32
ATxmega16E5	512Bytes	32	ADDR[4:0]	ADDR[10:5]	16
ATxmega8E5	512Bytes	32	ADDR[4:0]	ADDR[10:5]	16

9. EDMA – Enhanced DMA Controller

9.1 Features

- The EDMA Controller allows data transfers with minimal CPU intervention
  - from data memory to data memory
  - from data memory to peripheral
  - from peripheral to data memory
  - from peripheral to peripheral
- Four peripheral EDMA channels with separate:
  - transfer triggers
  - interrupt vectors
  - addressing modes
  - data matching
- Two peripheral channels can be combined to one standard channel with separate:
  - transfer triggers
  - interrupt vectors
  - addressing modes
  - data search
- Programmable channel priority
- From 1byte to 128KB of data in a single transaction
  - Up to 64K block transfer with repeat
  - 1 or 2 bytes burst transfers
- Multiple addressing modes
  - Static
  - Increment
- Optional reload of source and destination address at the end of each
  - Burst
  - Block
  - Transaction
- Optional Interrupt on end of transaction
- Optional connection to CRC Generator module for CRC on EDMA data

9.2 Overview

The four-channel enhanced direct memory access (EDMA) controller can transfer data between memories and peripherals, and thus offload these tasks from the CPU. It enables high data transfer rates with minimum CPU intervention, and frees up CPU time. The four EDMA channels enable up to four independent and parallel transfers.

The EDMA controller can move data between SRAM and peripherals, between SRAM locations and directly between peripheral registers. With access to all peripherals, the EDMA controller can handle automatic transfer of data to/from communication modules. The EDMA controller can also read from EEPROM memory.

Data transfers are done in continuous bursts of 1 or 2 bytes. They build block transfers of configurable size from 1 byte to 64KB. Repeat option can be used to repeat once each block transfer for single transactions up to 128KB. Source and destination addressing can be static or incremental. Automatic reload of source and/or destination addresses can be done after each burst or block transfer, or when a transaction is complete. Application software, peripherals, and events can trigger EDMA transfers.

The four EDMA channels have individual configuration and control settings. This includes source, destination, transfer triggers, and transaction sizes. They have individual interrupt settings. Interrupt requests can be generated when a transaction is complete or when the EDMA controller detects an error on an EDMA channel.

To enable flexibility in transfers, channels can be interlinked so that the second takes over the transfer when the first is finished.

The EDMA controller supports extended features such as double buffering, data match for peripherals and data search for SRAM or EEPROM.

The EDMA controller supports two types of channel. Each channel type can be selected individually.

10. Event System

10.1 Features

- System for direct peripheral-to-peripheral communication and signaling
- Peripherals can directly send, receive, and react to peripheral events
  - CPU and EDMA controller independent operation
  - 100% predictable signal timing
  - Short and guaranteed response time
  - Synchronous and asynchronous event routing
- Eight event channels for up to eight different and parallel signal routing and configurations
- Events can be sent and/or used by most peripherals, clock system, and software
- Additional functions include
  - Quadrature decoder with rotary filtering
  - Digital filtering of I/O pin state with configurable filter
  - Simultaneous synchronous and asynchronous events provided to peripheral
- Works in all sleep modes

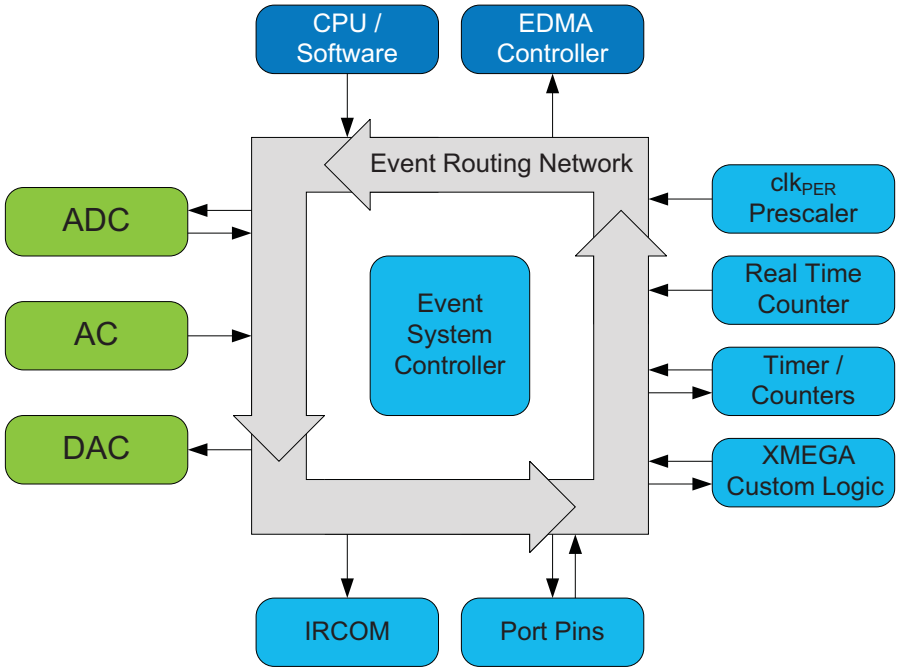
10.2 Overview

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, CPU, or EDMA controller resources, and is thus a powerful tool for reducing the complexity, size and execution time of application code. It allows for synchronized timing of actions in several peripheral modules. The event system enables also asynchronous event routing for instant actions in peripherals.

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 10-1 shows a basic diagram of all connected peripherals. The event system can directly connect together analog and digital converters, analog comparators, I/O port pins, the real-time counter, timer/counters, IR communication module (IRCOM) and XMEGA Custom Logic (programmable logic) block (XCL). It can also be used to trigger EDMA transactions (EDMA controller). Events can also be generated from software and peripheral clock.

Figure 10-1. Event system overview and connected peripherals.



The event routing network consists of eight software-configurable multiplexers that control how events are routed and used. These are called event channels, and allow up to eight parallel event configurations and routing. The maximum routing latency of an external event is two peripheral clock cycles due to re-synchronization, but several peripherals can directly use the asynchronous event without any clock delay. The event system works in all power sleep modes, but only asynchronous events can be routed in sleep modes where the system clock is not available.

11. System Clock and Clock options

11.1 Features

- Fast start-up time
- Safe run-time clock switching
- Internal Oscillators:
  - 32MHz run-time calibrated and tuneable oscillator
  - 8MHz calibrated oscillator with 2MHz output option and fast start-up
  - 32.768kHz calibrated oscillator
  - 32kHz Ultra Low Power (ULP) oscillator with 1kHz output
- External clock options
  - 0.4 - 16MHz Crystal Oscillator
  - 32kHz crystal oscillator with digital correction
  - External clock input in selectable pin location
- PLL with 20 - 128MHz output frequency
  - Internal and external clock options and 1 to 31x multiplication
  - Lock detector
- Clock Prescalers with 1x to 2048x division
- Fast peripheral clocks running at 2 and 4 times the CPU clock frequency
- Automatic Run-Time Calibration of the 32MHz internal oscillator
- External oscillator and PLL lock failure detection with optional non maskable interrupt

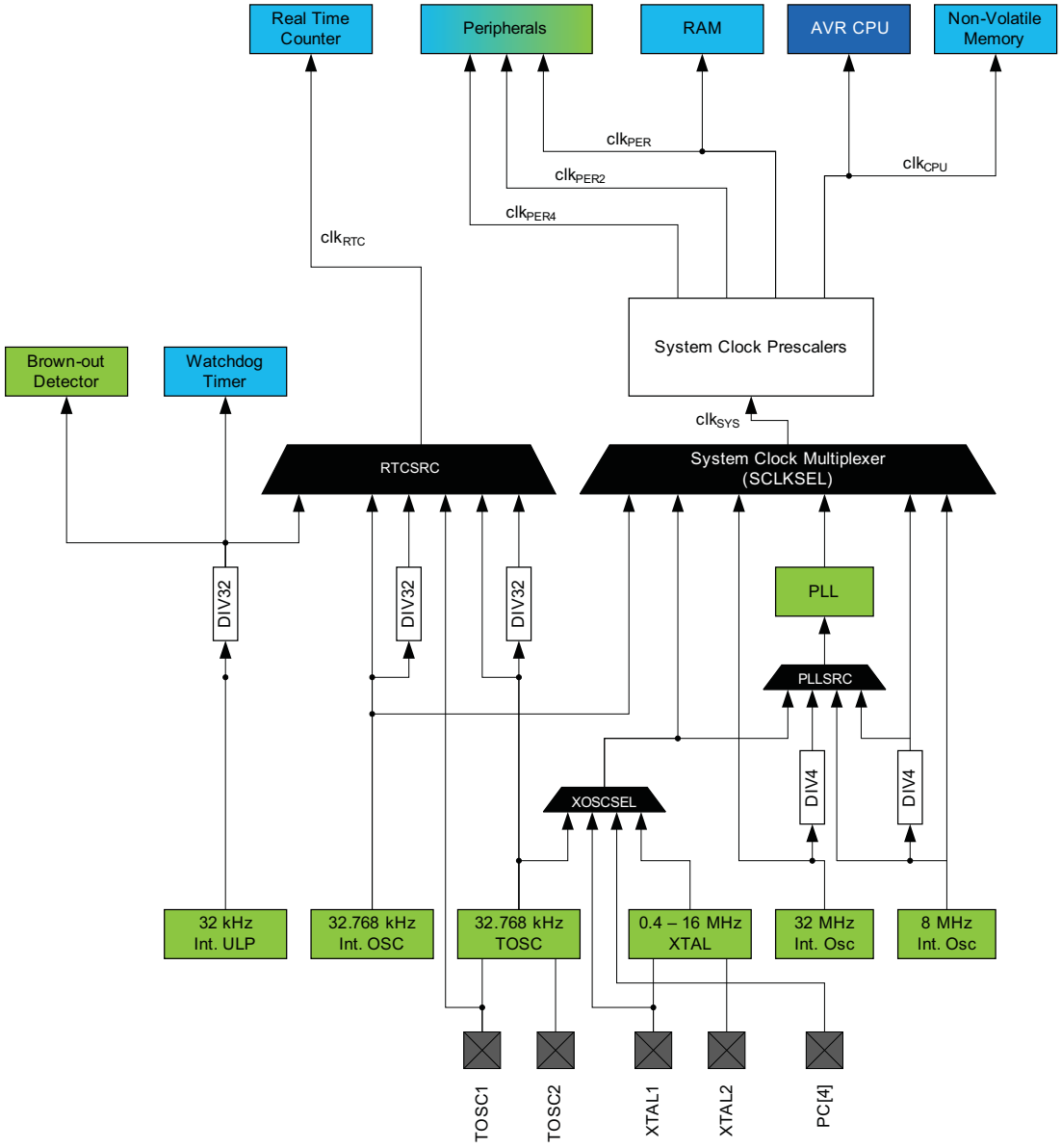
11.2 Overview

Atmel AVR XMEGA E5 devices have a flexible clock system supporting a large number of clock sources. It incorporates both accurate internal oscillators and external crystal oscillator and resonator support. A high-frequency phase locked loop (PLL) and clock prescalers can be used to generate a wide range of clock frequencies. A calibration feature (DFLL) is available, and can be used for automatic run-time calibration of the 32MHz internal oscillator to remove frequency drift over voltage and temperature. An oscillator failure monitor can be enabled to issue a nonmaskable interrupt and switch to the internal oscillator if the external oscillator or PLL fails.

When a reset occurs, all clock sources except the 32kHz ultra low power oscillator are disabled. After reset, the device will always start up running from the 2MHz output of the 8MHz internal oscillator. During normal operation, the system clock source and prescalers can be changed from software at any time.

Figure 11-1 presents the principal clock system in the XMEGA E5 family of devices. Not all of the clocks need to be active at a given time. The clocks for the CPU and peripherals can be stopped using sleep modes and power reduction registers, as described in “Power Management and Sleep Modes” on page 22.

Figure 11-1. The clock system, clock sources and clock distribution.



11.3 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 output of the 8MHz internal oscillator. The other clock sources, DFLL 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.

11.3.1 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.

11.3.2 32.768kHz Calibrated Internal Oscillator

This oscillator provides an approximate 32.768kHz clock. It is calibrated during production to provide a default frequency close to its nominal frequency. The calibration register can also be written from software for run-time calibration of the oscillator frequency. The oscillator employs a built-in prescaler, which provides both a 32.768kHz output and a 1.024kHz output.

11.3.3 32.768kHz Crystal Oscillator

A 32.768kHz crystal oscillator can be connected between the TOSC1 and TOSC2 pins and enables a dedicated low frequency oscillator input circuit. A low power mode with reduced voltage swing on TOSC2 is available. This oscillator can be used as a clock source for the system clock and RTC, and as the DFLL reference clock.

11.3.4 0.4 - 16MHz Crystal Oscillator

This oscillator can operate in four different modes optimized for different frequency ranges, all within 0.4 - 16MHz.

11.3.5 8MHz Calibrated Internal Oscillator

The 8MHz calibrated internal oscillator is the default system clock source after reset. It is calibrated during production to provide a default frequency close to its nominal frequency. The calibration register can also be written from software for run-time calibration of the oscillator frequency. The oscillator employs a built-in prescaler, with 2MHz output. The default output frequency at start-up and after reset is 2MHz. A low power mode option can be used to enable fast system wake-up from power-save mode. In all other modes, the low power mode can be enabled to significantly reduce the power consumption of the internal oscillator.

11.3.6 32MHz Run-time Calibrated Internal Oscillator

The 32MHz run-time calibrated internal oscillator is a high-frequency oscillator. It is calibrated during production to provide a default frequency close to its nominal frequency. A digital frequency looked loop (DFLL) can be enabled for automatic run-time calibration of the oscillator to compensate for temperature and voltage drift and optimize the oscillator accuracy. This oscillator can also be adjusted and calibrated to any frequency between 30MHz and 55MHz.

11.3.7 External Clock Sources

The XTAL1 and XTAL2 pins can be used to drive an external oscillator, either a quartz crystal or a ceramic resonator. XTAL1 or pin 4 of port C (PC4) can be used as input for an external clock signal. The TOSC1 and TOSC2 pins are dedicated to driving a 32.768kHz crystal oscillator.

11.3.8 PLL with 1x-31x Multiplication Factor

The built-in phase locked loop (PLL) can be used to generate a high-frequency system clock. The PLL has a user-selectable multiplication factor of from 1 to 31. In combination with the prescalers, this gives a wide range of output frequencies from all clock sources.

12. Power Management and Sleep Modes

12.1 Features

- Power management for adjusting power consumption and functions
- Five sleep modes
  - Idle
  - Power down
  - Power save
  - Standby
  - Extended standby
- Power reduction register to disable clock and turn off unused peripherals in active and idle modes

12.2 Overview

Various sleep modes and clock gating are provided in order to tailor power consumption to application requirements. This enables the Atmel AVR XMEGA microcontroller to stop unused modules to save power.

All sleep modes are available and can be entered from active mode. In active mode, the CPU is executing application code. When the device enters sleep mode, program execution is stopped and interrupts or a reset is used to wake the device again. The application code decides which sleep mode to enter and when. Interrupts from enabled peripherals and all enabled reset sources can restore the microcontroller from sleep to active mode.

In addition, power reduction registers provide a method to stop the clock to individual peripherals from software. When this is done, the current state of the peripheral is frozen, and there is no power consumption from that peripheral. This reduces the power consumption in active mode and idle sleep modes and enables much more fine-tuned power management than sleep modes alone.

12.3 Sleep Modes

Sleep modes are used to shut down modules and clock domains in the microcontroller in order to save power. XMEGA microcontrollers have five different sleep modes tuned to match the typical functional stages during application execution. A dedicated sleep instruction (SLEEP) is available to enter sleep mode. Interrupts are used to wake the device from sleep, and the available interrupt wake-up sources are dependent on the configured sleep mode. When an enabled interrupt occurs, the device will wake up and execute the interrupt service routine before continuing normal program execution from the first instruction after the SLEEP instruction. If other, higher priority interrupts are pending when the wake-up occurs, their interrupt service routines will be executed according to their priority before the interrupt service routine for the wake-up interrupt is executed. After wake-up, the CPU is halted for four cycles before execution starts.

The content of the register file, SRAM and registers are kept during sleep. If a reset occurs during sleep, the device will reset, start up, and execute from the reset vector.

12.3.1 Idle Mode

In idle mode the CPU and nonvolatile memory are stopped (note that any ongoing programming will be completed), but all peripherals, including the interrupt controller, event system and EDMA controller are kept running. Any enabled interrupt will wake the device.

12.3.2 Power-down Mode

In power-down mode, all clocks, including the real-time counter clock source, are stopped. This allows operation only of asynchronous modules that do not require a running clock. The only interrupts that can wake up the MCU are the two-wire interface address match interrupt and asynchronous port interrupts.

12.3.3 Power-save Mode

Power-save mode is identical to power down, with one exception. If the real-time counter (RTC) is enabled, it will keep running during sleep, and the device can also wake up from either an RTC overflow or compare match interrupt. Low power mode option of 8MHz internal oscillator enables instant oscillator wake-up time. This reduces the MCU wake-up time or enables the MCU wake-up from UART bus.

12.3.4 Standby Mode

Standby mode is identical to power down, with the exception that the enabled system clock sources are kept running while the CPU, peripheral, and RTC clocks are stopped. This reduces the wake-up time. The low power option of 8MHz internal oscillator can be enabled to further reduce the power consumption.

12.3.5 Extended Standby Mode

Extended standby mode is identical to power-save mode, with the exception that the enabled system clock sources are kept running while the CPU and peripheral clocks are stopped. This reduces the wake-up time. The low power option of 8MHz internal oscillator can be enabled to further reduce the power consumption.

13. System Control and Reset

13.1 Features

- Reset the microcontroller and set it to initial state when a reset source goes active
- Multiple reset sources that cover different situations
  - Power-on reset
  - External reset
  - Watchdog reset
  - Brownout reset
  - PDI reset
  - Software reset
- Asynchronous operation
  - No running system clock in the device is required for reset
- Reset status register for reading the reset source from the application code

13.2 Overview

The reset system issues a microcontroller reset and sets the device to its initial state. This is for situations where operation should not start or continue, such as when the microcontroller operates below its power supply rating. If a reset source goes active, the device enters and is kept in reset until all reset sources have released their reset. The I/O pins are immediately tri-stated. The program counter is set to the reset vector location, and all I/O registers are set to their initial values. The SRAM content is kept. However, if the device accesses the SRAM when a reset occurs, the content of the accessed location can not be guaranteed.

After reset is released from all reset sources, the default oscillator is started and calibrated before the device starts running from the reset vector address. By default, this is the lowest program memory address, 0, but it is possible to move the reset vector to the lowest address in the boot section.

The reset functionality is asynchronous, and so no running system clock is required to reset the device. The software reset feature makes it possible to issue a controlled system reset from the user software.

The reset status register has individual status flags for each reset source. It is cleared at power-on reset, and shows which sources have issued a reset since the last power-on.

13.3 Reset Sequence

A reset request from any reset source will immediately reset the device and keep it in reset as long as the request is active. When all reset requests are released, the device will go through three stages before the device starts running again:

- Reset counter delay
- Oscillator startup
- Oscillator calibration

If another reset requests occurs during this process, the reset sequence will start over again.

13.4 Reset Sources

13.4.1 Power-on Reset

A power-on reset (POR) is generated by an on-chip detection circuit. The POR is activated when the V_{CC} rises and reaches the POR threshold voltage (V_{POT}), and this will start the reset sequence.

The POR is also activated to power down the device properly when the V_{CC} falls and drops below the V_{POT} level. The V_{POT} level is higher for falling V_{CC} than for rising V_{CC}. Consult the datasheet for POR characteristics data.

13.4.2 Brownout Detection

The on-chip brownout detection (BOD) circuit monitors the V_{CC} level during operation by comparing it to a fixed, programmable level that is selected by the BODLEVEL fuses. If disabled, BOD is forced on at the lowest level during chip erase and when the PDI is enabled.

13.4.3 External Reset

The external reset circuit is connected to the external RESET pin. The external reset will trigger when the RESET pin is driven below the RESET pin threshold voltage, V_{RST}, for longer than the minimum pulse period, t_{EXT}. The reset will be held as long as the pin is kept low. The RESET pin includes an internal pull-up resistor.

13.4.4 Watchdog Reset

The watchdog timer (WDT) is a system function for monitoring correct program operation. If the WDT is not reset from the software within a programmable timeout period, a watchdog reset will be given. The watchdog reset is active for one to two clock cycles of the 2MHz internal oscillator. For more details see “WDT – Watchdog Timer” on page 26.

13.4.5 Software Reset

The software reset makes it possible to issue a system reset from software by writing to the software reset bit in the reset control register. The reset will be issued within two CPU clock cycles after writing the bit. It is not possible to execute any instruction from when a software reset is requested until it is issued.

13.4.6 Program and Debug Interface Reset

The program and debug interface reset contains a separate reset source that is used to reset the device during external programming and debugging. This reset source is accessible only from external debuggers and programmers.

14. WDT – Watchdog Timer

14.1 Features

- Issues a device reset if the timer is not reset before its timeout period
- Asynchronous operation from dedicated oscillator
- 1kHz output of the 32kHz ultra low power oscillator
- 11 selectable timeout periods, from 8ms to 8s
- Two operation modes:
  - Normal mode
  - Window mode
- Configuration lock to prevent unwanted changes

14.2 Overview

The watchdog timer (WDT) is a system function for monitoring correct program operation. It makes it possible to recover from error situations such as runaway or deadlocked code. The WDT is a timer, configured to a predefined timeout period, and is constantly running when enabled. If the WDT is not reset within the timeout period, it will issue a microcontroller reset. The WDT is reset by executing the WDR (watchdog timer reset) instruction from the application code.

The window mode makes it possible to define a time slot or window inside the total timeout period during which WDT must be reset. If the WDT is reset outside this window, either too early or too late, a system reset will be issued. Compared to the normal mode, this can also catch situations where a code error causes constant WDR execution.

The WDT will run in active mode and all sleep modes, if enabled. It is asynchronous, runs from a CPU-independent clock source, and will continue to operate to issue a system reset even if the main clocks fail.

The configuration change protection mechanism ensures that the WDT settings cannot be changed by accident. For increased safety, a fuse for locking the WDT settings is also available.

15. Interrupts and Programmable Multilevel Interrupt Controller

15.1 Features

- Short and predictable interrupt response time
- Separate interrupt configuration and vector address for each interrupt
- Programmable multilevel interrupt controller
  - Interrupt prioritizing according to level and vector address
  - Three selectable interrupt levels for all interrupts: low, medium and high
  - Selectable, round-robin priority scheme within low-level interrupts
  - Non-maskable interrupts for critical functions
- Interrupt vectors optionally placed in the application section or the boot loader section

15.2 Overview

Interrupts signal a change of state in peripherals, and this can be used to alter program execution. Peripherals can have one or more interrupts, and all are individually enabled and configured. When an interrupt is enabled and configured, it will generate an interrupt request when the interrupt condition is present. The programmable multilevel interrupt controller (PMIC) controls the handling and prioritizing of interrupt requests. When an interrupt request is acknowledged by the PMIC, the program counter is set to point to the interrupt vector, and the interrupt handler can be executed.

All peripherals can select between three different priority levels for their interrupts: low, medium, and high. Interrupts are prioritized according to their level and their interrupt vector address. Medium-level interrupts will interrupt low-level interrupt handlers. High-level interrupts will interrupt both medium- and low-level interrupt handlers. Within each level, the interrupt priority is decided from the interrupt vector address, where the lowest interrupt vector address has the highest interrupt priority. Low-level interrupts have an optional round-robin scheduling scheme to ensure that all interrupts are serviced within a certain amount of time.

Non-maskable interrupts (NMI) are also supported, and can be used for system critical functions.

15.3 Interrupt vectors

The interrupt vector is the sum of the peripheral's base interrupt address and the offset address for specific interrupts in each peripheral. The base addresses for the Atmel AVR XMEGA E5 devices are shown in Table 15-1. Offset addresses for each interrupt available in the peripheral are described for each peripheral in the XMEGA AU manual. For peripherals or modules that have only one interrupt, the interrupt vector is shown in Table 15-1. The program address is the word address.

Table 15-1. Peripheral module address map

Program address (base address)	Source	Interrupt Description
0x0000	RESET	
0x0002	OSCF_INT_vect	Crystal oscillator failure and PLL lock failure interrupt vector (NMI)
0x0004	PORTR_INT_vect	Port R Interrupt vector
0x0006	EDMA_INT_base	EDMA Controller Interrupt base
0x000E	RTC_INT_base	Real time counter interrupt base
0x0012	PORTC_INT_vect	Port C interrupt vector
0x0014	TWIC_INT_base	Two-wire interface on Port C interrupt base

Program address (base address)	Source	Interrupt Description
0x0018	TCC4_INT_base	Timer/counter 4 on port C interrupt base
0x0024	TCC5_INT_base	Timer/counter 5 on port C interrupt base
0x002C	SPIC_INT_vect	SPI on port C interrupt vector
0x002E	USARTC0_INT_base	USART 0 on port C interrupt base
0x0034	NVM_INT_base	Non-Volatile Memory interrupt base
0x0038	XCL_INT_base	XCL (programmable logic) module interrupt base
0x003C	PORTA_INT_vect	Port A interrupt vector
0x003E	ACA_INT_base	Analog comparator on Port A interrupt base
0x0044	ADCA_INT_base	Analog to digital converter on Port A interrupt base
0x0046	PORTD_INT_vect	Port D interrupt vector
0x0048	TCD5_INT_base	Timer/counter 5 on port D interrupt base
0x0050	USARTD0_INT_base	USART 0 on port D interrupt base

16. I/O Ports

16.1 Features

- 26 general purpose input and output pins with individual configuration
- Output driver with configurable driver and pull settings:
  - Totem-pole
  - Wired-AND
  - Wired-OR
  - Bus-keeper
  - Inverted I/O
- Input with asynchronous sensing with interrupts and events
  - Sense both edges
  - Sense rising edges
  - Sense falling edges
  - Sense low level
- Optional pull-up and pull-down resistor on input and Wired-OR/AND configurations
- Optional slew rate control per I/O port
- Asynchronous pin change sensing that can wake the device from all sleep modes
- One port interrupt with pin masking per I/O port
- Efficient and safe access to port pins
  - Hardware read-modify-write through dedicated toggle/clear/set registers
  - Configuration of multiple pins in a single operation
  - Mapping of port registers into bit-accessible I/O memory space
- Peripheral clocks output on port pin
- Real-time counter clock output to port pin
- Event channels can be output on port pin
- Remapping of digital peripheral pin functions
  - Selectable USART and timer/counters input/output pin locations
  - Selectable Analog Comparator output pin locations

16.2 Overview

One port consists of up to 8 pins ranging from pin 0 to 7. Each port pin can be configured as input or output with configurable driver and pull settings. They also implement asynchronous input sensing with interrupt and events for selectable pin change conditions.

Asynchronous pin-change sensing means that a pin change can wake the device from all sleep modes, including the modes where no clocks are running.

All functions are individual and configurable per pin, but several pins can be configured in a single operation. The pins have hardware read-modify-write (RMW) functionality for safe and correct change of drive value and/or pull resistor configuration. The direction of one port pin can be changed without unintentionally changing the direction of any other pin.

The port pin configuration also controls input and output selection of other device functions. It is possible to have both the peripheral clock and the real-time clock output to a port pin, and available for external use. The same applies to events from the event system that can be used to synchronize and control external functions. Other digital peripherals, such as USART, timer/counters, and analog comparator output can be remapped to selectable pin locations in order to optimize pin-out versus application needs.

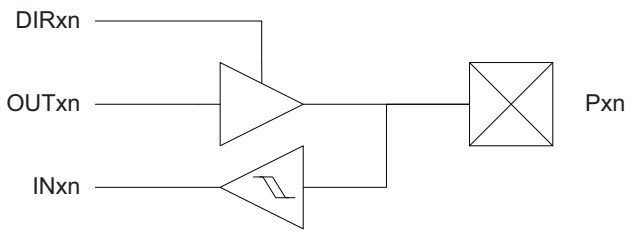
The notations of the ports are PORTA, PORTC, PORTD, and PORTR.

16.3 Output Driver

All port pins (Pxn) have programmable output configuration. The port pins also have configurable slew rate limitation to reduce electromagnetic emission.

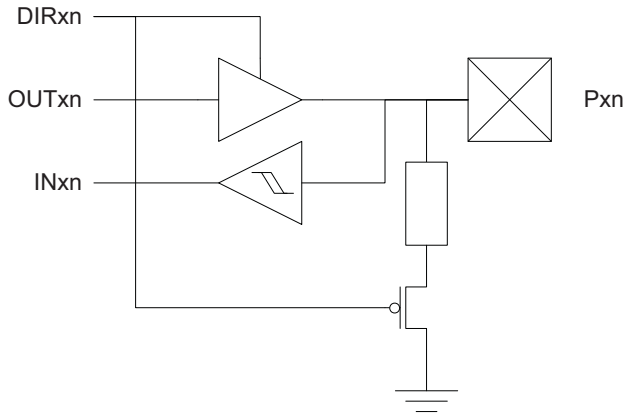
16.3.1 Push-pull

Figure 16-1. I/O configuration - Totem-pole.



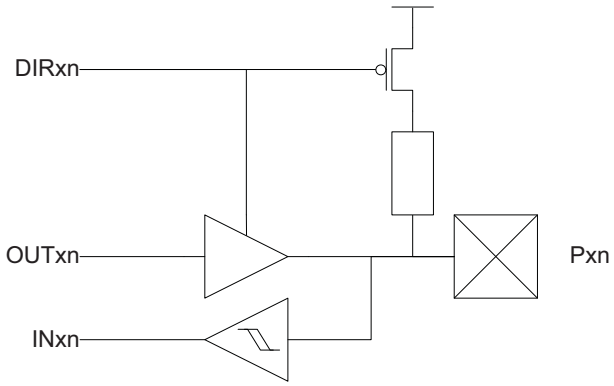
16.3.2 Pull-down

Figure 16-2. I/O configuration - Totem-pole with pull-down (on input).



16.3.3 Pull-up

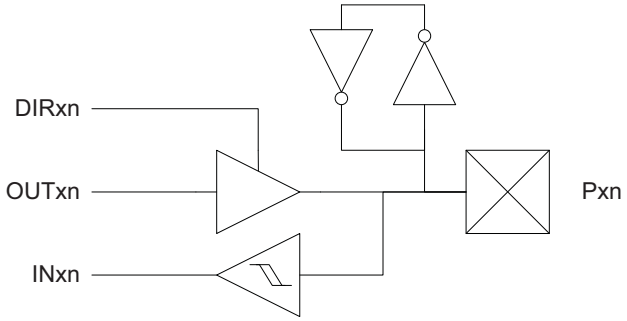
Figure 16-3. I/O configuration - Totem-pole with pull-up (on input).



16.3.4 Bus-keeper

The bus-keeper's weak output produces the same logical level as the last output level. It acts as a pull-up if the last level was '1', and pull-down if the last level was '0'.

Figure 16-4. I/O configuration - Totem-pole with bus-keeper.



16.3.5 Others

Figure 16-5. Output configuration - Wired-OR with optional pull-down.

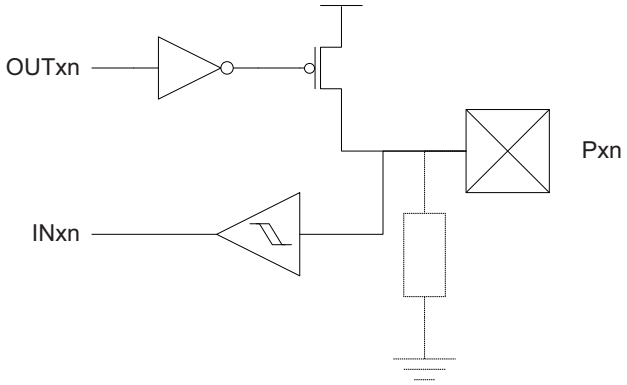
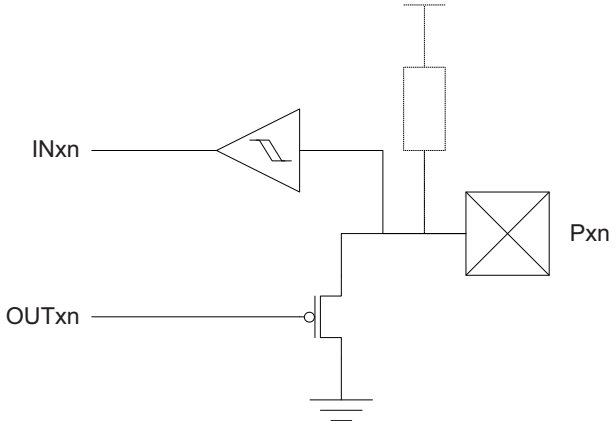


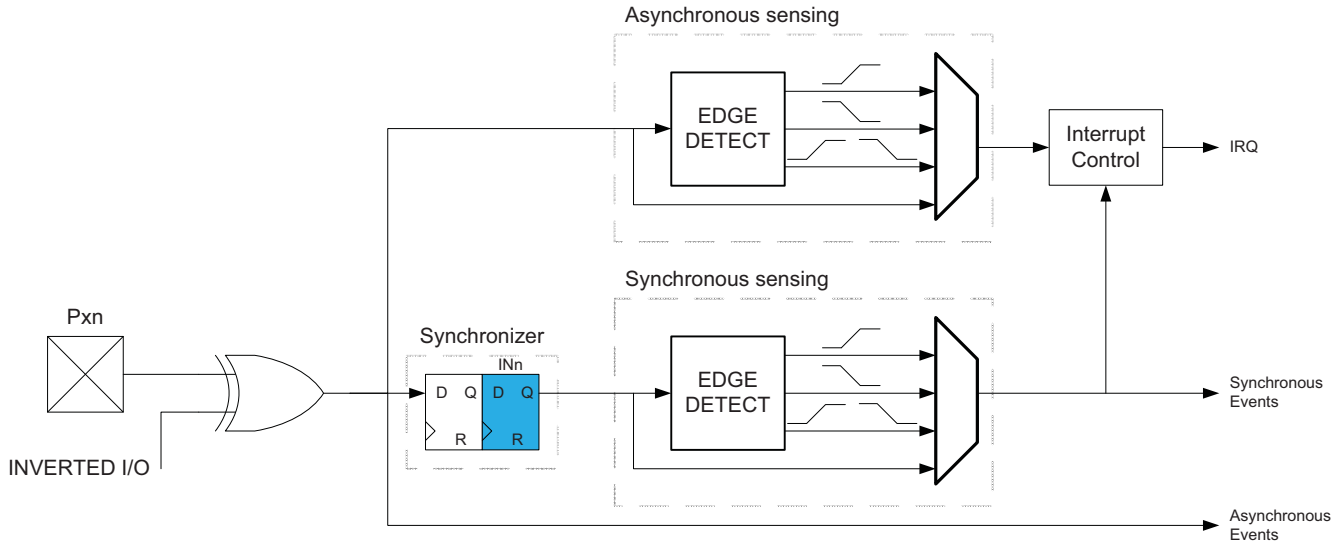
Figure 16-6. I/O configuration - Wired-AND with optional pull-up.



16.4 Input sensing

Input sensing is synchronous or asynchronous depending on the enabled clock for the ports, and the configuration is shown in Figure 16-7.

Figure 16-7. Input sensing system overview.



When a pin is configured with inverted I/O, the pin value is inverted before the input sensing.

16.5 Alternate Port Functions

Most port pins have alternate pin functions in addition to being a general purpose I/O pin. When an alternate function is enabled, it might override the normal port pin function or pin value. This happens when other peripherals that require pins are enabled or configured to use pins. If and how a peripheral will override and use pins is described in the section for that peripheral. “Pinout and Pin Functions” on page 57 shows which modules on peripherals that enable alternate functions on a pin, and which alternate functions that are available on a pin.

17. Timer Counter type 4 and 5

17.1 Features

- Three 16-bit timer/counter
  - One timer/counter of type 4
  - Two timer/counter of type 5
- 32-bit timer/counter support by cascading two timer/counters
- Up to four compare or capture (CC) channels
  - Four CC channels for timer/counters of type 4
  - Two CC channels for timer/counters of type 5
- Double buffered timer period setting
- Double buffered CC channels
- Waveform generation modes:
  - Frequency generation
  - Single-slope pulse width modulation
  - Dual-slope pulse width modulation
- Input capture:
  - Input capture with noise cancelling
  - Frequency capture
  - Pulse width capture
  - 32-bit input capture
- Timer overflow and error interrupts/events
- One compare match or input capture interrupt/event per CC channel
- Can be used with event system for:
  - Quadrature decoding
  - Count and direction control
  - Input capture
- Can be used with EDMA and to trigger EDMA transactions
- High-resolution extension
  - Increases frequency and waveform resolution by 4x (2-bit) or 8x (3-bit)
- Waveform extension
  - Low- and high-side output with programmable dead-time insertion (DTI)
- Fault extention
  - Event controlled fault protection for safe disabling of drivers

17.2 Overview

Atmel AVR XMEGA devices have a set of flexible, 16-bit timer/counters (TC). Their capabilities include accurate program execution timing, frequency and waveform generation, and input capture with time and frequency measurement of digital signals. Two timer/counters can be cascaded to create a 32-bit timer/counter with optional 32-bit input capture.

A timer/counter consists of a base counter and a set of compare or capture (CC) channels. The base counter can be used to count clock cycles or events. It has direction control and period setting that can be used for timing. The CC channels can be used together with the base counter to do compare match control, frequency generation, and pulse width modulation (PWM) generation, as well as various input capture operations. A timer/counter can be configured for either capture, compare, or capture and compare function.

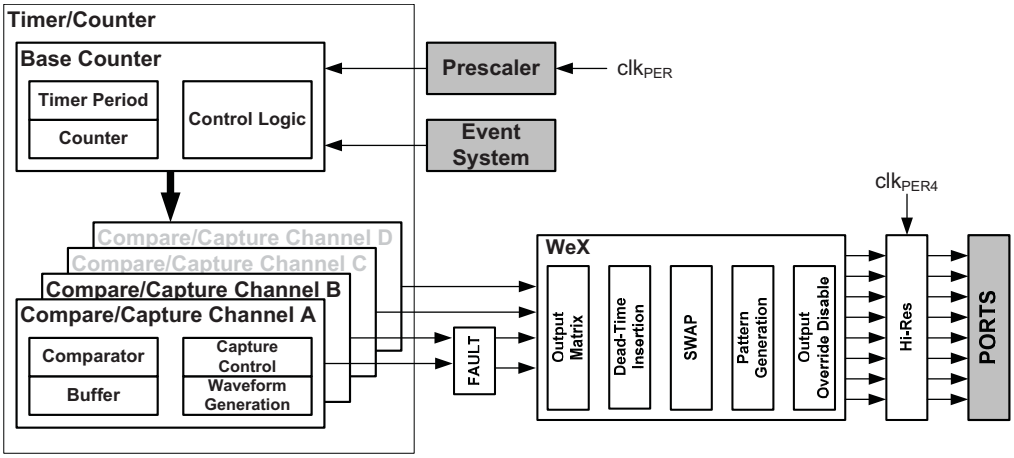
A timer/counter can be clocked and timed from the peripheral clock with optional prescaling, or from the event system. The event system can also be used for direction control, input capture trigger or to synchronize operations.

There are two differences between timer/counter type 4 and type 5. Timer/counter 4 has four CC channels, and timer/counter 5 has two CC channels. Both timer/counter 4 and 5 can be set in 8-bit mode, allowing the application to double the number of compare and capture channels that then get 8-bit resolution.

Some timer/counters have extensions that enable more specialized waveform generation. The waveform extension (WeX) is intended for motor control, ballast, LED, H-bridge, power converters, and other types of power control applications. It enables more customized waveform output distribution, and low- and high-side channel output with optional dead-time insertion. It can also generate a synchronized bit pattern across the port pins. The high-resolution (hi-res) extension can increase the waveform resolution by four or eight times by using an internal clock source four times faster than the peripheral clock. The fault extension (FAULT) enables fault protection for safe and deterministic handling, disabling and/or shut down of external drivers.

A block diagram of the 16-bit timer/counter with extensions and closely related peripheral modules (in grey) is shown in Figure 17-1.

Figure 17-1. 16-bit timer/counter and closely related peripherals.



PORTC has one timer/counter 4 and one timer/counter 5. PORTD has one timer/counter 5. Notation of these are TCC4 (timer/counter C4), TCC5 and TCD5 respectively.

18. WeX – Waveform Extension

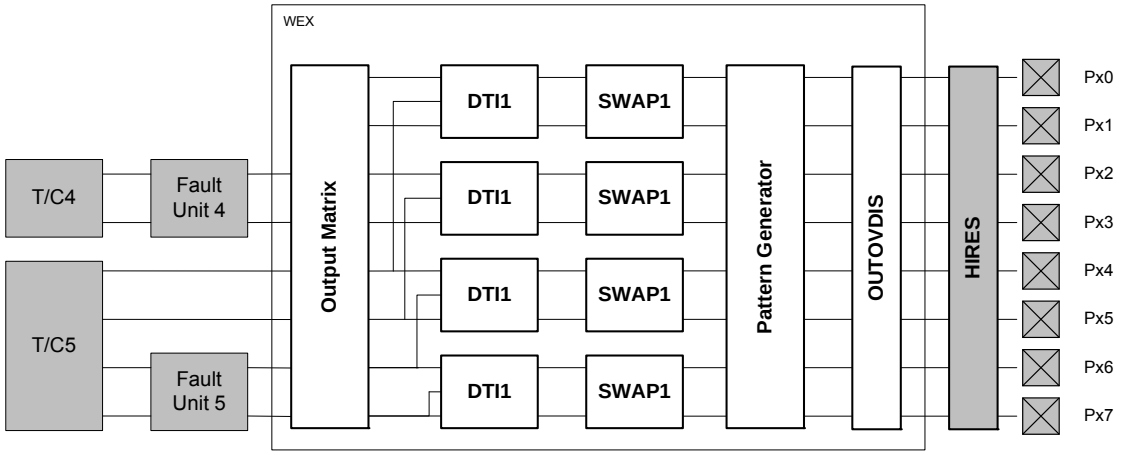
18.1 Features

- Module for more customized and advanced waveform generation
  - Optimized for various type of motor, ballast, and power stage control
- Output matrix for timer/counter waveform output distribution
  - Configurable distribution of compare channel output across port pins
  - Redistribution of dead-time insertion resource between TC4 and TC5.
- Four dead-time insertion (DTI) units, each with
  - Complementary high and low side with non overlapping outputs
  - Separate dead-time setting for high and low side
  - 8-bit resolution
- Four swap (SWAP) units
  - Separate port pair or low high side drivers swap
  - Double buffered swap feature
- Pattern generation creating synchronized bit pattern across the port pins
  - Double buffered pattern generation

18.2 Overview

The waveform extension (WEX) provides extra functions to the timer/counter in waveform generation (WG) modes. It is primarily intended for motor control, ballast, LED, H-bridge, power converters, and other types of power control applications. The WEX consist of five independent and successive units, as shown in Figure 18-1.

Figure 18-1. Waveform extension and closely related peripherals.



The output matrix (OTMX) can distribute and route out the waveform outputs from timer/counter 4 and 5 across the port pins in different configurations, each optimized for different application types. The dead time insertion (DTI) unit splits the four lower OTMX outputs into a two non-overlapping signals, the non-inverted low side (LS) and inverted high side (HS) of the waveform output with optional dead-time insertion between LS and HS switching.

The swap (SWAP) unit can swap the LS and HS pin position. This can be used for fast decay motor control. The pattern generation unit generates synchronized output waveform with constant logic level. This can be used for easy stepper motor and full bridge control.

The output override disable unit can disable the waveform output on selectable port pins to optimize the pins usage. This is to free the pins for other functional use, when the application does not need the waveform output spread across all the port pins as they can be selected by the OTMX configurations.

The waveform extension is available for TCC4 and TCC5. The notation of this is WEXC.

19. Hi-Res – High Resolution Extension

19.1 Features

- Increases waveform generator resolution up to 8x (three bits)
- Supports frequency, single-slope PWM, and dual-slope PWM generation
- Supports the WeX when this is used for the same timer/counter

19.2 Overview

The high-resolution (hi-res) extension can be used to increase the resolution of the waveform generation output from a timer/counter by four or eight. It can be used for a timer/counter doing frequency, single-slope PWM, or dual-slope PWM generation. It can also be used with the WeX if this is used for the same timer/counter.

The hi-res extension uses the peripheral 4x clock (ClkPER4). The system clock prescalers must be configured so the peripheral 4x clock frequency is four times higher than the peripheral and CPU clock frequency when the hi-res extension is enabled.

There is one hi-res extension that can be enabled for timer/counters pair on PORTC. The notation of this is HIRESC.

20. Fault Extension

20.1 Features

- Connected to timer/counter output and waveform extension input
- Event controlled fault protection for instant and predictable fault triggering
- Fast, synchronous and asynchronous fault triggering
- Flexible configuration with multiple fault sources
- Recoverable fault modes
  - Restart or halt the timer/counter on fault condition
  - Timer/counter input capture on fault condition
  - Waveform output active time reduction on fault condition
- Non-recoverable faults
  - Waveform output is forced to a pre-configured safe state on fault condition
  - Optional fuse output value configuration defining the output state during system reset
- Flexible fault filter selections
  - Digital filter to prevent false triggers from I/O pin glitches
  - Fault blanking to prevent false triggers during commutation
  - Fault input qualification to filter the fault input during the inactive output compare states

20.2 Overview

The fault extension enables event controlled fault protection by acting directly on the generated waveforms from timer/counter compare outputs. It can be used to trigger two types of faults with the following actions:

- Recoverable faults: the timer/counter can be restarted or halted as long as the fault condition is preset. The compare output pulse active time can be reduced as long as the fault condition is preset. This is typically used for current sensing regulation, zero crossing re-triggering, demagnetization re-triggering, and so on.
- Non-recoverable faults: the compare outputs are forced to a safe and pre-configured values that are safe for the application. This is typically used for instant and predictable shut down and to disable the high current or voltage drivers.

Events are used to trigger a fault condition. One or several simultaneous events are supported, both synchronously or asynchronously. By default, the fault extension supports asynchronous event operation, ensuring predictable and instant fault reaction, including system power modes where the system clock is stopped.

By using the input blanking, the fault input qualification or digital filter option in event system, the fault sources can be filtered to avoid false faults detection.

There are two fault extensions, one for each of the timer/counter 4 and timer/counter 5 on PORTC. The notation of these are FAULTC4 and FAULTC5, respectively.

21. RTC – 16-bit Real-Time Counter

21.1 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
- Correction for external crystal oscillator frequency error down to ±0.5 ppm accuracy

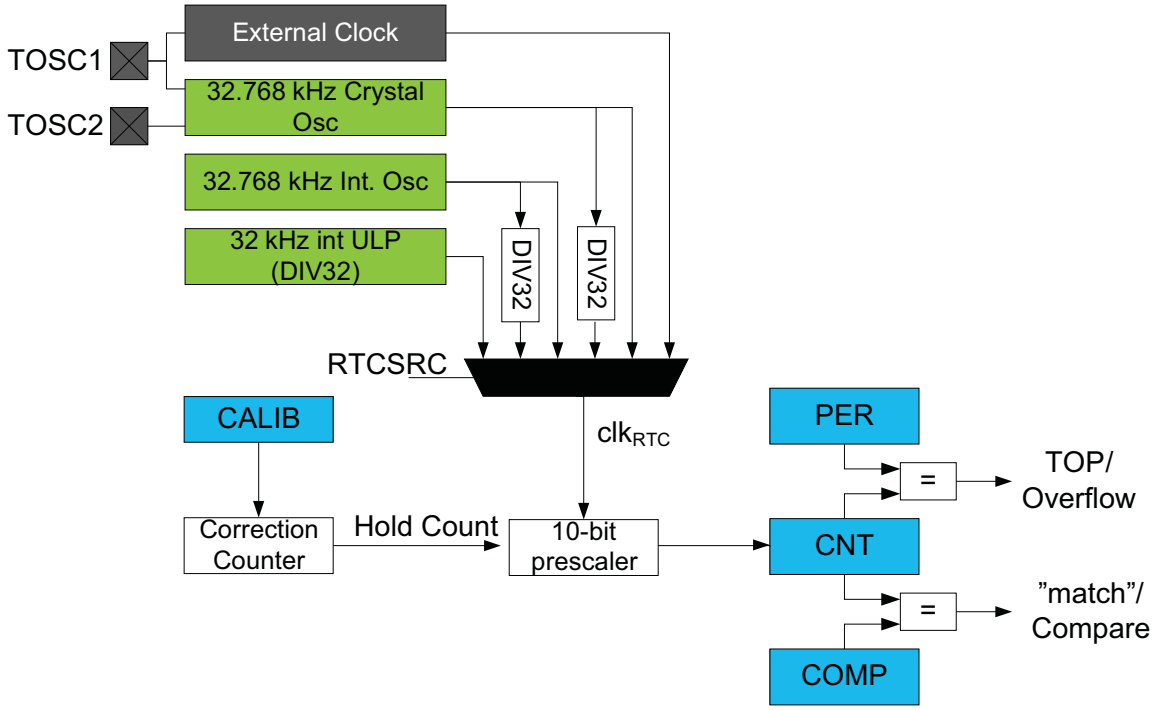
21.2 Overview

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 30.5µs, 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 21-1. Real-time counter overview.



The RTC also supports correction when operated using external 32.768 kHz crystal oscillator. An externally calibrated value will be used for correction. The calibration can be done by measuring the default RTC frequency relative to a more accurate clock input to the device as system clock. The RTC can be calibrated to an accuracy of ±0.5 PPM. The RTC correction operation will either speed up (by skipping count) or slow down (adding extra cycles) the prescaler to account for the crystal oscillator error.

22. TWI – Two Wire Interface

22.1 Features

- One two-wire interface
  - Phillips I^2C 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
  - Bridge mode with independent and simultaneous master and slave operation
- 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, 400kHz and 1MHz 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)
- Supports SMBUS Layer 1 timeouts
- Configurable timeout values
- Independent timeout counters in master and slave (Bridge mode support)

22.2 Overview

The two-wire interface (TWI) is a bidirectional, two-wire communication interface. It is I^2C 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. One bus can have many slaves and one or several masters that can take control of the bus.

The TWI module supports master and slave functionality. The master and slave functionality are separated from each other, and can be enabled and operate simultaneously and separately. The master module supports multi-master bus operation and arbitration. It contains the baud rate generator. Quick command and smart mode can be enabled to auto-trigger operations and reduce software complexity. The master can support 100kHz, 400kHz and 1MHz bus frequency.

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. By using the bridge option, the slave can be mapped to different pin locations. The master and slave can support 100kHz, 400kHz and 1MHz bus frequency.

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.

It is also possible to enable the bridge mode. In this mode, the slave I/O pins are selected from an alternative port, enabling independent and simultaneous master and slave operation.

PORTC has one TWI. Notation of this peripheral is TWIC. Alternative TWI Slave location in bridge mode is on PORTD.

23. SPI – Serial Peripheral Interface

23.1 Features

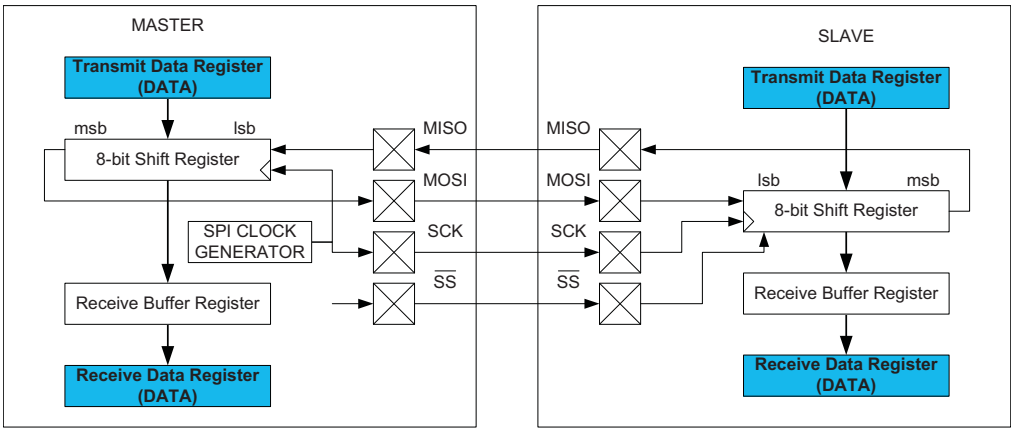
- One SPI peripheral
- Full-duplex, three-wire synchronous data transfer
- Master or slave operation
- Lsb first or msb first data transfer
- Eight programmable bit rates
- Interrupt flag at the end of transmission
- Write collision flag to indicate data collision
- Wake up from idle sleep mode
- Double speed master mode

23.2 Overview

The Serial Peripheral Interface (SPI) is a high-speed, full duplex, synchronous data transfer interface using three or four pins. It allows fast communication between an AVR XMEGA device and peripheral devices or between several microcontrollers.

A device connected to the bus must act as a master or slave. The master initiates and controls all data transactions. The interconnection between master and slave devices with SPI is shown in Figure 23-1. The system consists of two shift registers and a clock generator. The SPI master initiates the communication by pulling the slave select (SS) signal low for the desired slave. Master and slave prepare the data to be sent in their respective shift registers, and the master generates the required clock pulses on the SCK line to interchange data. Data are always shifted from master to slave on the master output, slave input (MOSI) line, and from slave to master on the master input, slave output (MISO) line. After each data packet, the master can synchronize the slave by pulling the $\overline{SS}$ line high.

Figure 23-1. SPI master-slave interconnection



By default, the SPI module is single buffered and transmit direction and double buffered in the receive direction. A byte written to the transmit data register will be copied to the shift register when a full character has been received. When receiving data, a received character must be read from the transmit data register before the third character has been completely shifted in to avoid losing data. Optionally, buffer modes can be enabled. When used, one buffer is available for transmitter and a double buffer for reception.

PORTC has one SPI. Notation of this is SPIC

24. USART

24.1 Features

- Two identical USART peripherals
- Full-duplex or one-wire half-duplex operation
- Asynchronous or synchronous operation
  - Synchronous clock rates up to 1/2 of the device clock frequency
  - Asynchronous clock rates up to 1/8 of the device clock frequency
- Supports serial frames with:
  - 5, 6, 7, 8, or 9 data bits
  - Optionally even and odd parity bits
  - 1 or 2 stop bits
- Fractional baud rate generator
  - Can generate desired baud rate from any system clock frequency
  - No need for external oscillator with certain frequencies
- Built-in error detection and correction schemes
  - Odd or even parity generation and parity check
  - Data overrun and framing error detection
  - Noise filtering includes false start bit detection and digital low-pass filter
- Separate interrupts for
  - Transmit complete
  - Transmit data register empty
  - Receive complete
- Multiprocessor communication mode
  - Addressing scheme to address a specific devices on a multidevice bus
  - Enable unaddressed devices to automatically ignore all frames
- System wake-up from Start bit
- Master SPI mode
  - Double buffered operation
  - Configurable data order
  - Operation up to 1/2 of the peripheral clock frequency
- IRCOM module for IrDA compliant pulse modulation/demodulation
- One USART is connected to XMEGA Custom Logic (XCL) module:
  - Extend serial frame length up to 256 bit by using the peripheral counter
  - Modulate/demodulate data within the frame by using the glue logic outputs

24.2 Overview

The universal synchronous and asynchronous serial receiver and transmitter (USART) is a fast and flexible serial communication module. The USART supports full-duplex with asynchronous and synchronous operation and single wire half-duplex communication with asynchronous operation. The USART can be configured to operate in SPI master mode and used for SPI communication.

Communication is frame based, and the frame format can be customized to support a wide range of standards. The USART is buffered in both directions, enabling continued data transmission without any delay between frames. Separate interrupts for receive and transmit complete enable fully interrupt driven communication. Frame error and buffer overflow are detected in hardware and indicated with separate status flags. Even or odd parity generation and parity check can also be enabled.

In one-wire configuration, the TxD pin is connected to the RxD pin internally, limiting the IO pins usage. If the receiver is enabled when transmitting, it will receive what the transmitter is sending. This mode can be used for bit error detection.

The clock generator includes a fractional baud rate generator that is able to generate a wide range of USART baud rates from any system clock frequencies. This removes the need to use an external crystal oscillator with a specific frequency to achieve a required baud rate. It also supports external clock input in synchronous slave operation.

An IRCOM module can be enabled for one USART to support IrDA 1.4 physical compliant pulse modulation and demodulation for baud rates up to 115.2Kbps.

One USART can be connected to the XMEGA Custom Logic module (XCL). When used with the XCL, the data length within an USART/SPI frame can be controlled by the peripheral counter (PEC) within the XCL. This enables configurable frame length up to 256 bits. In addition, the TxD/RxD data can be encoded/decoded before the signal is fed into the USART receiver, or after the signal is output from transmitter when the USART is connected to XCL LUT outputs.

When the USART is set in master SPI mode, all USART-specific logic is disabled, leaving the transmit and receive buffers, shift registers, and baud rate generator enabled. The registers are used in both modes, but their functionality differs for some control settings. Pin control and interrupt generation are identical in both modes.

PORTC and PORTD each has one USART. Notation of these peripherals are USARTC0 and USARTD0, respectively.

25. IRCOM – IR Communication Module

25.1 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

25.2 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.

26. XCL – XMEGA Custom Logic Module

26.1 Features

- Two independent 8-bit timer/counter with:
  - Period and compare channel for each timer/counter
  - Input Capture for each timer
  - Serial peripheral data length control for each timer
  - Timeout support for each timer
  - Timer underflow interrupt/event
  - Compare match or input capture interrupt/event for each timer
- One 16-bit timer/counter by cascading two 8-bit timer/counters with:
  - Period and compare channel
  - Input capture
  - Timeout support
  - Timer underflow interrupt/event
  - Compare match or input capture interrupt/event
- Programmable lookup table supporting multiple configurations:
  - Two 2-input units
  - One 3-input unit
  - RS configuration
  - Duplicate input with selectable delay on one input or output
  - Connection to external I/O pins, event system or one selectable USART
- Combinatorial Logic Functions using programmable truth table:
  - AND, NAND, OR, NOR, XOR, XNOR, NOT, MUX
- Sequential Logic Functions:
  - D-Flip-Flop, D Latch, RS Latch
- Input sources:
  - From external pins or the event system
  - One input source includes selectable delay or synchronizing option
  - Can be shared with selectable USART pin locations
- Outputs:
  - Available on external pins or event system
  - Includes selectable delay or synchronizing option
  - Can override selectable USART pin locations
- Operates in active mode and all sleep modes

26.2 Overview

The XMEGA Custom Logic module (XCL) consists of two sub-units, each including 8-bit timer/counter with flexible settings, peripheral counter working with one software selectable USART module, delay elements, glue logic with programmable truth table and a global logic interconnect array.

The timer/counter configuration allows for two 8-bits timer/counters. Each timer/counter supports normal, compare and input capture operation, with common flexible clock selections and event channels for each timer. By cascading the two 8-bit timer/counters, the XCL can be used as a 16-bit timer/counter.

The peripheral counter (PEC) configuration, the XCL is connected to one software selectable USART. This USART controls the counter operation, and the PEC can optionally control the data length within the USART frame.

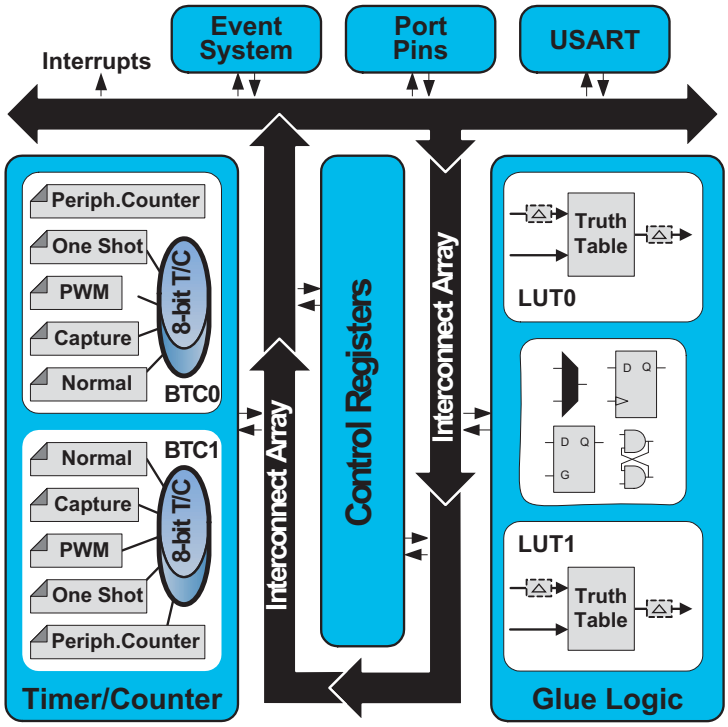
The glue logic configuration, the XCL implements two programmable lookup tables (LUTs). Each defines the truth table corresponding to the logical condition between two inputs. Any combinatorial function logic is possible. The LUT inputs can be connected to I/O pins or event system channels. If the LUT is connected to the USART0 pin locations, the data

lines (TXD/RXD) data encoding/decoding will be possible. Connecting together the LUT units, RS Latch or any combinatorial logic between two operands or three inputs can be enabled.

The LUT works in all sleep modes. Combined with event system and one I/O pin, the LUT can wake-up the system if, and only if, condition on up to 3 input pins is true.

A block diagram of the programmable logic unit with extensions and closely related peripheral modules (in grey) is shown in Figure 26-1.

Figure 26-1. XMEGA custom logic module and closely related peripherals.



27. CRC – Cyclic Redundancy Check Generator

27.1 Features

- Cyclic redundancy check (CRC) generation and checking for
  - Communication data
  - Program or data in flash memory
  - Data in SRAM and I/O memory space
- Integrated with flash memory, EDMA controller and CPU
  - Continuous CRC on data going through an EDMA channel
  - Automatic CRC of the complete or a selectable range of the flash memory
  - CPU can load data to the CRC generator through the I/O interface
- CRC polynomial software selectable to
  - CRC-16 (CRC-CCITT)
  - CRC-32 (IEEE 802.3)
- Zero remainder detection

27.2 Overview

A cyclic redundancy check (CRC) is an error detection technique test algorithm used to find accidental errors in data, and it is commonly used to determine the correctness of a data transmission, and data present in the data and program memories. A CRC takes a data stream or a block of data as input and generates a 16- or 32-bit output that can be appended to the data and used as a checksum. When the same data are later received or read, the device or application repeats the calculation. If the new CRC result does not match the one calculated earlier, the block contains a data error. The application will then detect this and may take a corrective action, such as requesting the data to be sent again or simply not using the incorrect data.

Typically, an n-bit CRC applied to a data block of arbitrary length will detect any single error burst not longer than n bits (any single alteration that spans no more than n bits of the data), and will detect the fraction $1-2^{-n}$ of all longer error bursts. The CRC module in XMEGA devices supports two commonly used CRC polynomials; CRC-16 (CRC-CCITT) and CRC-32 (IEEE 802.3).

- CRC-16:
  - Polynomial: $x^{16} + x^{12} + x^5 + 1$
  - Hex Value: 0x1021
- CRC-32:
  - Polynomial: $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$
  - Hex Value: 0x04C11DB7

28. ADC – 12-bit Analog to Digital Converter

28.1 Features

- 12-bit resolution
- Up to 300 thousand samples per second
  - Down to 2.3µs conversion time with 8-bit resolution
  - Down to 3.35µs conversion time with 12-bit resolution
- Differential and single-ended input
  - Up to 16 single-ended inputs
  - 16x8 differential inputs with optional gain
- Built-in differential gain stage
  - 1/2x, 1x, 2x, 4x, 8x, 16x, 32x, and 64x gain options
- Single, continuous and scan conversion options
- Four internal inputs
  - Internal temperature sensor
  - DAC output
  - V_{CC} voltage divided by 10
  - 1.1V bandgap voltage
- Internal and external reference options
- Compare function for accurate monitoring of user defined thresholds
- Offset and gain correction
- Averaging
- Over-sampling and decimation
- Optional event triggered conversion for accurate timing
- Optional interrupt/event on compare result
- Optional EDMA transfer of conversion results

28.2 Overview

The ADC converts analog signals to digital values. The ADC has 12-bit resolution and is capable of converting up to 300 thousand samples per second (ksps). The input selection is flexible, and both single-ended and differential measurements can be done. For differential measurements, an optional gain stage is available to increase the dynamic range. In addition, several internal signal inputs are available. The ADC can provide both signed and unsigned results.

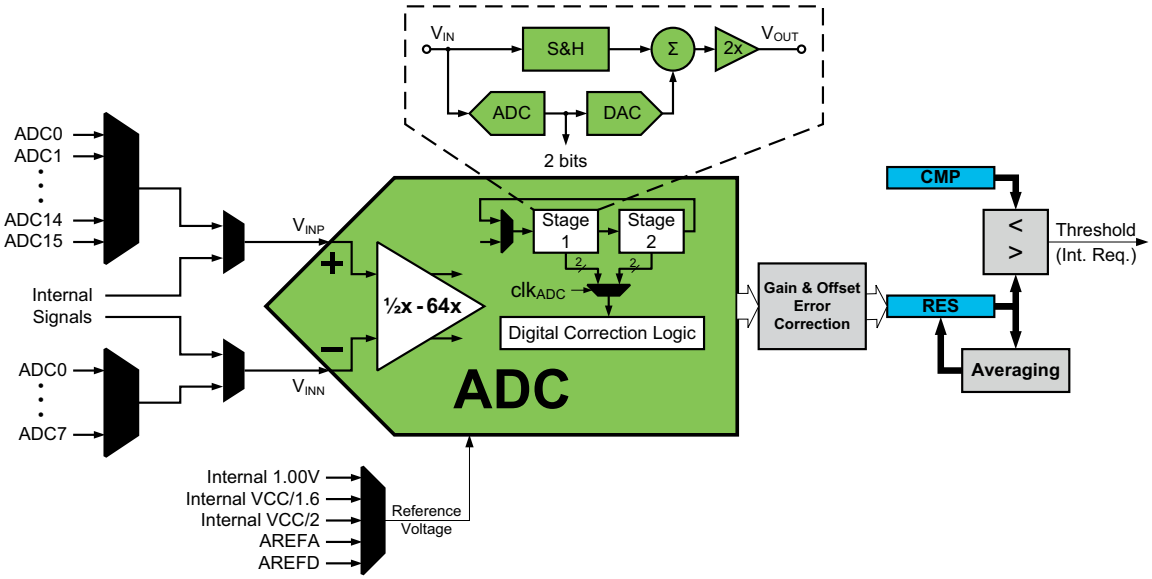
The ADC measurements can either be started by application software or an incoming event from another peripheral in the device. The ADC measurements can be started with predictable timing, and without software intervention. It is possible to use EDMA to move ADC results directly to memory or peripherals when conversions are done.

Both internal and external reference voltages can be used. An integrated temperature sensor is available for use with the ADC. The output from the DAC, VCC/10 and the bandgap voltage can also be measured by the ADC.

The ADC has a compare function for accurate monitoring of user defined thresholds with minimum software intervention required.

When operation in noisy conditions, the average feature can be enabled to increase the ADC resolution. Up to 1024 samples can be averaged, enabling up to 16-bit resolution results. In the same way, using the over-sampling and decimation mode, the ADC resolution is increased up to 16-bits, which results in up to 4-bit extra LSB resolution. The ADC includes various calibration options. In addition to standard production calibration, the user can enable the offset and gain correction to improve the absolute ADC accuracy.

Figure 28-1. ADC overview.



The ADC may be configured for 8- or 12-bit result, reducing the propagation delay from 3.35µs for 12-bit to 2.3µs for 8-bit result. ADC conversion results are provided left- or right adjusted with eases calculation when the result is represented as a signed.

PORTA has one ADC. Notation of this peripheral is ADCA.

29. DAC – Digital to Analog Converter

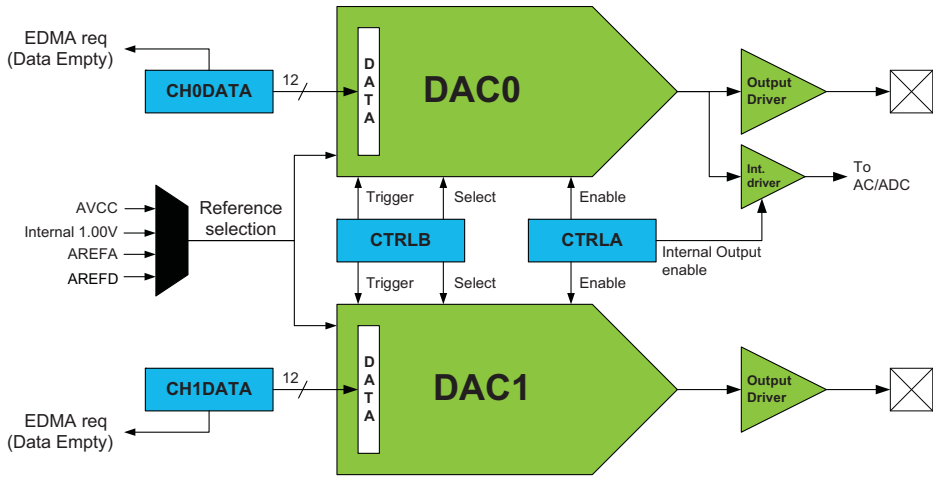
29.1 Features

- One Digital to Analog Converter (DAC)
- 12-bit resolution
- Two independent, continuous-drive output channels
- Up to 1 million samples per second conversion rate per DAC channel
- Built-in calibration that removes:
  - Offset error
  - Gain error
- Multiple conversion trigger sources
  - On new available data
  - Events from the event system
- Drive capabilities and support for
  - Resistive loads
  - Capacitive loads
  - Combined resistive and capacitive loads
- Internal and external reference options
- DAC output available as input to analog comparator and ADC
- Low-power mode, with reduced drive strength
- Optional EDMA transfer of data

29.2 Overview

The digital-to-analog converter (DAC) converts digital values to voltages. The DAC has two channels, each with 12-bit resolution, and is capable of converting up to one million samples per second (Msps) on each channel. The built-in calibration system can remove offset and gain error when loaded with calibration values from software.

Figure 29-1. DAC overview.



A DAC conversion is automatically started when new data to be converted are available. Events from the event system can also be used to trigger a conversion, and this enables synchronized and timed conversions between the DAC and other peripherals, such as a timer/counter. The EDMA controller can be used to transfer data to the DAC.

The DAC is capable of driving both resistive and capacitive loads aswell as loads which combine both. A low-power mode is available, which will reduce the drive strength of the output. Internal and external voltage references can be used. The DAC output is also internally available for use as input to the analog comparator or ADC.

PORTA has one DAC. Notation of this peripheral is DACA.

30. AC – Analog Comparator

30.1 Features

- Two Analog Comparators
- Selectable propagation delay
- Selectable hysteresis
  - No
  - Small
  - Large
- Analog Comparator output available on pin
- Flexible Input Selection
  - All pins on the port
  - Output from the DAC
  - Bandgap reference voltage.
  - A 64-level programmable voltage scaler of the internal VCC voltage
- Interrupt and event generation on
  - Rising edge
  - Falling edge
  - Toggle
- Window function interrupt and event generation on
  - Signal above window
  - Signal inside window
  - Signal below window
- Constant current source with configurable output pin selection
- Source of asynchronous event

30.2 Overview

The Analog Comparator (AC) compares the voltage level on two inputs and gives a digital output based on this comparison. The Analog Comparator may be configured to give interrupt requests and/or synchronous/asynchronous events upon several different combinations of input change.

One important property of the Analog Comparator when it comes to the dynamic behavior, is the hysteresis. This parameter may be adjusted in order to find the optimal operation for each application.

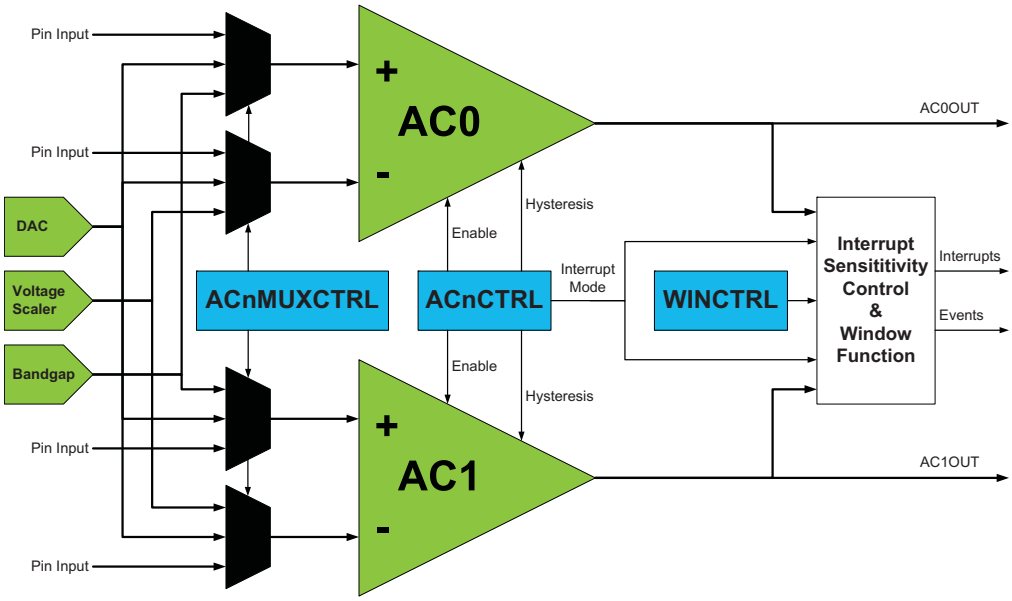
The input section includes analog port pins, several internal signals and a 64-level programmable voltage scaler. The analog comparator output state can also be directly available on a pin for use by external devices. Using as pair they can also be set in Window mode to monitor a signal compared to a voltage window instead of a voltage level.

A constant current source can be enabled and output on a selectable pin. This can be used to replace, for example, external resistors used to charge capacitors in capacitive touch sensing applications.

The analog comparators are always grouped in pairs on each port. These are called analog comparator 0 (AC0) and analog comparator 1 (AC1). They have identical behavior, but separate control registers. Used as pair, they can be set in window mode to compare a signal to a voltage range instead of a voltage level.

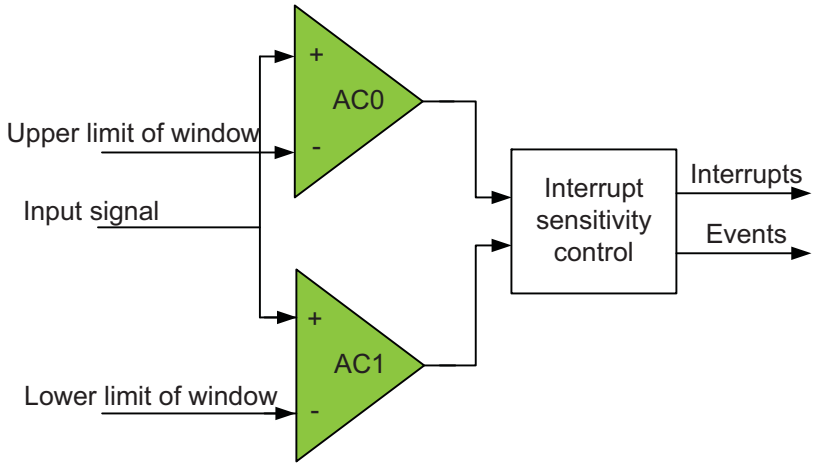
PORTA has one AC pair. Notation is ACA.

Figure 30-1. Analog comparator overview.



The window function is realized by connecting the external inputs of the two analog comparators in a pair as shown in Figure 30-2.

Figure 30-2. Analog comparator window function.



31. Programming and Debugging

31.1 Features

- Programming
  - External programming through PDI interface
    - Minimal protocol overhead for fast operation
    - Built-in error detection and handling for reliable operation
  - Boot loader support for programming through any communication interface
- Debugging
  - Nonintrusive, real-time, on-chip debug system
  - No software or hardware resources required from device except pin connection
  - Program flow control
  - Go, Stop, Reset, Step Into, Step Over, Step Out, Run-to-Cursor
  - Unlimited number of user program breakpoints
  - Unlimited number of user data breakpoints, break on:
  - Data location read, write, or both read and write
  - Data location content equal or not equal to a value
  - Data location content is greater or smaller than a value
  - Data location content is within or outside a range
  - No limitation on device clock frequency
- Program and Debug Interface (PDI)
  - Two-pin interface for external programming and debugging
  - Uses the Reset pin and a dedicated pin
  - No I/O pins required during programming or debugging

31.2 Overview

The Program and Debug Interface (PDI) is an Atmel proprietary interface for external programming and on-chip debugging of a device. The PDI supports fast programming of nonvolatile memory (NVM) spaces; flash, EEPOM, fuses, lock bits, and the user signature row.

Debug is supported through an on-chip debug system that offers nonintrusive, real-time debug. It does not require any software or hardware resources except for the device pin connection. Using the Atmel tool chain, it offers complete program flow control and support for an unlimited number of program and complex data breakpoints. Application debug can be done from a C or other high-level language source code level, as well as from an assembler and disassemble level.

Programming and debugging can be done through the PDI physical layer. This is a two-pin interface that uses the Reset pin for the clock input (PDI_CLK) and one other dedicated pin for data input and output (PDI_DATA). Any external programmer or on-chip debugger/emulator can be directly connected to this interface.

32. Pinout and Pin Functions

The device pinout is shown in Pinout and Block Diagram on page 3. In addition to general purpose I/O functionality, each pin can have several alternate functions. This will depend on which peripheral is enabled and connected to the actual pin. Only one of the pin functions can be used at time.

32.1 Alternate Pin Function Description

The tables below show the notation for all pin functions available and describe its function.

32.1.1 Operation/Power Supply

V_{CC}	Digital supply voltage
AV_{CC}	Analog supply voltage
GND	Ground

32.1.2 Port Interrupt functions

SYNC	Port pin with full synchronous and limited asynchronous interrupt function
ASYNC	Port pin with full synchronous and full asynchronous interrupt function

32.1.3 Analog functions

ACn	Analog Comparator input pin n
ACnOUT	Analog Comparator n Output
ADCn	Analog to Digital Converter input pin n
DACn	Digital to Analog Converter output pin n
A_{REF}	Analog Reference input pin

32.1.4 Timer/Counter and WEX functions

OCnx	Output Compare channel x for timer/counter n
OCnxLS	Output Compare Channel x Low Side for Timer/Counter n
OCnxHS	Output Compare Channel x High Side for Timer/Counter n

32.1.5 Communication functions

SCL	Serial Clock for TWI
SDA	Serial Data for TWI
SCLIN	Serial Clock In for TWI when external driver interface is enabled
SCLOUT	Serial Clock Out for TWI when external driver interface is enabled

SDAIN	Serial Data In for TWI when external driver interface is enabled
SDAOUT	Serial Data Out for TWI when external driver interface is enabled
XCKn	Transfer Clock for USART n
RXDn	Receiver Data for USART n
TXDn	Transmitter Data for USART n
$\overline{SS}$	Slave Select for SPI
MOSI	Master Out Slave In for SPI
MISO	Master In Slave Out for SPI
SCK	Serial Clock for SPI

32.1.6 Oscillators, Clock and Event

TOSCn	Timer Oscillator pin n
XTALn	Input/Output for Oscillator pin n
CLKOUT	Peripheral Clock Output
EVOUT	Event Channel Output
RTCOUT	RTC Clock Source Output

32.1.7 Debug/System functions

$\overline{RESET}$	Reset pin
PDI_CLK	Program and Debug Interface Clock pin
PDI_DATA	Program and Debug Interface Data pin

32.2 Alternate Pin Functions

The tables below show the primary/default function for each pin on a port in the first column, the pin number in the second column, and then all alternate pin functions in the remaining columns. The head row shows what peripheral that enable and use the alternate pin functions.

For better flexibility, some alternate functions also have selectable pin locations for their functions, this is noted under the first table where this apply.

Table 32-1. PORT A – Alternate Functions.

PORT A	Pin#	ADCA POS/ GAINPOS	ADCA NEG/ GAINNEG	DACA	ACA POS	ACA NEG	ACA OUT	REFA
PA0	6	ADC 0	ADC 0		AC0	AC0		AREF
PA1	5	ADC 1	ADC 1		AC1	AC1		
PA2	4	ADC 2	ADC 2	DAC0	AC2			
PA3	3	ADC 3	ADC 3	DAC1	AC3	AC3		
PA4	2	ADC 4	ADC 4		AC4			
PA5	31	ADC 5	ADC 5		AC5	AC5		
PA6	30	ADC 6	ADC 6		AC6		AC1OUT	
PA7	29	ADC 7	ADC 7			AC7	AC0OUT	

Table 32-2. PORT C – Alternate Functions.

PORT C	Pin #	TCC4	WEXC	TCC5	USARTC0	SPIC	TWI	XCL (LUT)	EXTCLK	AC OUT
PC0	16	OC4A	OC4ALS				SDA	IN1/OUT0		
PC1	15	OC4B	OC4AHS		XCK0		SCL	IN2		
PC2	14	OC4C	OC4BLS		RXD0			IN0		
PC3	13	OC4D	OC4BHS		TXD0			IN3		
PC4	12	OC4A	OC4CLS	OC5A		$\overline{SS}$		IN1/OUT0	EXTCLK	
PC5	11	OC4B	OC4CHS	OC5B	XCK0	SCK		IN2		
PC6	10	OC4C	OC4DLS		RXD0	MISO		IN0		AC1OUT
PC7	9	OC4D	OC4DHS		TXD0	MOSI		IN3		AC0OUT

Table 32-3. Debug – Program and Debug Functions.

DEBUG	Pin #	PROG
RESET	8	PDI CLOCK
PDI	7	PDI DATA

Table 32-4. PORT R – Alternate Functions.

PORT R	Pin #	XTAL	TOSC	EXTCLK	CLOCKOUT	EVENTOUT	RTCOUT	AC OUT
PR0	20	XTAL2	TOSC2		CLKOUT	EVOUT	RTCOUT	AC1 OUT
PR1	19	XTAL1	TOSC1	EXTCLK				AC0 OUT

Table 32-5. PORT D – Alternate Functions.

PORT D	Pin #	ADCAPOS GAINPOS	TCD5	USART D0	TWID (Bridge)	XCL (LUT)	XCL (TC)	CLOCK OUT	EVENT OUT	RTCOUT	ACOUT	REFD
PD0	28	ADC8			SDA	IN1/ OUT0						AREF
PD1	27	ADC9		XCK0	SCL	IN2						
PD2	26	ADC10		RXD0		IN0	OC0					
PD3	25	ADC11		TXD0		IN3	OC1					
PD4	24	ADC12	OC5A			IN1/ OUT0		CLKOUT	EVOUT			
PD5	23	ADC13	OC5B	XCK0		IN2						
PD6	22	ADC14		RXD0		IN0				RTCOUT	AC1OUT	
PD7	21	ADC15		TXD0		IN3		CLKOUT	EVOUT		AC0OUT	

33. Peripheral Module Address Map

The address maps show the base address for each peripheral and module in XMEGA E5. For complete register description and summary for each peripheral module, refer to the XMEGA E Manual.

Table 33-1. Peripheral module address map.

Base Address	Name	Description
0x0000	GPIO	General Purpose IO Registers
0x0010	VPORT0	Virtual Port A
0x0014	VPORT1	Virtual Port C
0x0018	VPORT2	Virtual Port D
0x001C	VPORT3	Virtual Port R
0x0030	CPU	CPU
0x0040	CLK	Clock Control
0x0048	SLEEP	Sleep Controller
0x0050	OSC	Oscillator Control
0x0060	DFLLRC32M	DFLL for the 32MHz Internal Oscillator
0x0070	PR	Power Reduction
0x0078	RST	Reset Controller
0x0080	WDT	Watch-Dog Timer
0x0090	MCU	MCU Control
0x00A0	PMIC	Programmable Multilevel Interrupt Controller
0x00B0	PORTCFG	Port Configuration
0x00D0	CRC	CRC Module
0x0100	EDMA	Enhanced DMA Controller
0x0180	EVSYS	Event System
0x01C0	NVM	Non Volatile Memory (NVM) Controller
0x0200	ADCA	Analog to Digital Converter on port A
0x0300	DACA	Digital to Analog Converter on port A
0x0380	ACA	Analog Comparator pair on port A
0x0400	RTC	Real Time Counter
0x0460	XCL	XMEGA Custom Logic Module
0x0480	TWIC	Two-Wire Interface on port C
0x0600	PORTA	Port A
0x0640	PORTC	Port C

Base Address	Name	Description
0x0660	PORTD	Port D
0x07E0	PORTR	Port R
0x0800	TCC4	Timer/Counter 4 on port C
0x0840	TCC5	Timer/Counter 5 on port C
0x0880	FAULTC4	Fault Extension on TCC4
0x0890	FAULTC5	Fault Extensionon TCC5
0x08A0	WEXC	Waveform Extension on port C
0x08B0	HIRESC	High Resolution Extension on port C
0x08C0	USARTC0	USART 0 on port C
0x08E0	SPIC	Serial Peripheral Interface on port C
0x08F8	IRCOM	Infrared Communication Module
0x0940	TCD5	Timer/Counter 5 on port D
0x09C0	USARTD0	USART 0 on port D

34. Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
Arithmetic and Logic Instructions					
ADD	Rd, Rr	Add without Carry	Rd ← Rd + Rr	Z,C,N,V,S,H	1
ADC	Rd, Rr	Add with Carry	Rd ← Rd + Rr + C	Z,C,N,V,S,H	1
ADIW	Rd, K	Add Immediate to Word	Rd ← Rd + 1:Rd + K	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract without Carry	Rd ← Rd - Rr	Z,C,N,V,S,H	1
SUBI	Rd, K	Subtract Immediate	Rd ← Rd - K	Z,C,N,V,S,H	1
SBC	Rd, Rr	Subtract with Carry	Rd ← Rd - Rr - C	Z,C,N,V,S,H	1
SBCI	Rd, K	Subtract Immediate with Carry	Rd ← Rd - K - C	Z,C,N,V,S,H	1
SBIW	Rd, K	Subtract Immediate from Word	Rd + 1:Rd ← Rd + 1:Rd - K	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND	Rd ← Rd • Rr	Z,N,V,S	1
ANDI	Rd, K	Logical AND with Immediate	Rd ← Rd • K	Z,N,V,S	1
OR	Rd, Rr	Logical OR	Rd ← Rd v Rr	Z,N,V,S	1
ORI	Rd, K	Logical OR with Immediate	Rd ← Rd v K	Z,N,V,S	1
EOR	Rd, Rr	Exclusive OR	Rd ← Rd ⊕ Rr	Z,N,V,S	1
COM	Rd	One's Complement	Rd ← $FF - Rd	Z,C,N,V,S	1
NEG	Rd	Two's Complement	Rd ← $00 - Rd	Z,C,N,V,S,H	1
SBR	Rd,K	Set Bit(s) in Register	Rd ← Rd v K	Z,N,V,S	1
CBR	Rd,K	Clear Bit(s) in Register	Rd ← Rd • ($FFh - K)	Z,N,V,S	1
INC	Rd	Increment	Rd ← Rd + 1	Z,N,V,S	1
DEC	Rd	Decrement	Rd ← Rd - 1	Z,N,V,S	1
TST	Rd	Test for Zero or Minus	Rd ← Rd • Rd	Z,N,V,S	1
CLR	Rd	Clear Register	Rd ← Rd ⊕ Rd	Z,N,V,S	1
SER	Rd	Set Register	Rd ← $FF	None	1
MUL	Rd,Rr	Multiply Unsigned	R1:R0 ← Rd x Rr (UU)	Z,C	2
MULS	Rd,Rr	Multiply Signed	R1:R0 ← Rd x Rr (SS)	Z,C	2
MULSU	Rd,Rr	Multiply Signed with Unsigned	R1:R0 ← Rd x Rr (SU)	Z,C	2
FMUL	Rd,Rr	Fractional Multiply Unsigned	R1:R0 ← Rd x Rr<<1 (UU)	Z,C	2
FMULS	Rd,Rr	Fractional Multiply Signed	R1:R0 ← Rd x Rr<<1 (SS)	Z,C	2
FMULSU	Rd,Rr	Fractional Multiply Signed with Unsigned	R1:R0 ← Rd x Rr<<1 (SU)	Z,C	2
DES	K	Data Encryption	if (H = 0) then R15:R0 ← Encrypt(R15:R0, K) else if (H = 1) then R15:R0 ← Decrypt(R15:R0, K)		1/2
Branch instructions					
RJMP	k	Relative Jump	PC ← PC + k + 1	None	2
IJMP		Indirect Jump to (Z)	PC(15:0) ← Z, PC(21:16) ← 0	None	2
EIJMP		Extended Indirect Jump to (Z)	PC(15:0) ← Z, PC(21:16) ← EIND	None	2
JMP	k	Jump	PC ← k	None	3
RCALL	k	Relative Call Subroutine	PC ← PC + k + 1	None	2 / 3(1)

Mnemonics	Operands	Description	Operation	Flags	#Clocks
ICALL		Indirect Call to (Z)	PC(15:0) ← Z, PC(21:16) ← 0	None	2 / 3[1]
EICALL		Extended Indirect Call to (Z)	PC(15:0) ← Z, PC(21:16) ← EIND	None	3[1]
CALL	k	call Subroutine	PC ← k	None	3 / 4[1]
RET		Subroutine Return	PC ← STACK	None	4 / 5[1]
RETI		Interrupt Return	PC ← STACK	I	4 / 5[1]
CPSE	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) PC ← PC + 2 or 3	None	1 / 2 / 3
CP	Rd,Rr	Compare	Rd - Rr	Z,C,N,V,S,H	1
CPC	Rd,Rr	Compare with Carry	Rd - Rr - C	Z,C,N,V,S,H	1
CPI	Rd,K	Compare with Immediate	Rd - K	Z,C,N,V,S,H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b) = 0) PC ← PC + 2 or 3	None	1 / 2 / 3
SBRS	Rr, b	Skip if Bit in Register Set	if (Rr(b) = 1) PC ← PC + 2 or 3	None	1 / 2 / 3
SBIC	A, b	Skip if Bit in I/O Register Cleared	if (I/O(A,b) = 0) PC ← PC + 2 or 3	None	2 / 3 / 4
SBIS	A, b	Skip if Bit in I/O Register Set	If (I/O(A,b) =1) PC ← PC + 2 or 3	None	2 / 3 / 4
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then PC ← PC + k + 1	None	1 / 2
BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then PC ← PC + k + 1	None	1 / 2
BREQ	k	Branch if Equal	if (Z = 1) then PC ← PC + k + 1	None	1 / 2
BRNE	k	Branch if Not Equal	if (Z = 0) then PC ← PC + k + 1	None	1 / 2
BRCS	k	Branch if Carry Set	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRCC	k	Branch if Carry Cleared	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRSH	k	Branch if Same or Higher	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRLO	k	Branch if Lower	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRMI	k	Branch if Minus	if (N = 1) then PC ← PC + k + 1	None	1 / 2
BRPL	k	Branch if Plus	if (N = 0) then PC ← PC + k + 1	None	1 / 2
BRGE	k	Branch if Greater or Equal, Signed	if (N ⊕ V= 0) then PC ← PC + k + 1	None	1 / 2
BRLT	k	Branch if Less Than, Signed	if (N ⊕ V= 1) then PC ← PC + k + 1	None	1 / 2
BRHS	k	Branch if Half Carry Flag Set	if (H = 1) then PC ← PC + k + 1	None	1 / 2
BRHC	k	Branch if Half Carry Flag Cleared	if (H = 0) then PC ← PC + k + 1	None	1 / 2
BRTS	k	Branch if T Flag Set	if (T = 1) then PC ← PC + k + 1	None	1 / 2
BRTC	k	Branch if T Flag Cleared	if (T = 0) then PC ← PC + k + 1	None	1 / 2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then PC ← PC + k + 1	None	1 / 2
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC ← PC + k + 1	None	1 / 2
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	None	1 / 2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	None	1 / 2
Data transfer instructions					
MOV	Rd, Rr	Copy Register	Rd ← Rr	None	1
MOVW	Rd, Rr	Copy Register Pair	Rd+1:Rd ← Rr+1:Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1

Mnemonics	Operands	Description	Operation	Flags	#Clocks
LDS	Rd, k	Load Direct from data space	Rd ← (k)	None	2[1][2]
LD	Rd, X	Load Indirect	Rd ← (X)	None	1[1][2]
LD	Rd, X+	Load Indirect and Post-Increment	Rd ← (X) X ← X + 1	None	1[1][2]
LD	Rd, -X	Load Indirect and Pre-Decrement	X ← X - 1, ← X - 1 Rd ← (X) ← (X)	None	2[1][2]
LD	Rd, Y	Load Indirect	Rd ← (Y) ← (Y)	None	1[1][2]
LD	Rd, Y+	Load Indirect and Post-Increment	Rd ← (Y) Y ← Y + 1	None	1[1][2]
LD	Rd, -Y	Load Indirect and Pre-Decrement	Y ← Y - 1 Rd ← (Y)	None	2[1][2]
LDD	Rd, Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2[1][2]
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	1[1][2]
LD	Rd, Z+	Load Indirect and Post-Increment	Rd ← (Z), Z ← Z+1	None	1[1][2]
LD	Rd, -Z	Load Indirect and Pre-Decrement	Z ← Z - 1, Rd ← (Z)	None	2[1][2]
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2[1][2]
STS	k, Rr	Store Direct to Data Space	(k) ← Rd	None	2[1]
ST	X, Rr	Store Indirect	(X) ← Rr	None	1[1]
ST	X+, Rr	Store Indirect and Post-Increment	(X) ← Rr, X ← X + 1	None	1[1]
ST	-X, Rr	Store Indirect and Pre-Decrement	X ← X - 1, (X) ← Rr	None	2[1]
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	1[1]
ST	Y+, Rr	Store Indirect and Post-Increment	(Y) ← Rr, Y ← Y + 1	None	1[1]
ST	-Y, Rr	Store Indirect and Pre-Decrement	Y ← Y - 1, (Y) ← Rr	None	2[1]
STD	Y+q, Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2[1]
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	1[1]
ST	Z+, Rr	Store Indirect and Post-Increment	(Z) ← Rr Z ← Z + 1	None	1[1]
ST	-Z, Rr	Store Indirect and Pre-Decrement	Z ← Z - 1	None	2[1]
STD	Z+q,Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2[1]
LPM		Load Program Memory	R0 ← (Z)	None	3
LPM	Rd, Z	Load Program Memory	Rd ← (Z)	None	3
LPM	Rd, Z+	Load Program Memory and Post-Increment	Rd ← (Z), Z ← Z + 1	None	3
ELPM		Extended Load Program Memory	R0 ← (RAMPZ:Z)	None	3
ELPM	Rd, Z	Extended Load Program Memory	Rd ← (RAMPZ:Z)	None	3
ELPM	Rd, Z+	Extended Load Program Memory and Post-Increment	Rd ← (RAMPZ:Z), Z ← Z + 1	None	3
SPM		Store Program Memory	(RAMPZ:Z) ← R1:R0	None	-
SPM	Z+	Store Program Memory and Post-Increment by 2	(RAMPZ:Z) ← R1:R0, Z ← Z + 2	None	-

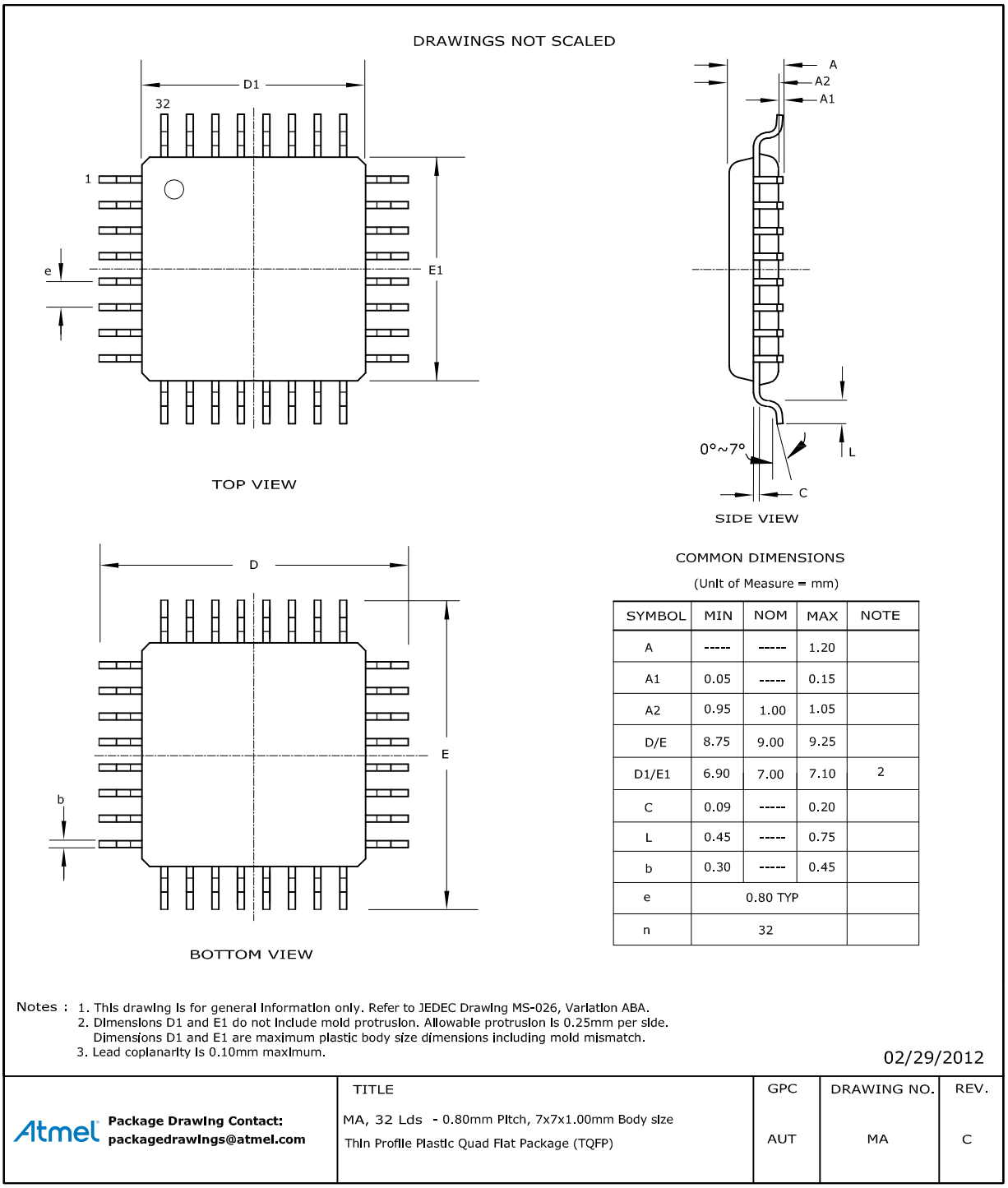
Mnemonics	Operands	Description	Operation	Flags	#Clocks
IN	Rd, A	In From I/O Location	Rd ← I/O(A)	None	1
OUT	A, Rr	Out To I/O Location	I/O(A) ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	1[1]
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2[1]
XCH	Z, Rd	Exchange RAM location	Temp ← Rd, Rd ← (Z), (Z) ← Temp	None	2
LAS	Z, Rd	Load and Set RAM location	Temp ← Rd, Rd ← (Z), (Z) ← Temp v (Z)	None	2
LAC	Z, Rd	Load and Clear RAM location	Temp ← Rd, Rd ← (Z), (Z) ← ($FFh – Rd) • (Z)	None	2
LAT	Z, Rd	Load and Toggle RAM location	Temp ← Rd, Rd ← (Z), (Z) ← Temp ⊕ (Z)	None	2
Bit and bit-test instructions					
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0, C ← Rd(7)	Z,C,N,V,H	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0, C ← Rd(0)	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0) ← C, Rd(n+1) ← Rd(n), C ← Rd(7)	Z,C,N,V,H	1
ROR	Rd	Rotate Right Through Carry	Rd(7) ← C, Rd(n) ← Rd(n+1), C ← Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0) ↔ Rd(7..4)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
SBI	A, b	Set Bit in I/O Register	I/O(A, b) ← 1	None	1
CBI	A, b	Clear Bit in I/O Register	I/O(A, b) ← 0	None	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1

Mnemonics	Operands	Description	Operation	Flags	#Clocks
SEV		Set Two's Complement Overflow	V ← 1	V	1
CLV		Clear Two's Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half Carry Flag in SREG	H ← 1	H	1
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1
MCU control instructions					
BREAK		Break	(See specific descr. for BREAK)	None	1
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep)	None	1
WDR		Watchdog Reset	(see specific descr. for WDR)	None	1

Notes:
1. Cycle times for data memory accesses assume internal memory accesses, and are not valid for accesses via the external RAM interface.
2. One extra cycle must be added when accessing internal SRAM.

35. Packaging information

35.1 32A



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			

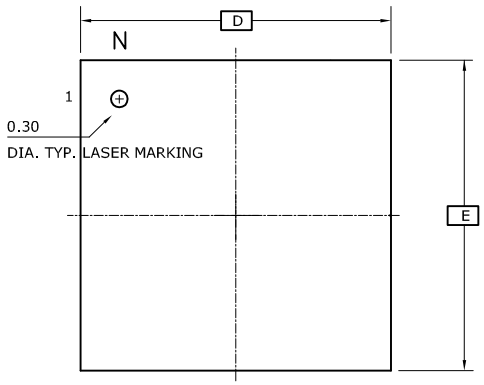
Notes : 1. This drawing is for general information only. Refer to JEDEC Drawing MS-026, Variation ABA.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity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

02/29/2012

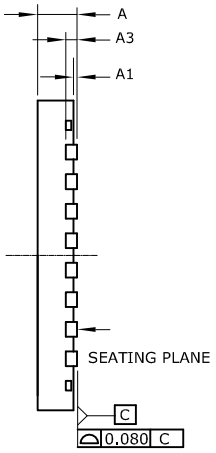
Atmel Package Drawing Contact: packagedrawings@atmel.com	TITLE	GPC	DRAWING NO.	REV.
	MA, 32 Lds - 0.80mm Pitch, 7x7x1.00mm Body size Thin Profile Plastic Quad Flat Package (TQFP)	AUT	MA	C

35.2 32Z

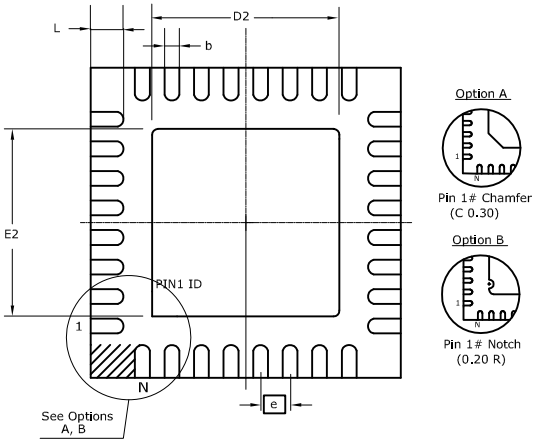
DRAWINGS NOT SCALED



TOP VIEW



SIDE VIEW



BOTTOM VIEW

COMMON DIMENSIONS

(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	0.80	0.85	0.90	
A1	0.00	----	0.05	
A3		0.20 REF		
D/E		5.00 BSC		
D2/E2	3.00	3.10	3.20	
L	0.30	0.40	0.50	
b	0.18	0.25	0.30	2
e		0.50 BSC		
n		32		

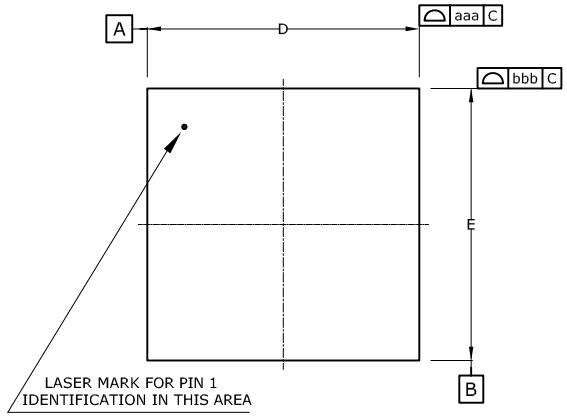
Notes : 1. This drawing is for general information only. Refer to JEDEC Drawing MO-220, Variation VHHD-2, for proper dimensions, tolerances, datums, etc.
2. Dimension b applies to metallized terminal and is measured between 0.15mm and 0.30mm from the terminal tip.
If the terminal has the optical radius on the other end of the terminal, the dimension should not be measured in that radius area.

01/31/2012

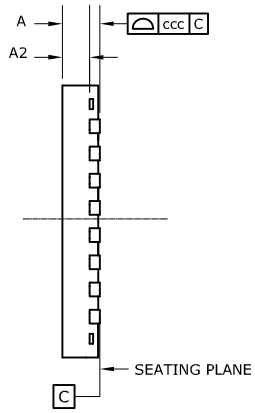
	TITLE	GPC	DRAWING NO.	REV.
Atmel **Package Drawing Contact: packagedrawings@atmel.com**	PN, 32 Leads , 0.50mm pitch, 5 x 5 mm Very Thin quad Flat No Lead Package (VQFN) Sawn	ZMF	PN	I

35.3 32MA

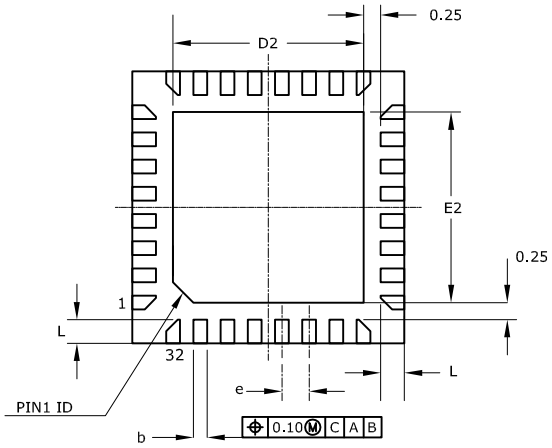
DRAWINGS NOT SCALED



TOP VIEW



SIDE VIEW



BOTTOM VIEW

COMMON DIMENSIONS

(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	0.50	0.55	0.60	
A2	-----	0.40	-----	
D/E		4.00 bsc		
D2/E2	2.70	2.80	2.90	
L	0.30	0.35	0.40	
b	0.15	0.20	0.25	2
e		0.40 bsc		
L/F Thickness		0.152 REF		
Stand Off		MAX 0.05		
TOLERANCES OF FORM AND POSITION				
aaa		0.10		
bbb		0.10		
ccc		0.08		

Notes : 1. This drawing is for general information only.
2. Dimension b applies to metallized terminal and is measured between 0.15mm and 0.25mm from the terminal tip.
If the terminal has the optional radius on the other end of the terminal, the dimension b should not be measured in that radius area.

10/09/2012

	TITLE	GPC	DRAWING NO.	REV.
Atmel **Package Drawing Contact: packagedrawings@atmel.com**	PM, 32 Leads, 0.40mm pitch, 4x4x0.60mm Ultra Thin Quad Flat No Lead Package (UQFN) Sawn	ZZC	PM	A

36. Electrical Characteristics

All typical values are measured at T = 25°C unless other temperature condition is given. All minimum and maximum values are valid across operating temperature and voltage unless other conditions are given.

36.1 Absolute Maximum Ratings

Symbol	Parameter	Min.	Typ.	Max.	Units
V_{CC}	Power supply voltage	-0.3		4	V
I_{VCC}	Current into a V_{CC} pin			200	mA
I_{GND}	Current out of a Gnd pin			200	mA
V_{PIN}	Pin voltage with respect to Gnd and V_{CC}	-0.5		V_{CC}+0.5	V
I_{PIN}	I/O pin sink/source current	-25		25	mA
T_A	Storage temperature	-65		150	°C
T_j	Junction temperature			150	°C

36.2 General Operating Ratings

The device must operate within the ratings listed in Table 36-1 in order for all other electrical characteristics and typical characteristics of the device to be valid.

Table 36-1. General operating conditions.

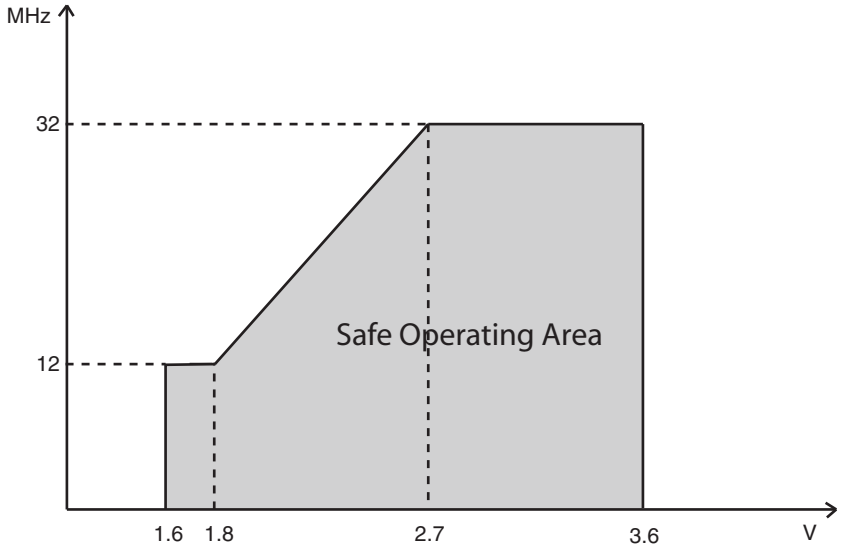
Symbol	Parameter	Min.	Typ.	Max.	Units
V_{CC}	Power supply voltage	1.6		3.6	V
AV_{CC}	Analog supply voltage	1.6		3.6	V
T_A	Temperature range	-40		85	°C
T_j	Junction temperature	-40		105	°C

Table 36-2. Operating voltage and frequency.

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
Clk_{CPU}	CPU clock frequency	V_{CC} = 1.6V	0		12	MHz
		V_{CC} = 1.8V	0		12	
		V_{CC} = 2.7V	0		32	
		V_{CC} = 3.6V	0		32	

The maximum CPU clock frequency depends on V_{CC}. As shown in Figure 36-1 the Frequency vs. V_{CC} curve is linear between 1.8V < V_{CC} < 2.7V.

Figure 36-1. Maximum Frequency vs. V_{CC}.



36.3 Current Consumption

Table 36-3. 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		20		µA
			V_{CC} = 3.0V		30		
		1MHz, Ext. Clk	V_{CC} = 1.8V		140		
			V_{CC} = 3.0V		270		
		2MHz, Ext. Clk	V_{CC} = 1.8V		280	400	
			V_{CC} = 3.0V		0.6	1.2	mA
		32MHz, Ext. Clk	V_{CC} = 3.0V		6	10	
	Idle power consumption (1)	32kHz, Ext. Clk	V_{CC} = 1.8V		5		µA
			V_{CC} = 3.0V		10		
		1MHz, Ext. Clk	V_{CC} = 1.8V		55		
			V_{CC} = 3.0V		100		
		2MHz, Ext. Clk	V_{CC} = 1.8V		110	250	
			V_{CC} = 3.0V		200	350	
		32MHz, Ext. Clk	V_{CC} = 3.0V		3.5	5	mA
	Power-down power consumption	All disabled, T = 25°C	V_{CC} = 3.0V		0.1	0.9	µA
		All disabled, T = 85°C			1	3	
		WDT and sampled BOD enabled, T = 25°C	V_{CC} = 3.0V		0.5		
		WDT and sampled BOD enabled, T = 85°C			1.2	3.5	
	Power-save power consumption	RTC from ULP clock, WDT and sampled BOD enabled, T = 25°C	V_{CC} = 1.8V		0.4		µA
			V_{CC} = 3.0V		0.6		
		RTC from ULP clock, WDT, sampled BOD enabled and 8MHz internal oscillator in low power mode, T = 25°C	V_{CC} = 1.8V		0.5		
			V_{CC} = 3.0V		0.6		
		RTC on 1kHz low power 32.768kHz TOSC, T = 25°C	V_{CC} = 1.8V		0.8		
			V_{CC} = 3.0V		0.9		
		RTC from low power 32.768kHz TOSC, T = 25°C	V_{CC} = 1.8V		0.9		
			V_{CC} = 3.0V		1.0		
	Reset power consumption	Current through $\overline{\text{RESET}}$ pin substracted, T = 25°C	V_{CC} = 3.0V		110		µA

Notes: 1. All Power Reduction Registers set.

Table 36-4. Current consumption for modules and peripherals.

Symbol	Parameter	Condition[1]		Min.	Typ.	Max.	Units
I_{CC}	Internal ULP oscillator				100		nA
	32.768kHz int. oscillator				27		µA
	8MHz int. oscillator	Normal power mode			65		µA
		Low power mode			45		
	32MHz int. oscillator				275		µA
		DFLL enabled with 32.768kHz int. osc. as reference			400		
	PLL	20x multiplication factor, 32MHz int. osc. DIV4 as reference			230		µA
	Watchdog timer				0.3		µA
	BOD	Continuous mode			245		µA
		Sampled mode			0.4		
	Internal 1.0V reference				200		µA
	Internal temperature sensor				100		µA
	ADC	16ksps V_{REF} = Ext ref			1.5		mA
			CURRLIMIT = LOW		1.4		
			CURRLIMIT = MEDIUM		1.3		
			CURRLIMIT = HIGH		1.2		
		75ksps, V_{REF} = Ext ref	CURRLIMIT = LOW		1.7		
		300ksps, V_{REF} = Ext ref			3.1		
	DAC	250ksps V_{REF} = Ext ref No load	Normal mode		1.9		mA
			Low Power mode		1.1		
	AC				200		µA
	EDMA				200		µA
	Timer/counter				25		µA
	USART	Rx and Tx enabled, 9600 BAUD			8		µA
	XCL	16-bit timer/counter			6		µA
	Flash memory and EEPROM programming				4		mA

Note: 1. All parameters measured as the difference in current consumption between module enabled and disabled. All data at VCC = 3.0V, Clk_{SYS} = 1MHz external clock without prescaling, T = 25°C unless other conditions are given.

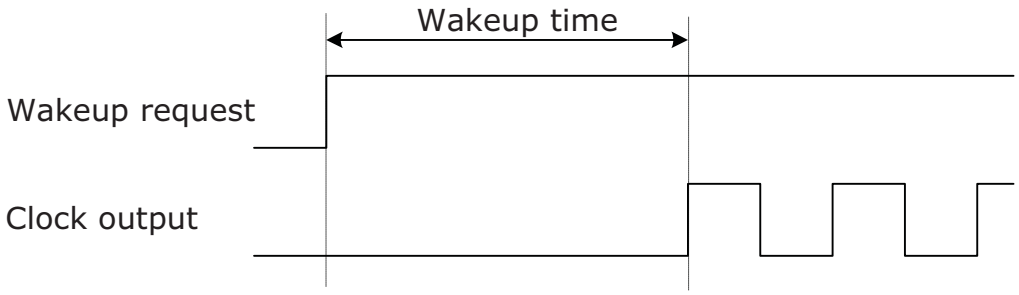
36.4 Wake-up Time from Sleep Modes

Table 36-5. Device wake-up time from sleep modes with various system clock sources.

Symbol	Parameter	Condition		Min.	Typ.(1)	Max.	Units
t_{wakeup}	Wake-up time from idle, standby, and extended standby mode	External 2MHz clock			0.2		µs
		32kHz internal oscillator			120		
		8MHz internal oscillator			0.5		
		32MHz internal oscillator			0.2		
	Wake-up time from power save mode	External 2MHz clock			4.5		
		32kHz internal oscillator			320		
		8MHz internal oscillator	Normal mode		4.5		
			Low power mode		0.5		
		32MHz internal oscillator			0.2		
	Wake-up time from power down mode	External 2MHz clock			4.5		
		32kHz internal oscillator			320		
		8MHz internal oscillator			4.5		
		32MHz internal oscillator			5.0		

Note: 1. The wake-up time is the time from the wake-up request is given until the peripheral clock is available on pin, see Figure 36-2. All peripherals and modules start execution from the first clock cycle, expect the CPU that is halted for four clock cycles before program execution starts.

Figure 36-2. Wake-up time definition.



36.5 I/O Pin Characteristics

The I/O pins complies with the JEDEC LVTTL and LVCMOS specification and the high- and low level input and output voltage limits reflect or exceed this specification.

Table 36-6. I/O pin characteristics.

Symbol	Parameter	Condition		Min.	Typ.	Max.	Units
I_{OH} (1)/ I_{OL} (2)	I/O pin source/sink current			-15		15	mA
V_{IH}	High level input voltage, except XTAL1 and RESET pin	V_{CC} = 2.4 - 3.6V		0.7*V_{CC}		V_{CC}+0.5	V
		V_{CC} = 1.6 - 2.4V		0.8*V_{CC}		V_{CC}+0.5	
V_{IL}	Low level input voltage, except XTAL1 and RESET pin	V_{CC} = 2.4- 3.6V		-0.5		0.3*V_{CC}	V
		V_{CC} = 1.6 - 2.4V		-0.5		0.2*V_{CC}	
V_{OH}	High level output voltage	V_{CC} = 3.3V	I_{OH} = -4mA	2.6	3.1		V
		V_{CC} = 3.0V	I_{OH} = -3mA	2.1	2.7		
		V_{CC} = 1.8V	I_{OH} = -1mA	1.4	1.7		
V_{OL}	Low level output voltage	V_{CC} = 3.3V	I_{OL} = 8mA		0.20	0.76	V
		V_{CC} = 3.0V	I_{OL} = 5mA		0.15	0.64	
		V_{CC} = 1.8V	I_{OL} = 3mA		0.10	0.46	
I_{IN}	Input leakage current	T = 25°C			<0.01	1.0	µA
R_P	Pull/buss keeper resistor				27		kΩ

Notes:
1. The sum of all I_{OH} for PA[7:5] on PORTA must not exceed 100mA.
   The sum of all I_{OH} for PA[4:0] on PORTA must not exceed 200mA.
   The sum of all I_{OH} for PORTD and PORTR must not exceed 100mA.
   The sum of all I_{OH} for PORTC and PDI must not exceed 100mA.
2. The sum of all I_{OL} for PA[7:5] on PORTA must not exceed 100mA.
   The sum of all I_{OL} for PA[4:0] on PORTA must not exceed 100mA.
   The sum of all I_{OL} for PORTD and PORTR must not exceed 100mA.
   The sum of all I_{OL} for PORTC PDI must not exceed 100mA.

36.6 ADC Characteristics

Table 36-7. Power supply, reference and input range.

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
AVCC	Analog supply voltage		V_{CC}- 0.3		V_{CC}+ 0.3	V
VREF	Reference voltage		1		AV_{CC}- 0.6	V
R_{in}	Input resistance	Switched			4.5	kΩ
C_{in}	Input capacitance	Switched			5	pF
R_{AREF}	Reference input resistance	(leakage only)		>10		MΩ
C_{AREF}	Reference input capacitance	Static load		7		pF

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
Vin	Input range		0		V_{REF}	V
Vin	Conversion range	Differential mode, Vinp - Vinn	-0.95*V_{REF}		0.95*V_{REF}	V
Vin	Conversion range	Single ended unsigned mode, Vinp	-0.05*V_{REF}		0.95*V_{REF}	V

Table 36-8. Clock and timing.

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
Clk_{ADC}	ADC Clock frequency	Maximum is 1/4 of Peripheral clock frequency	100		1800	kHz
		Measuring internal signals		125		
f_{ClkADC}	Sample rate		16		300	ksps
f_{ADC}	Sample rate	Current limitation (CURRLIMIT) off	16		300	ksps
		CURRLIMIT = LOW			250	
		CURRLIMIT = MEDIUM			150	
		CURRLIMIT = HIGH			50	
	Sampling Time	1/2 Clk_{ADC} cycle	0.25		5	µs
	Conversion time (latency)	(RES+2)/2+(GAIN !=0) RES (Resolution) = 8 or 12	6		10	Clk_{ADC} cycles
	Start-up time	ADC clock cycles		12	24	Clk_{ADC} cycles
	ADC settling time	After changing reference or input mode		7	7	Clk_{ADC} cycles

Table 36-9. Accuracy characteristics.

Symbol	Parameter	Condition[2]		Min.	Typ.	Max.	Units
RES	Resolution	12-bit resolution	Differential	8	12	12	Bits
			Single ended signed	7	11	11	
			Single ended unsigned	8	12	12	
INL[1]	Integral non-linearity	Differential mode	16kSPS, VREF = 3V		1		lsb
			16kSPS, VREF = 1V		2		
			300kSPS, VREF = 3V		1		
			300kSPS, VREF = 1V		2		
		Single ended unsigned mode	16kSPS, VREF = 3.0V		1	1.5	
			16kSPS, VREF = 1.0V		2	3	

Symbol	Parameter	Condition[2]		Min.	Typ.	Max.	Units
DNL[1]	Differential non-linearity	Differential mode	16kSPS, VREF = 3V		1		lsb
			16kSPS, VREF = 1V		2		
			300kSPS, VREF = 3V		1		
			300kSPS, VREF = 1V		2		
		Single ended unsigned mode	16kSPS, VREF = 3.0V		1	1.5	
			16kSPS, VREF = 1.0V		2	3	
	Offset Error	Differential mode			8		mV
			Temperature drift		0.01		mV/K
			Operating voltage drift		0.25		mV/V
	Gain Error	Differential mode	External reference		-5		mV
			AVCC/1.6		-5		
			AVCC/2.0		-6		
			Bandgap		±10		
			Temperature drift		0.02		mV/K
			Operating voltage drift		2		mV/V
	Gain Error	Single ended unsigned mode	External reference		-8		mV
			AVCC/1.6		-8		
			AVCC/2.0		-8		
			Bandgap		±10		
			Temperature drift		0.03		mV/K
			Operating voltage drift		2		mV/V

Notes: 1. Maximum numbers are based on characterisation and not tested in production, and valid for 10% to 90% input voltage range.
2. Unless otherwise noted all linearity, offset and gain error numbers are valid under the condition that external VREF is used.

Table 36-10. Gain stage characteristics.

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
R_{in}	Input resistance	Switched		4.0		kΩ
C_{sample}	Input capacitance	Switched		4.4		pF
	Signal range	Gain stage output	0		AV_{CC}- 0.6	V
	Propagation delay	ADC conversion rate	1/2	1	3	Clk_{ADC} cycles
	Clock rate	Same as ADC	100		1800	kHz

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
	Gain error	0.5x gain		-1		%
		1x gain		-1		
		8x gain		-1		
		64x gain		-1.5		
	Offset error, input referred	0.5x gain		10		mV
		1x gain		5		
		8x gain		5		
		64x gain		5		

36.7 DAC Characteristics

Table 36-11. Power supply, reference and output range.

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
AV_{CC}	Analog supply voltage		V_{CC}- 0.3		V_{CC}+ 0.3	
AV_{REF}	External reference voltage		1.0		V_{CC}- 0.6	V
$R_{channel}$	DC output impedance				50	Ω
	Linear output voltage range		0.15		VREF-0.15	V
R_{AREF}	Reference input resistance			>10		MΩ
CAREF	Reference input capacitance	Static load		7		pF
	Minimum Resistance load		1			kΩ
	Maximum capacitance load				100	pF
		1000Ω serial resistance			1	nF
	Output sink/source	Operating within accuracy specification			AV_{CC}/1000	mA
		Safe operation			10	

Table 36-12. Clock and timing.

Symbol	Parameter	Condition		Min.	Typ.	Max.	Units
f_{DAC}	Conversion rate	C_{load}=100pF, maximum step size	Normal mode	0		1000	ksps
			Low power mode	0		500	

Table 36-13. Accuracy characteristics.

Symbol	Parameter	Condition		Min.	Typ.	Max.	Units
RES	Input Resolution					12	Bits
INL [1]	Integral non-linearity	V_{REF}= Ext 1.0V	V_{CC} = 1.6V		±2.0	±3	lsb
			V_{CC} = 3.6V		±1.5	±2.5	
		V_{REF}=AV_{CC}	V_{CC} = 1.6V		±2.0	±4	
			V_{CC} = 3.6V		±1.5	±4	
		V_{REF}=INT1V	V_{CC} = 1.6V		±5.0		
			V_{CC} = 3.6V		±5.0		
DNL [1]	Differential non-linearity	V_{REF}=Ext 1.0V	V_{CC} = 1.6V		±1.5	3	lsb
			V_{CC} = 3.6V		±0.6	1.5	
		V_{REF}=AV_{CC}	V_{CC} = 1.6V		±1.0	3.5	
			V_{CC} = 3.6V		±0.6	1.5	
		V_{REF}=INT1V	V_{CC} = 1.6V		±4.5		
			V_{CC} = 3.6V		±4.5		
	Gain error	After calibration			<4		lsb
	Gain calibration step size				4		lsb
	Gain calibration drift	V_{REF}= Ext 1.0V			<0.2		mV/K
	Offset error	After calibration			<1		lsb
	Offset calibration step size				1		

Note: 1. Maximum numbers are based on characterisation and not tested in production, and valid for 5% to 95% output voltage range.

36.8 Analog Comparator Characteristics

Table 36-14. Analog comparator characteristics.

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
V_{off}	Input offset voltage			10		mV
I_{lk}	Input leakage current			<10	50	nA
	Input voltage range		-0.1		AV_{CC}	V
	AC startup time			50		µs
V_{hys1}	Hysteresis, none	V_{CC} = 1.6V - 3.6V		0		mV
V_{hys2}	Hysteresis, small	V_{CC} = 1.6V - 3.6V		12		mV
V_{hys3}	Hysteresis, large	V_{CC} = 1.6V - 3.6V		28		mV
t_{delay}	Propagation delay	V_{CC} = 3.0V, T= 85°C		22	30	ns
		V_{CC} = 1.6V - 3.6V		21	40	
	64-Level Voltage Scaler Integral non-linearity (INL)			0.3	0.5	lsb
	Current source accuracy after calibration			5		%
	Current source calibration range	Single mode	4		6	µA
	Current source calibration range	Double mode	8		12	µA

36.9 Bandgap and Internal 1.0V Reference Characteristics

Table 36-15. Bandgap and Internal 1.0V reference characteristics.

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
	Startup time	As reference for ADC	1 Clk_{PER} + 2.5µs			µs
		As input voltage to ADC and AC		1.5		
BANDGAP	Bandgap voltage			1.1		V
INT1V	Internal 1.00V reference for ADC and DAC	T= 25°C, after calibration	0.99	1.0	1.01	V
	Variation over voltage and temperature	Calibrated at T= 25°C		±3		%

36.9.1 Brownout Detection Characteristics

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
V_{BOT}	BOD level 0 falling V_{CC}		1.50	1.65	1.75	V
	BOD level 1 falling V_{CC}			1.8		
	BOD level 2 falling V_{CC}			2.0		
	BOD level 3 falling V_{CC}			2.2		
	BOD level 4 falling V_{CC}			2.4		
	BOD level 5 falling V_{CC}			2.6		
	BOD level 6 falling V_{CC}			2.8		
	BOD level 7 falling V_{CC}			3.0		
T_{BOD}	Detection time	Continuous mode		0.4		µs
		Sampled mode		1.0		ms
V_{HYST}	Hysteresis	BOD level 0 - 7. Min value measured at BOD level 0		1.0		%

36.10 External Reset Characteristics

Table 36-16. External reset characteristics

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
t_{EXT}	Minimum reset pulse width		90	1000		ns
V_{RST}	Reset threshold voltage (V_{IH})	V_{CC} = 2.7 - 3.6V	0.6*V_{CC}			V
		V_{CC} = 1.6 - 2.7V	0.6*V_{CC}			
	Reset threshold voltage (V_{IL})	V_{CC} = 2.7 - 3.6V			0.5*V_{CC}	
		V_{CC} = 1.6 - 2.7V			0.4*V_{CC}	
R_{RST}	Reset pin Pull-up Resistor			25		kΩ

36.11 Power-on Reset Characteristics

Table 36-17. Power-on reset characteristics.

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
V_{POT-} [1]	POR threshold voltage falling V_{CC}	V_{CC} falls faster than 1V/ms	0.4	1.0		V
		V_{CC} falls at 1 V/ms or slower	0.8	1.3		
V_{POT+}	POR threshold voltage raising V_{CC}			1.3	1.59	V

Note: 1. VPOT- values are only valid when BOD is disabled. When BOD is enabled VPOT- = VPOT+.

36.12 Flash and EEPROM Characteristics

Table 36-18. Endurance and data retention.

Parameter	Condition		Min.	Typ.	Max.	Units
Flash	Write/Erase cycles	25°C	10K			Cycle
		85°C	10K			
	Data retention	25°C	100			Year
		85°C	25			
EEPROM	Write/Erase cycles	25°C	100K			Cycle
		85°C	100K			
	Data retention	25°C	100			Year
		85°C	25			

Table 36-19. Programming time.

Parameter	Condition	Min.	Typ.(1)	Max.	Units
Chip Erase	32KB Flash, EEPROM(2)		50		ms
	16KB Flash, EEPROM(2)		45		
	8KB Flash, EEPROM(2)		42		
Flash	Page erase		4		ms
	Page write		4		
	Atomic page erase and write		8		
EEPROM	Page erase		4		ms
	Page write		4		
	Atomic page erase and write		8		

Notes:
1. Programming is timed from the 2MHz output of 8MHz internal oscillator.
2. EEPROM is not erased if the EESAVE fuse is programmed.

36.13 Clock and Oscillator Characteristics

36.13.1 Calibrated 32.768kHz Internal Oscillator Characteristics

Table 36-20. 32.768kHz internal oscillator characteristics.

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
	Frequency			32.768		kHz
	Factory calibration accuracy	T = 25°C, V_{CC} = 3.0V	-0.5		0.5	%
	User calibration accuracy		-0.5		0.5	%

36.13.2 Calibrated 8MHz Internal Oscillator Characteristics

Table 36-21. 8MHz internal oscillator characteristics.

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
	Frequency range		4.4		9.4	MHz
	Factory calibrated frequency			8		MHz
	Factory calibration accuracy	T = 25°C, V_{CC} = 3.0V	-0.5		0.5	%
	User calibration accuracy		-0.5		0.5	%

36.13.3 Calibrated and tunable 32MHz Internal Oscillator Characteristics

Table 36-22. 32MHz internal oscillator characteristics.

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
	Frequency range	DFLL can tune to this frequency over voltage and temperature	30		55	MHz
	Factory calibrated frequency			32		MHz
	Factory calibration accuracy	T = 25°C, V_{CC} = 3.0V	-1.5		1.5	%
	User calibration accuracy		-0.2		0.2	%
	DFLL calibration step size			0.23		%

36.13.4 32.768kHz Internal ULP Oscillator Characteristics

Table 36-23. 32.768kHz internal ULP oscillator characteristics.

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
	Output frequency			32		kHz
	Accuracy		-30		30	%

36.13.5 Internal Phase Locked Loop (PLL) Characteristics

Table 36-24. Internal PLL characteristics.

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
f_{IN}	Input frequency	Output frequency must be within f_{OUT}	0.4		64	MHz
f_{OUT}	Output frequency (1)	V_{CC}= 1.6 - 1.8V	20		48	MHz
		V_{CC}= 2.7 - 3.6V	20		128	
	Start-up time			25		µs
	Re-lock time			25		µs

Note: 1. The maximum output frequency vs. supply voltage is linear between 1.8V and 2.7V, and can never be higher than four times the maximum CPU frequency.

36.13.6 External Clock Characteristics

Figure 36-3. External clock drive waveform.

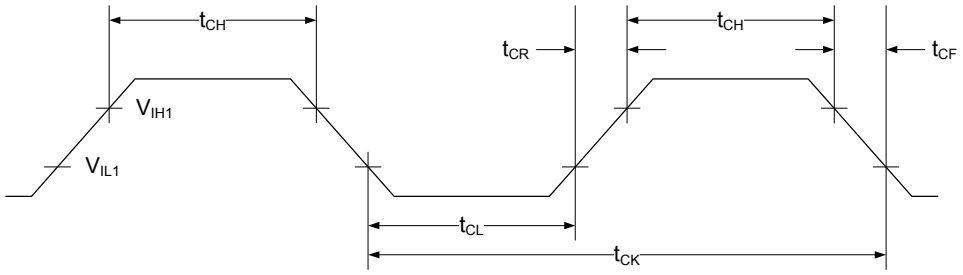


Table 36-25. External clock used as system clock without prescaling.

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
$1/t_{CK}$	Clock Frequency (1)	V_{CC} = 1.6 - 1.8V	0		12	MHz
		V_{CC} = 2.7 - 3.6V	0		32	
t_{CK}	Clock Period	V_{CC} = 1.6 - 1.8V	83.3			ns
		V_{CC} = 2.7 - 3.6V	31.5			
t_{CH}	Clock High Time	V_{CC} = 1.6 - 1.8V	30.0			ns
		V_{CC} = 2.7 - 3.6V	12.5			
t_{CL}	Clock Low Time	V_{CC} = 1.6 - 1.8V	30.0			ns
		V_{CC} = 2.7 - 3.6V	12.5			
t_{CR}	Rise Time (for maximum frequency)	V_{CC} = 1.6 - 1.8V			10	ns
		V_{CC} = 2.7 - 3.6V			3	
t_{CF}	Fall Time (for maximum frequency)	V_{CC} = 1.6 - 1.8V			10	ns
		V_{CC} = 2.7 - 3.6V			3	
Δt_{CK}	Change in period from one clock cycle to the next				10	%

Note: 1. The maximum frequency vs. supply voltage is linear between 1.6V and 2.7V, and the same applies for all other parameters with supply voltage conditions.

Table 36-26. External clock with prescaler (1) for system clock.

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
$1/t_{CK}$	Clock Frequency (2)	V_{CC} = 1.6 - 1.8V	0		90	MHz
		V_{CC} = 2.7 - 3.6V	0		142	
t_{CK}	Clock Period	V_{CC} = 1.6 - 1.8V	11			ns
		V_{CC} = 2.7 - 3.6V	7			
t_{CH}	Clock High Time	V_{CC} = 1.6 - 1.8V	4.5			ns
		V_{CC} = 2.7 - 3.6V	2.4			
t_{CL}	Clock Low Time	V_{CC} = 1.6 - 1.8V	4.5			ns
		V_{CC} = 2.7 - 3.6V	2.4			
t_{CR}	Rise Time (for maximum frequency)	V_{CC} = 1.6 - 1.8V			1.5	ns
		V_{CC} = 2.7 - 3.6V			1.0	
t_{CF}	Fall Time (for maximum frequency)	V_{CC} = 1.6 - 1.8V			1.5	ns
		V_{CC} = 2.7 - 3.6V			1.0	
Δt_{CK}	Change in period from one clock cycle to the next				10	%

Notes: 1. System Clock Prescalers must be set so that maximum CPU clock frequency for device is not exceeded.
2. The maximum frequency vs. supply voltage is linear between 1.6V and 2.7V, and the same applies for all other parameters with supply voltage conditions.

36.13.7 External 16MHz Crystal Oscillator and XOSC Characteristics

Table 36-27. External 16MHz crystal oscillator and XOSC characteristics.

Symbol	Parameter	Condition		Min.	Typ.	Max.	Units
	Cycle to cycle jitter	XOSCPWR=0	FRQRANGE=0		<10		ns
			FRQRANGE=1, 2, or 3		<1		
		XOSCPWR=1			<1		
	Long term jitter	XOSCPWR=0	FRQRANGE=0		<6		ns
			FRQRANGE=1, 2, or 3		<0.5		
		XOSCPWR=1			<0.5		
	Frequency error	XOSCPWR=0	FRQRANGE=0		<0.1		%
			FRQRANGE=1		<0.05		
			FRQRANGE=2 or 3		<0.005		
		XOSCPWR=1			<0.005		

Symbol	Parameter	Condition		Min.	Typ.	Max.	Units
	Duty cycle	XOSCPWR=0	FRQRANGE=0		40		%
			FRQRANGE=1		42		
			FRQRANGE=2 or 3		45		
		XOSCPWR=1			48		
R_Q	Negative impedance (1)	XOSCPWR=0, FRQRANGE=0	0.4MHz resonator, CL=100pF				Ω
			1MHz crystal, CL=20pF				
			2MHz crystal, CL=20pF				
		XOSCPWR=0, FRQRANGE=1, CL=20pF	2MHz crystal				
			8MHz crystal				
			9MHz crystal				
		XOSCPWR=0, FRQRANGE=2, CL=20pF	8MHz crystal				
			9MHz crystal				
			12MHz crystal				
		XOSCPWR=0, FRQRANGE=3, CL=20pF	9MHz crystal				
			12MHz crystal				
			16MHz crystal				
		XOSCPWR=1, FRQRANGE=0, CL=20pF	9MHz crystal				
			12MHz crystal				
			16MHz crystal				
		XOSCPWR=1, FRQRANGE=1, CL=20pF	9MHz crystal				
			12MHz crystal				
			16MHz crystal				
		XOSCPWR=1, FRQRANGE=2, CL=20pF	12MHz crystal				
			16MHz crystal				
		XOSCPWR=1, FRQRANGE=3, CL=20pF	12MHz crystal				
			16MHz crystal				
C_{XTAL1}	Parasitic capacitance XTAL1 pin				5.4		pF
C_{XTAL2}	Parasitic capacitance XTAL2 pin				7.1		
C_{LOAD}	Parasitic capacitance load				3.07		

Note: 1. Numbers for negative impedance are not tested in production but guaranteed from design and characterization.

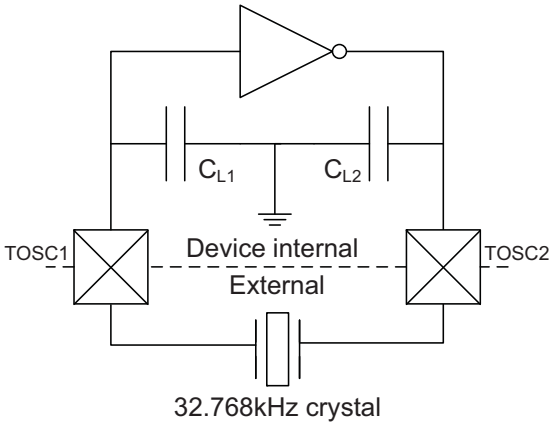
36.13.8 External 32.768kHz Crystal Oscillator and TOSC Characteristics

Table 36-28. External 32.768kHz crystal oscillator and TOSC characteristics.

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
ESR/R1	Recommended crystal equivalent series resistance (ESR)	Crystal load capacitance 6.5pF			60	kΩ
		Crystal load capacitance 9.0pF			35	
C_{TOSC1}	Parasitic capacitance TOSC1 pin			5.3		pF
C_{TOSC2}	Parasitic capacitance TOSC2 pin			7.4		pF
	Recommended safety factor	capacitance load matched to crystal specification	3.0			

Note: 1. See Figure 36-4 for definition.

Figure 36-4. TOSC input capacitance.



The parasitic capacitance between the TOSC pins is C_{L1} + C_{L2} in series as seen from the crystal when oscillating without external capacitors.

36.14 SPI Characteristics

Figure 36-5. SPI timing requirements in master mode.

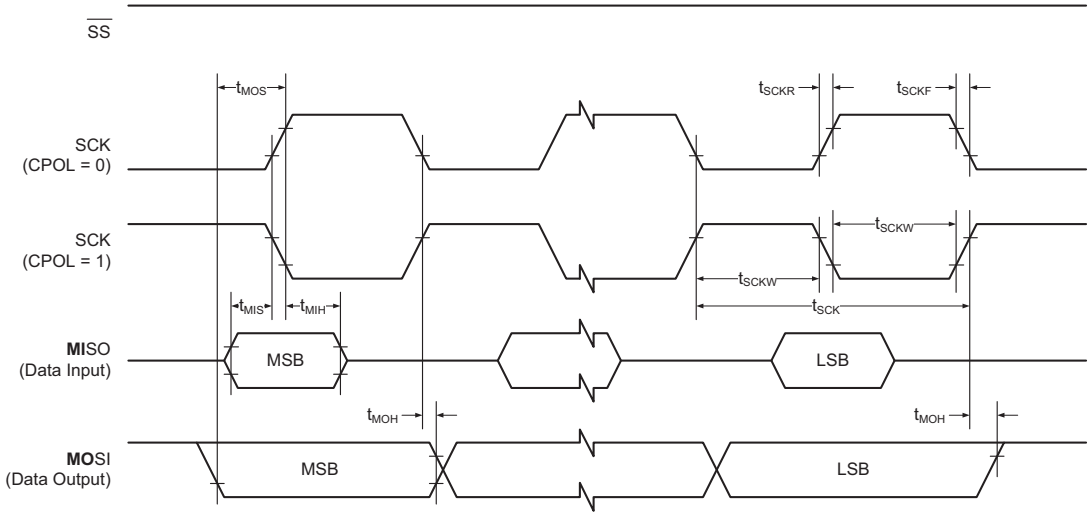


Figure 36-6. SPI timing requirements in slave mode.

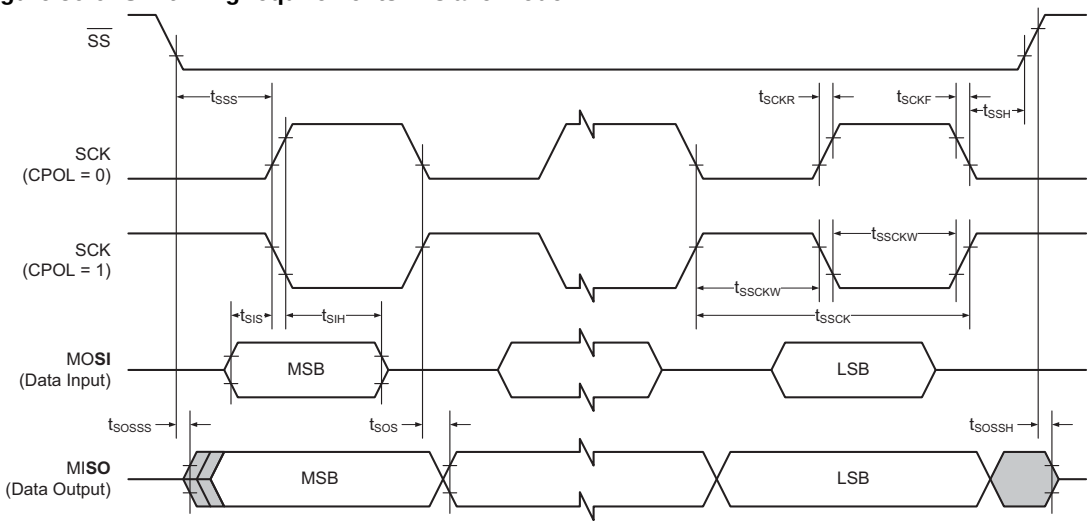


Table 36-29. SPI timing characteristics and requirements.

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
t_{SCK}	SCK period	Master				ns
t_{SCKW}	SCK high/low width	Master		0.5×SCK		
t_{SCKR}	SCK rise time	Master		2.7		
t_{SCKF}	SCK fall time	Master		2.7		
t_{MIS}	MISO setup to SCK	Master		10		
t_{MIH}	MISO hold after SCK	Master		10		
t_{MOS}	MOSI setup SCK	Master		0.5×SCK		
t_{MOH}	MOSI hold after SCK	Master		1.0		
t_{SSCK}	Slave SCK Period	Slave	$4 \times t\ Clk_{PER}$			
t_{SSCKW}	SCK high/low width	Slave	$2 \times t\ Clk_{PER}$			
t_{SSCKR}	SCK rise time	Slave			1600	
t_{SSCKF}	SCK fall time	Slave			1600	
t_{SIS}	MOSI setup to SCK	Slave	3.0			
t_{SIH}	MOSI hold after SCK	Slave	$t\ Clk_{PER}$			
t_{SSS}	$\overline{SS}$ setup to SCK	Slave	21			
t_{SSH}	$\overline{SS}$ hold after SCK	Slave	20			
t_{SOS}	MISO setup SCK	Slave		8.0		
t_{SOH}	MISO hold after SCK	Slave		13		
t_{SOSS}	MISO setup after $\overline{SS}$ low	Slave		11		
t_{SOSH}	MISO hold after $\overline{SS}$ high	Slave		8.0		

36.15 Two-Wire Interface Characteristics

Table 36-6 on page 76 describes the requirements for devices connected to the two-wire interface (TWI) Bus. The Atmel AVR XMEGA TWI meets or exceeds these requirements under the noted conditions. Timing symbols refer to Figure 36-7.

Figure 36-7. Two-wire interface bus timing

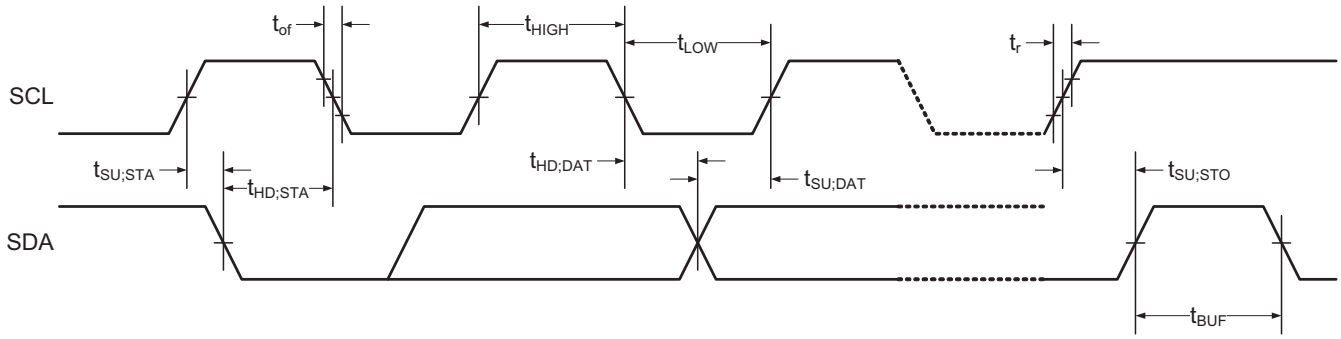


Table 36-30. Two-wire interface characteristics.

Symbol	Parameter	Condition		Min.	Typ.	Max.	Units
V_{IH}	Input high voltage			$0.7V_{CC}$		V_{CC}+0.5	V
V_{IL}	Input low voltage			-0.5		$0.3V_{CC}$	V
V_{hys}	Hysteresis of Schmitt trigger inputs			$0.05V_{CC}$ (1)			V
V_{OL}	Output low voltage	3mA, sink current		0		0.4	V
I_{OL}	Low level output current	$f_{SCL} \le 400kHz$	V_{OL} = 0.4V	3			mA
		$f_{SCL} \le 1MHz$		20			
t_r	Rise time for both SDA and SCL	$f_{SCL} \le 400kHz$		$20+0.1C_b$ (1)(2)		300	ns
		$f_{SCL} \le 1MHz$				120	
t_{of}	Output fall time from V_{IHmin} to V_{ILmax}	10pF< C_b <400pF (2)	$f_{SCL} \le 400kHz$	$20+0.1C_b$ (1)(2)		250	ns
			$f_{SCL} \le 1MHz$			120	
t_{SP}	Spikes suppressed by Input filter			0		50	ns
I_I	Input current for each I/O Pin	0.1 V_{CC} <V_I <0.9 V_{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		1	MHz
R_P	Value of pull-up resistor	$f_{SCL} \le 100kHz$		(V_{CC}-0.4V) /I_{OL}		100ns/C_b	Ω
		$f_{SCL} \le 400kHz$				300ns/C_b	
		$f_{SCL} \le 1MHz$				550ns/C_b	
$t_{HD;STA}$	Hold time (repeated) START condition	$f_{SCL} \le 100kHz$		4			µs
		$f_{SCL} \le 400kHz$		0.6			
		$f_{SCL} \le 1MHz$		0.26			

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
t_{LOW}	Low period of SCL Clock	$f_{SCL} \leq 100kHz$	4.7			µs
		$f_{SCL} \leq 400kHz$	1.3			
		$f_{SCL} \leq 1MHz$	0.5			
t_{HIGH}	High period of SCL Clock	$f_{SCL} \leq 100kHz$	4			µs
		$f_{SCL} \leq 400kHz$	0.6			
		$f_{SCL} \leq 1MHz$	0.26			
$t_{SU;STA}$	Set-up time for a repeated START condition	$f_{SCL} \leq 100kHz$	4.7			µs
		$f_{SCL} \leq 400kHz$	0.6			
		$f_{SCL} \leq 1MHz$	0.26			
$t_{HD;DAT}$	Data hold time	$f_{SCL} \leq 100kHz$	0		3.45	µs
		$f_{SCL} \leq 400kHz$	0		0.9	
		$f_{SCL} \leq 1MHz$	0		0.45	
$t_{SU;DAT}$	Data setup time	$f_{SCL} \leq 100kHz$	250			ns
		$f_{SCL} \leq 400kHz$	100			
		$f_{SCL} \leq 1MHz$	50			
$t_{SU;STO}$	Setup time for STOP condition	$f_{SCL} \leq 100kHz$	4			µs
		$f_{SCL} \leq 400kHz$	0.6			
		$f_{SCL} \leq 1MHz$	0.26			
t_{BUF}	Bus free time between a STOP and START condition	$f_{SCL} \leq 100kHz$	4.7			µs
		$f_{SCL} \leq 400kHz$	1.3			
		$f_{SCL} \leq 1MHz$	0.5			

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

37. Typical Characteristics

37.1 Current consumption

37.1.1 Active Mode Supply Current

Figure 37-1. Active mode supply current vs. frequency.

f_{SYS} = 0 – 1MHz external clock, T = 25°C

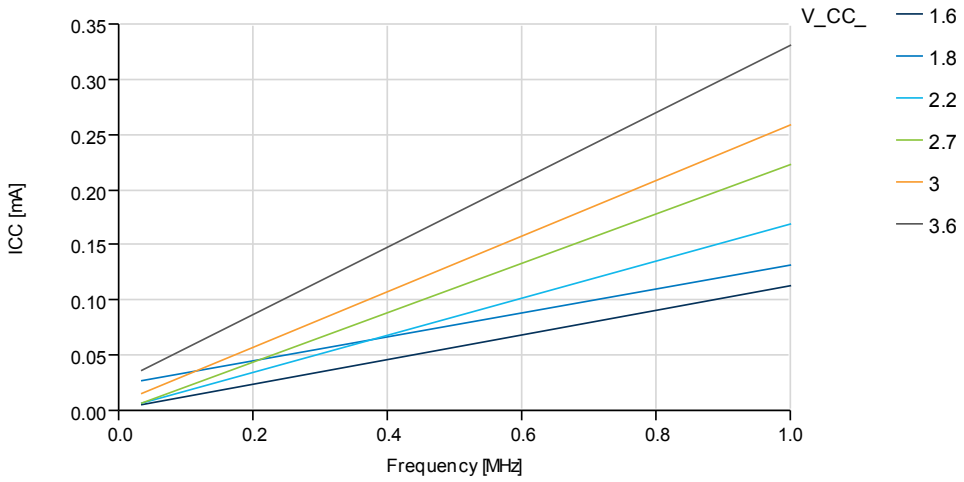


Figure 37-2. Active mode supply current vs. frequency.

f_{SYS} = 0 – 32MHz external clock, T = 25°C

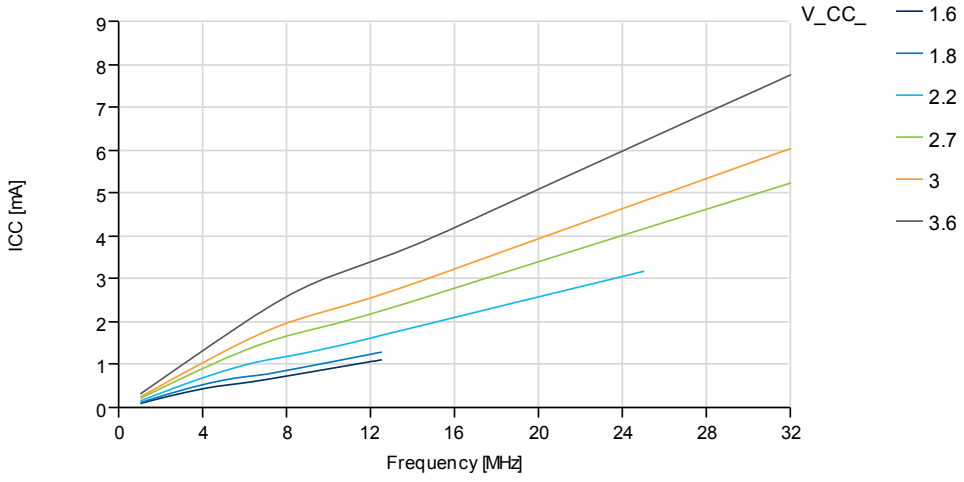


Figure 37-3. Active mode supply current vs. V_{CC}.

f_{SYS} = 32.768kHz internal oscillator

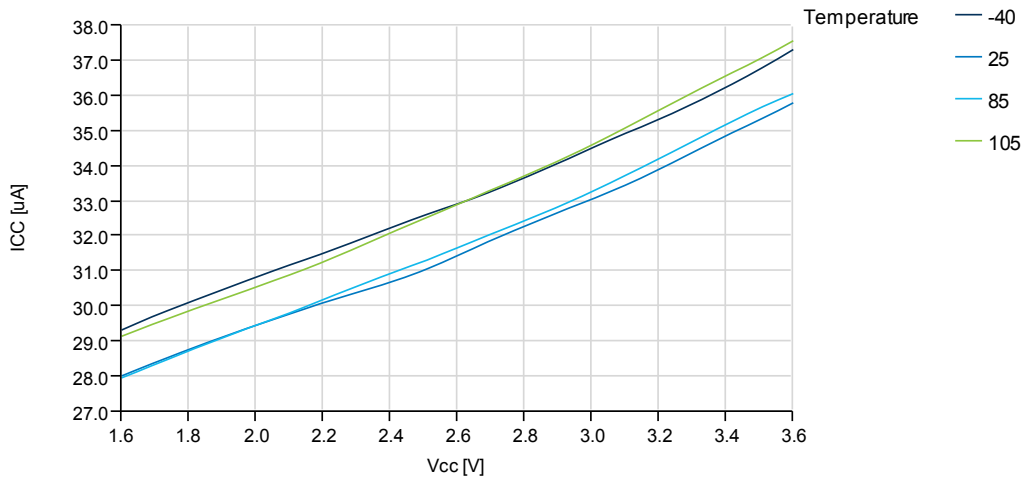


Figure 37-4. Active mode supply current vs. V_{CC}.

f_{SYS} = 1MHz external clock

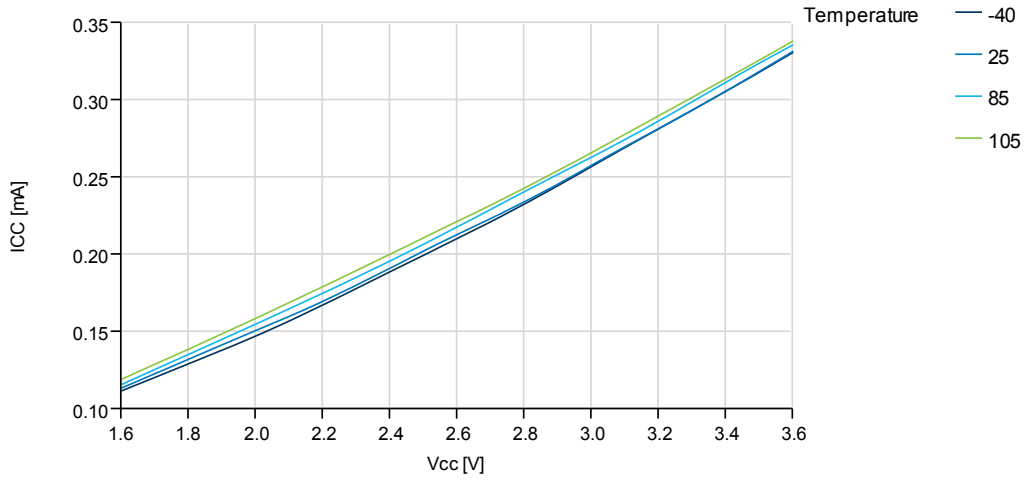


Figure 37-5. Active mode supply current vs. V_{CC}.

f_{SYS} = 8MHz internal oscillator prescaled to 2MHz

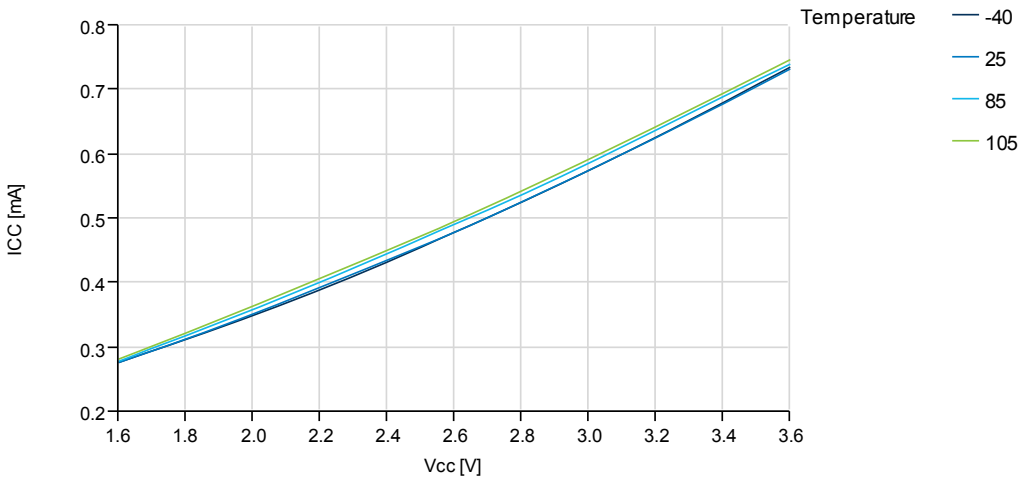


Figure 37-6. Active mode supply current vs. V_{CC}.

f_{SYS} = 8MHz internal oscillator

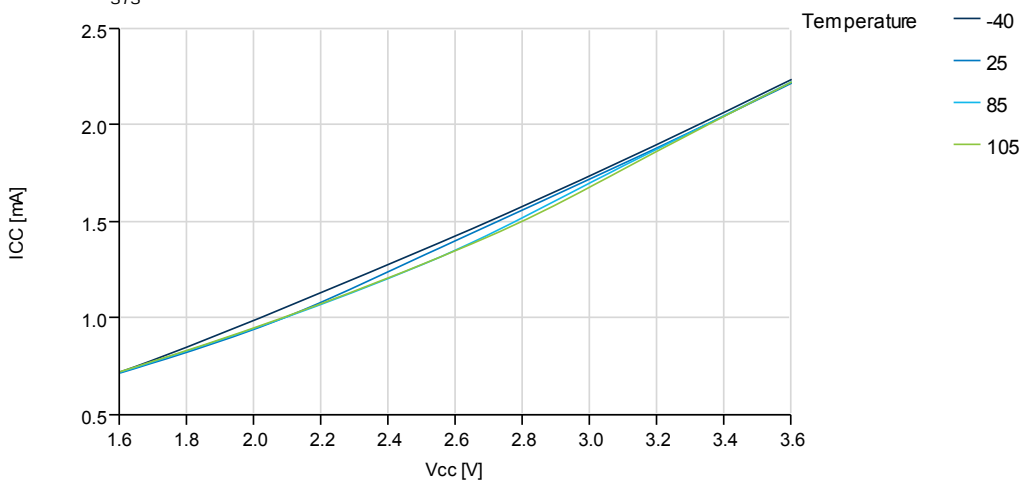


Figure 37-7. Active mode supply current vs. V_{CC}.

f_{SYS} = 32MHz internal oscillator prescaled to 8MHz

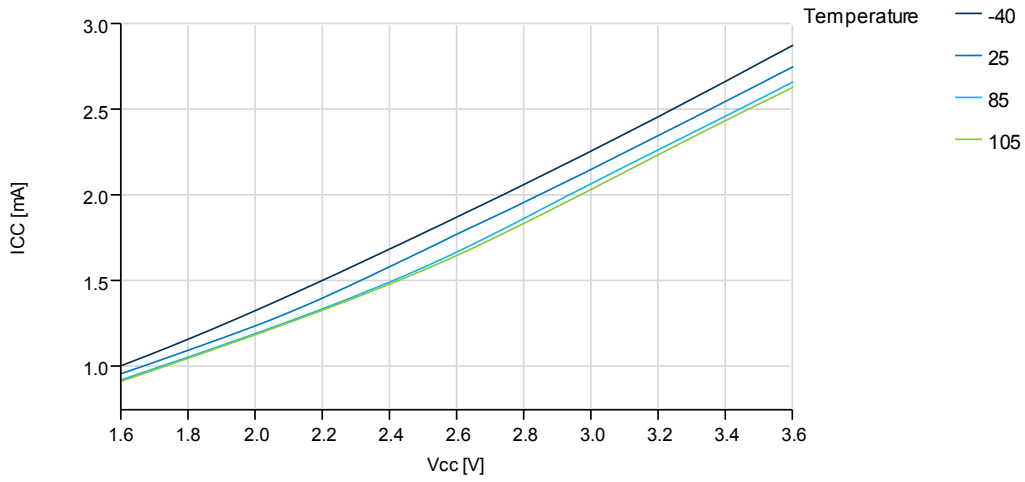
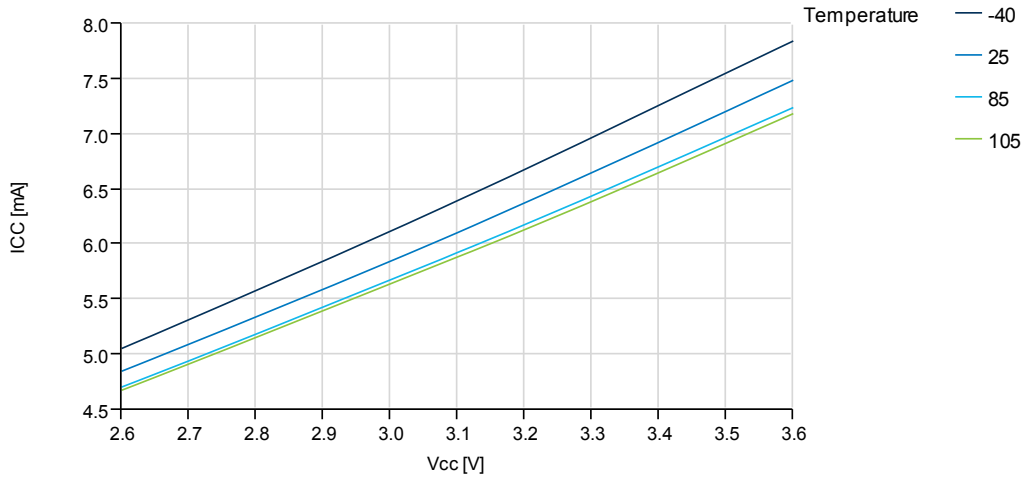


Figure 37-8. Active mode supply current vs. V_{CC}.

f_{SYS} = 32MHz internal oscillator



37.1.2 Idle Mode Supply Current

Figure 37-9. Idle mode supply current vs. frequency.

f_{SYS} = 0 - 1MHz external clock, T = 25°C.

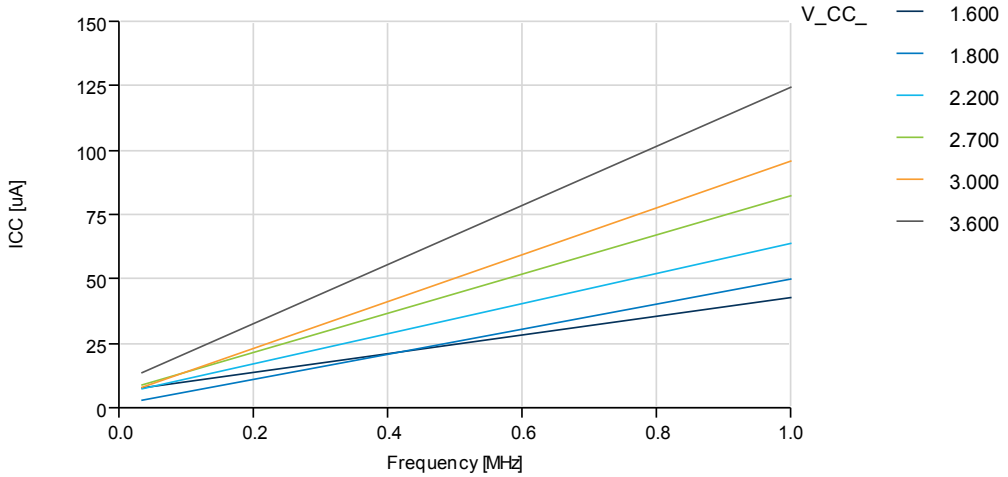


Figure 37-10.Idle mode supply current vs. frequency.

f_{SYS} = 1 - 32MHz external clock, T = 25°C.

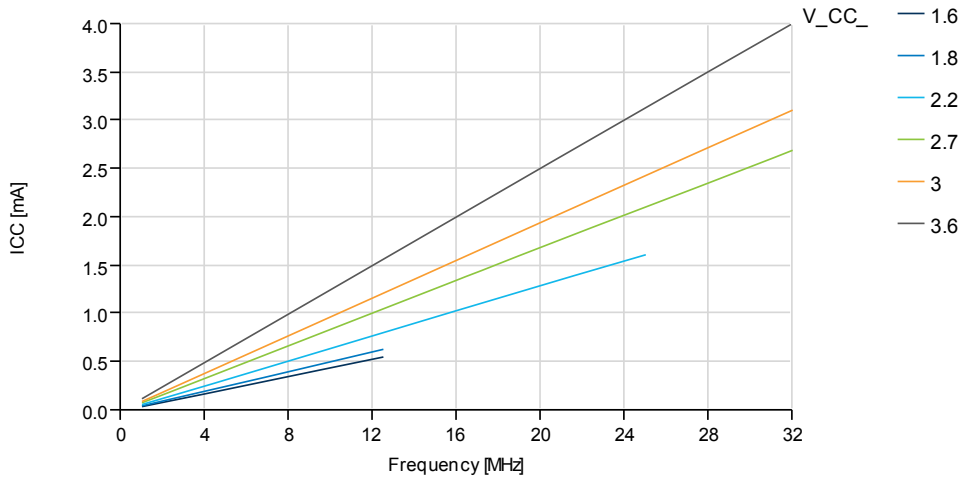


Figure 37-11.Idle mode supply current vs. V_{CC}.

f_{SYS} = 32.768kHz internal oscillator

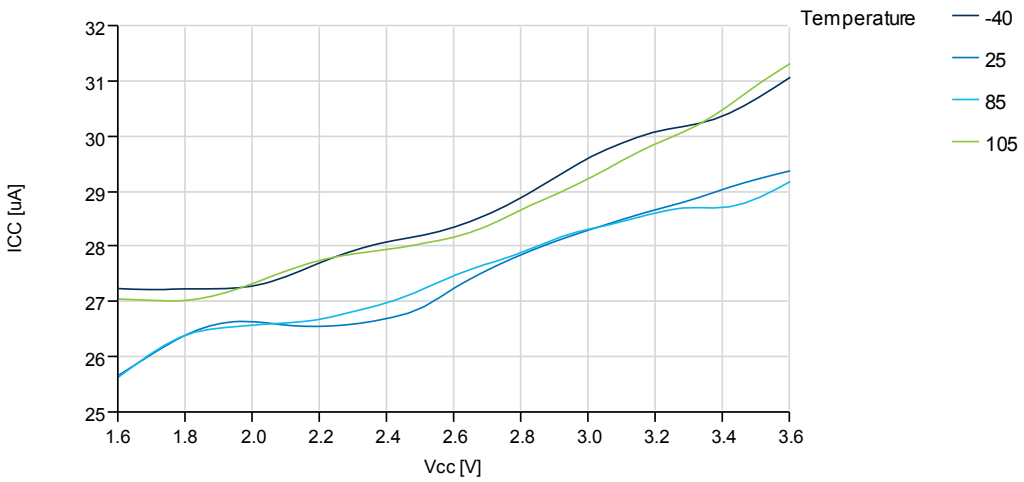


Figure 37-12.Idle mode supply current vs. V_{CC}.

f_{SYS} = 1MHz external clock

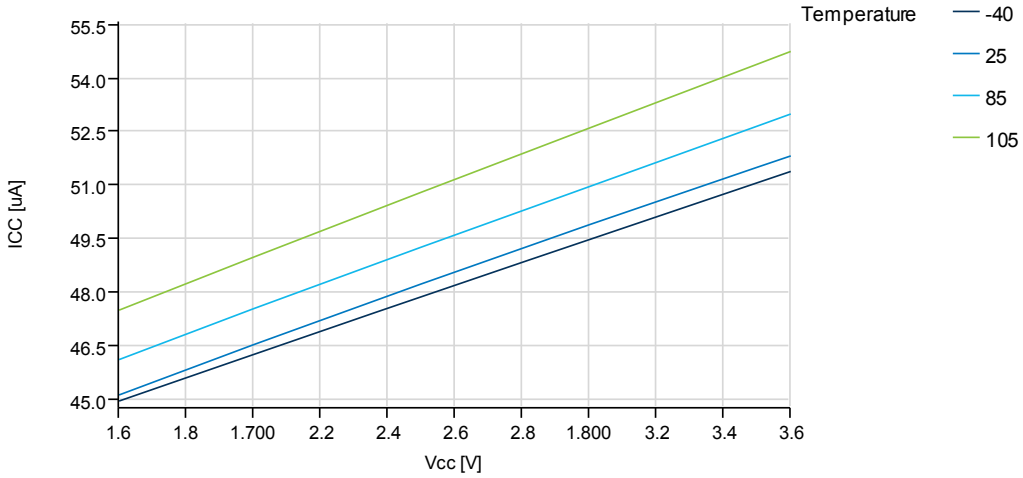


Figure 37-13.Idle mode supply current vs. V_{CC}.

f_{SYS} = 8MHz internal oscillator prescaled to 2MHz

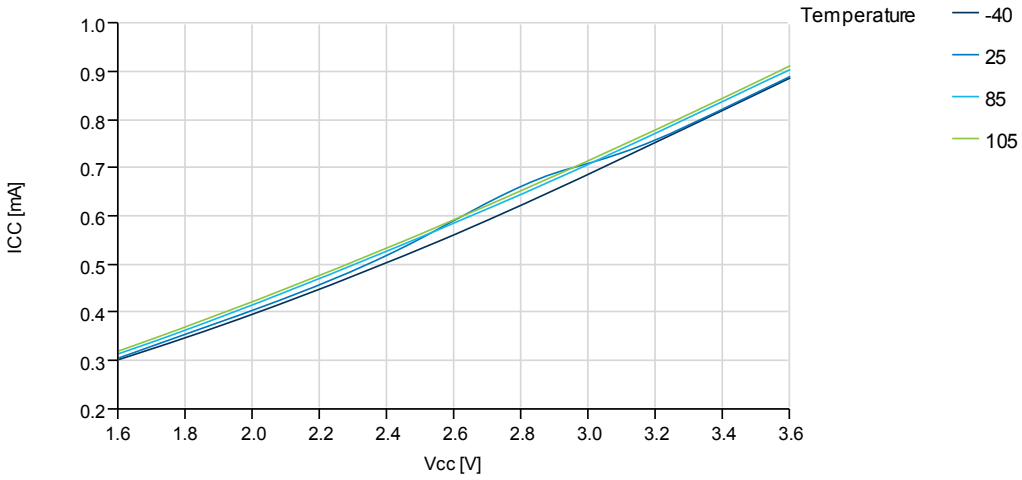


Figure 37-14.Idle mode supply current vs. V_{CC}.

f_{SYS} = 8MHz internal oscillator.

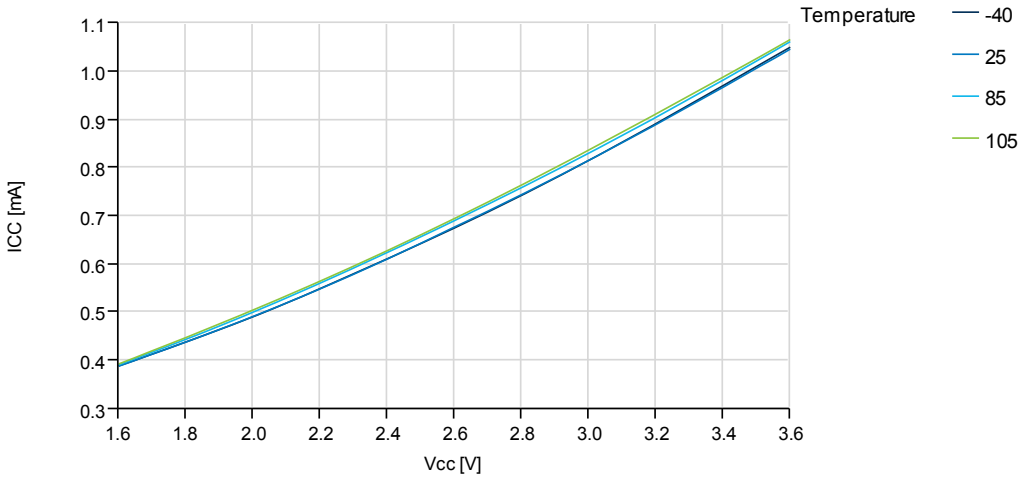


Figure 37-15.Idle mode supply current vs. V_{CC}.

f_{SYS} = 32MHz internal oscillator prescaled to 8MHz.

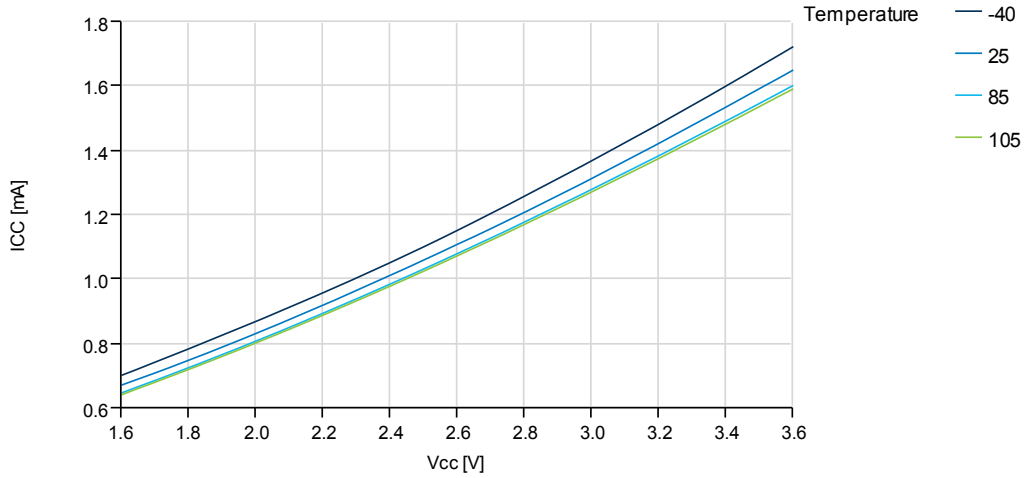
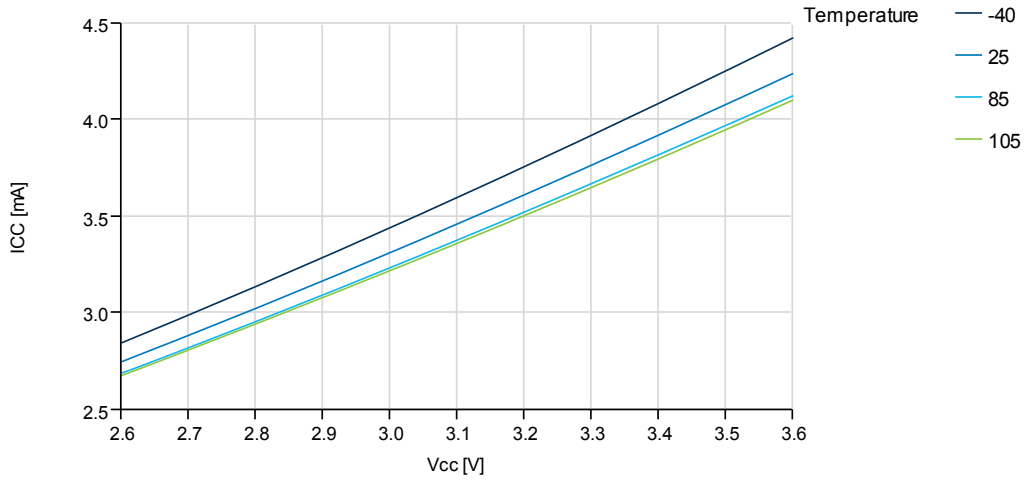


Figure 37-16.Idle mode supply current vs. V_{CC}.

f_{SYS} = 32MHz internal oscillator.



37.1.3 Power-down Mode Supply Current

Figure 37-17.Power-down mode supply current vs. temperature.

All functions disabled.

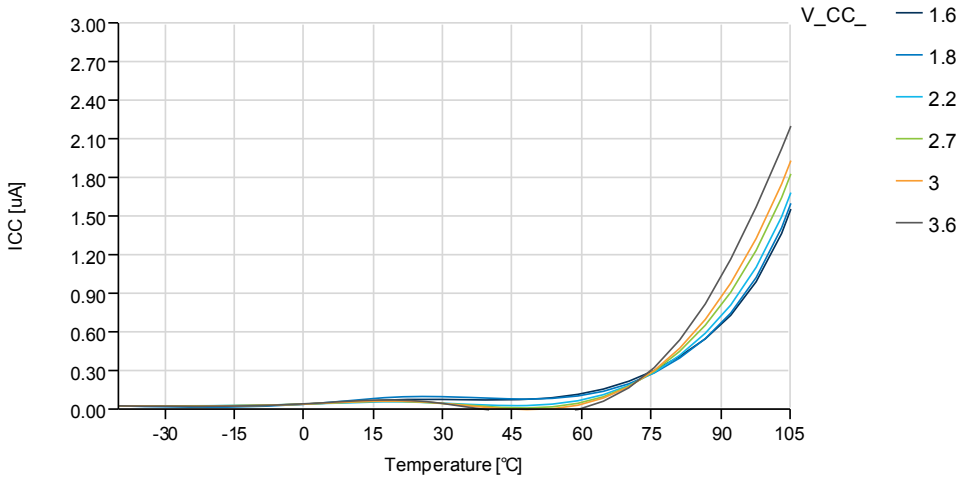


Figure 37-18.Power-down mode supply current vs. V_{CC}.

All functions disabled.

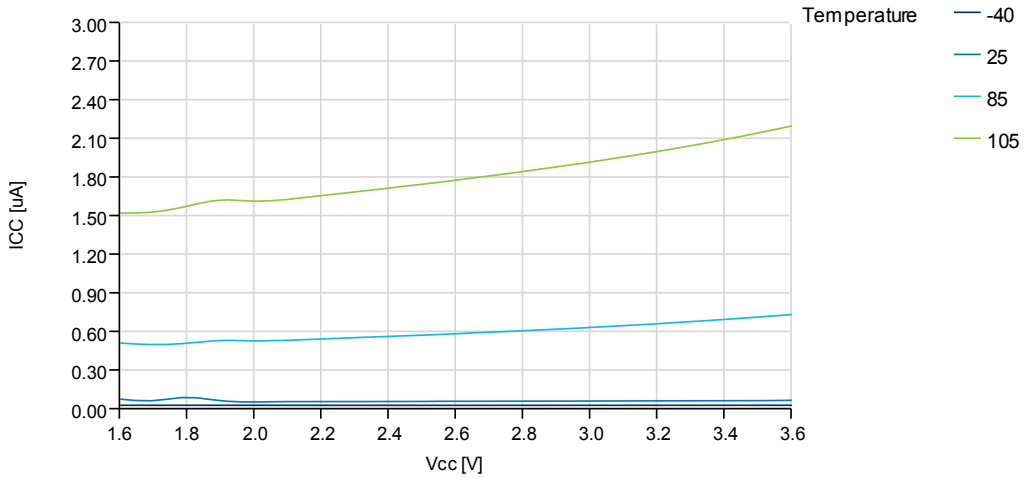
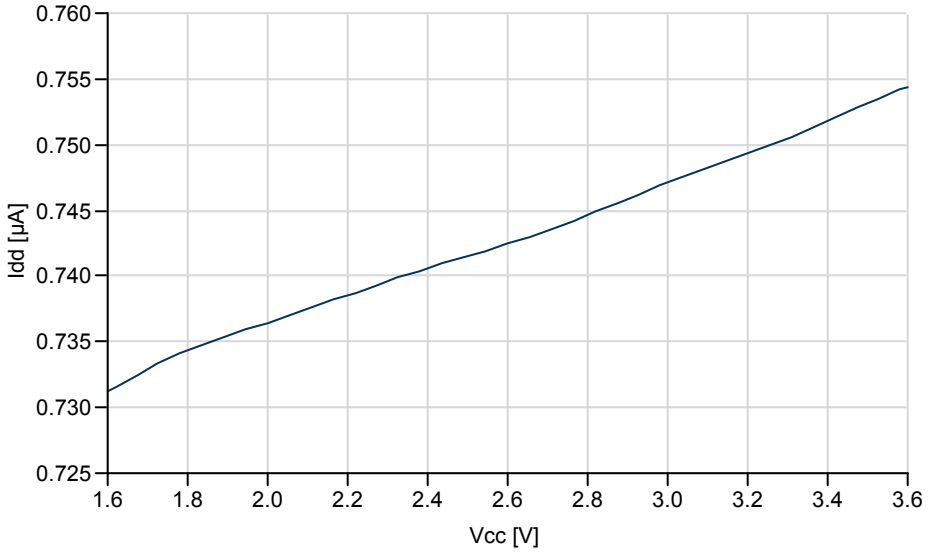


Figure 37-19.Power-down mode supply current vs. temperature.

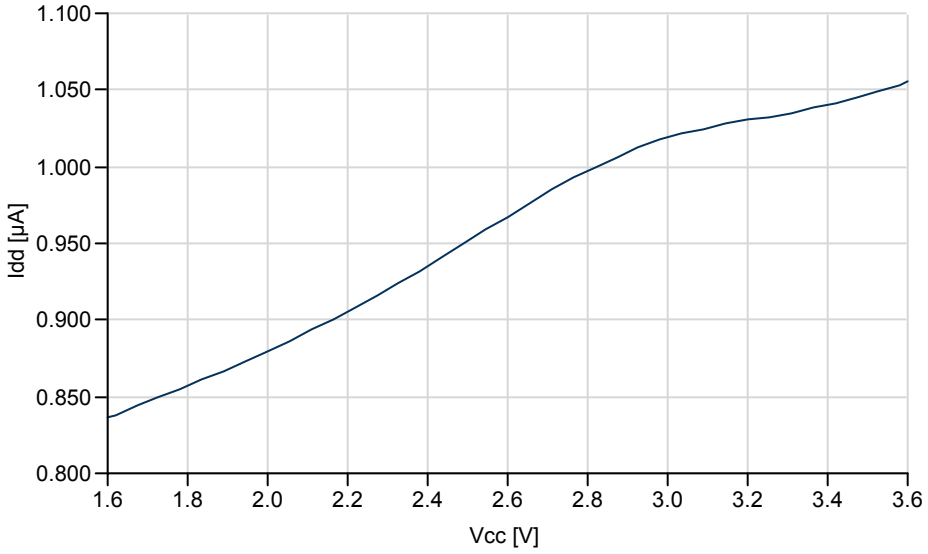
Sampled BOD with Watchdog Timer running on ULP oscillator.



37.1.4 Power-save Mode Supply Current

Figure 37-20.Power-save mode supply current vs. V_{CC}.

Real Time Counter enabled and running from 1.024kHz output of 32.768kHz TOSC.



37.1.5 Standby Mode Supply Current

Figure 37-21.Standby supply current vs. V_{CC}.

Standby, f_{SYS} = 1MHz.

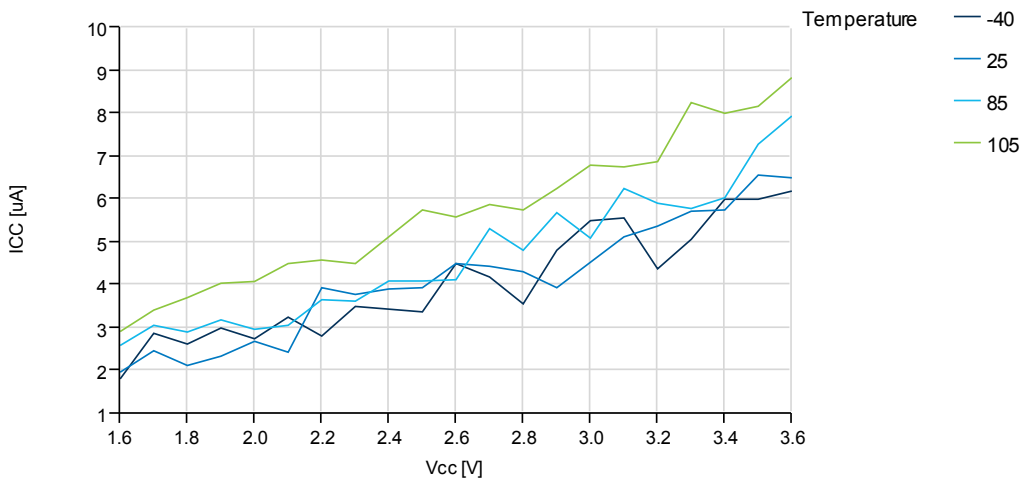
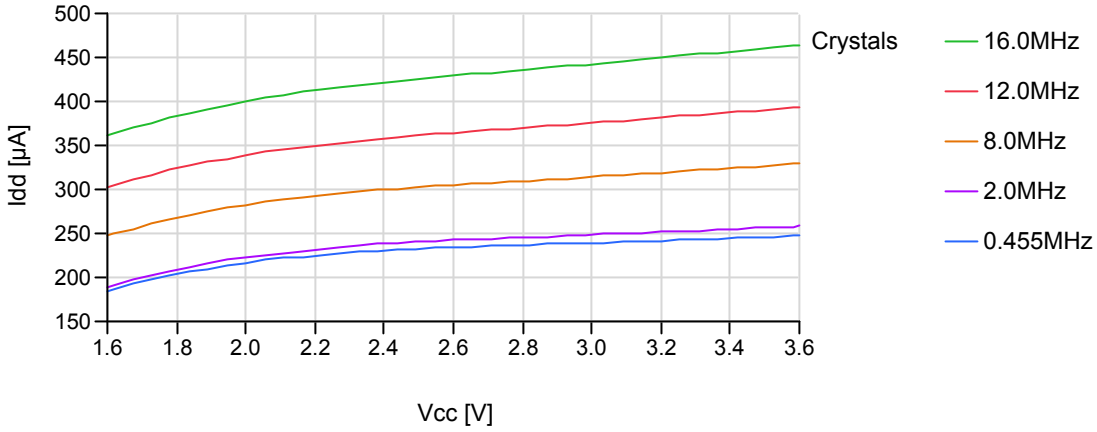


Figure 37-22.Standby supply current vs. V_{CC}.

25°C, running from different crystal oscillators.



37.2 I/O Pin Characteristics

37.2.1 Pull-up

Figure 37-23.I/O pin pull-up resistor current vs. input voltage.

V_{CC} = 1.8V.

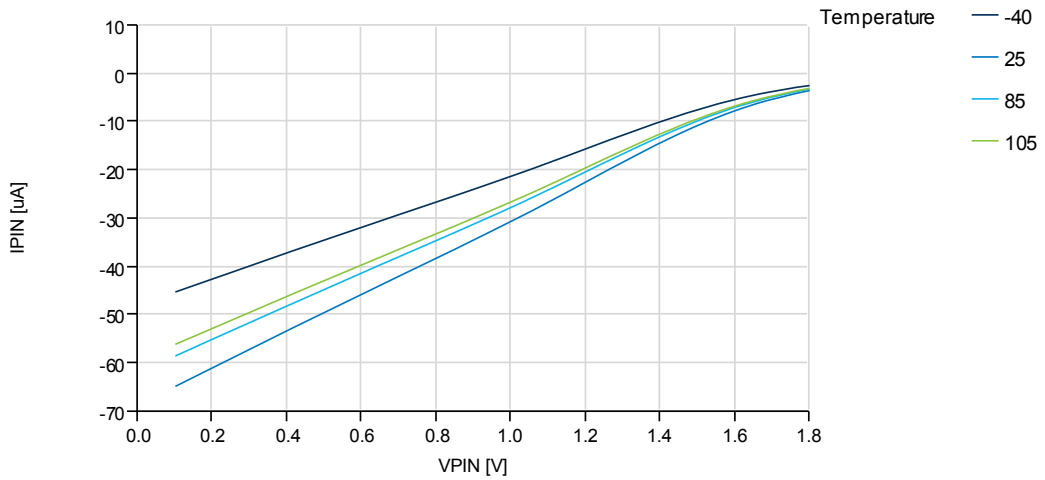


Figure 37-24.I/O pin pull-up resistor current vs. input voltage.

V_{CC} = 3.0V.

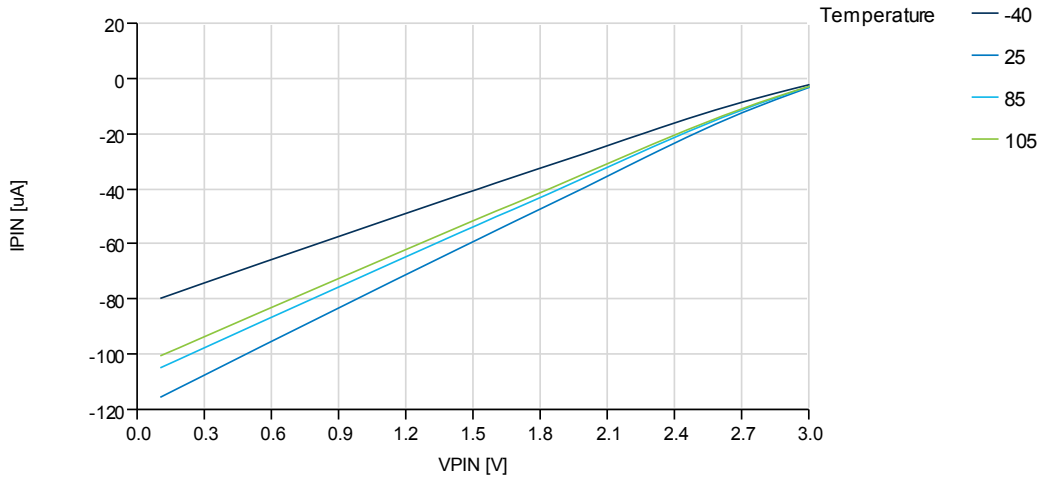
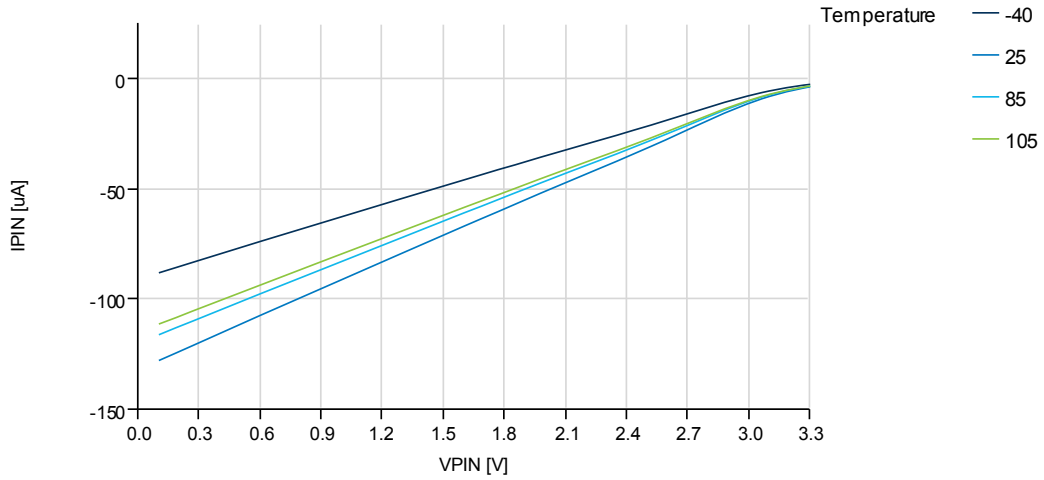


Figure 37-25.I/O pin pull-up resistor current vs. input voltage.

V_{CC} = 3.3V.



37.2.2 Output Voltage vs. Sink/Source Current

Figure 37-26.I/O pin output voltage vs. source current.

V_{CC} = 1.8V.

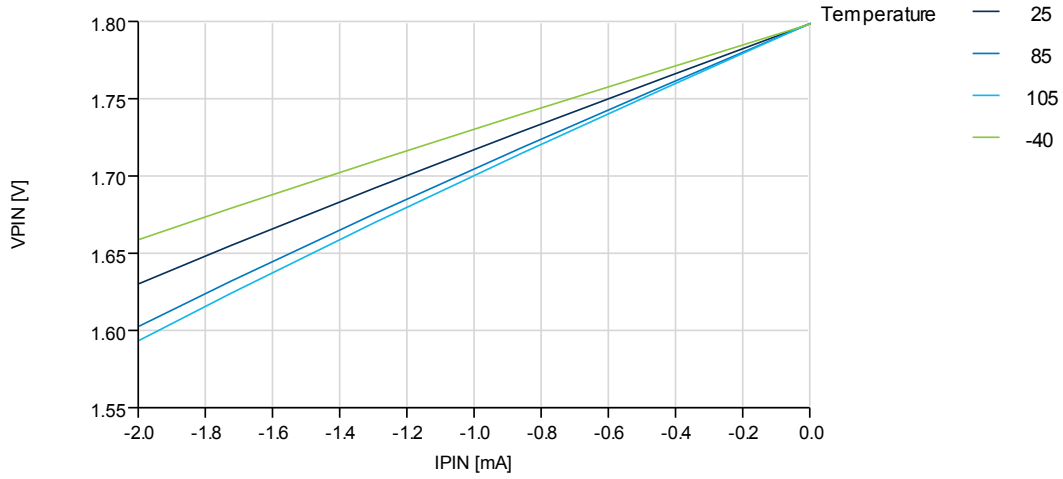


Figure 37-27.I/O pin output voltage vs. source current.

V_{CC} = 3.0V.

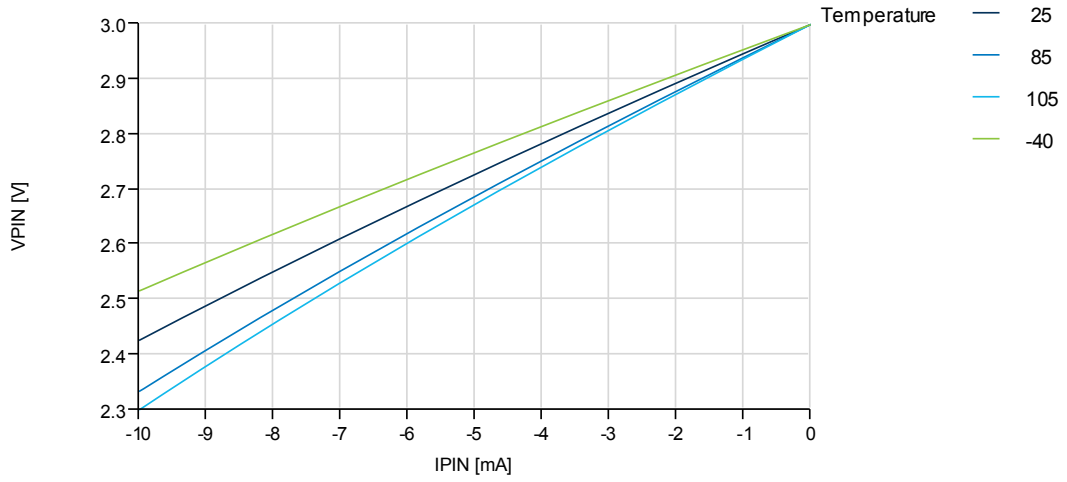


Figure 37-28.I/O pin output voltage vs. source current.

V_{CC} = 3.3V.

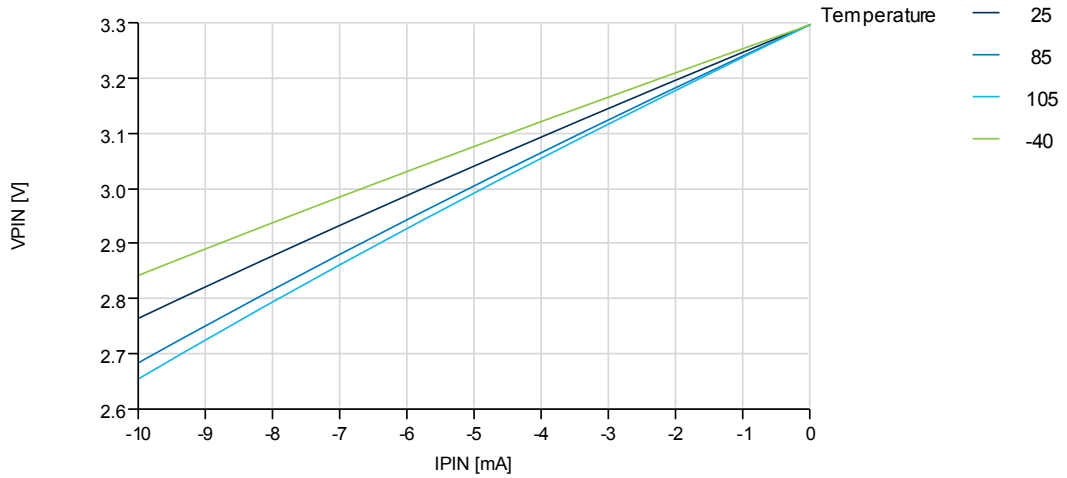


Figure 37-29.I/O pin output voltage vs. source current.

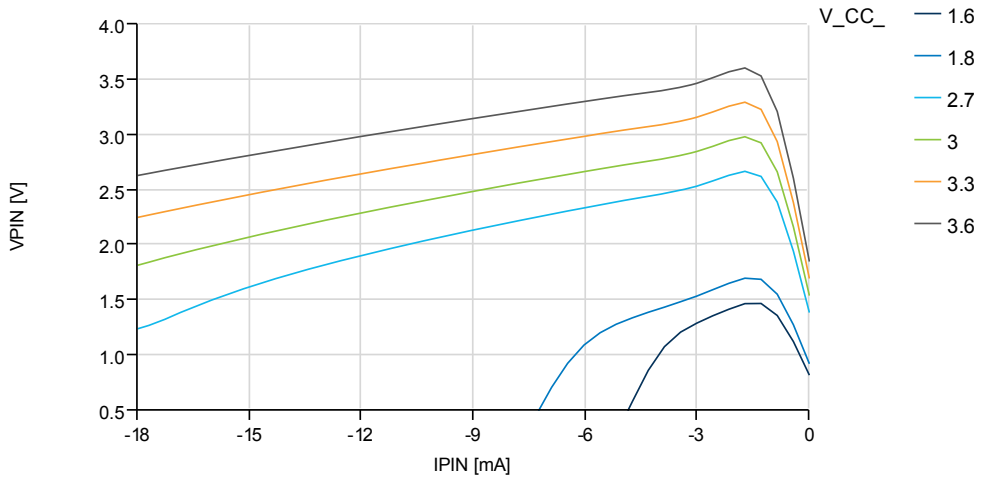


Figure 37-30.I/O pin output voltage vs. sink current.

V_{CC} = 1.8V.

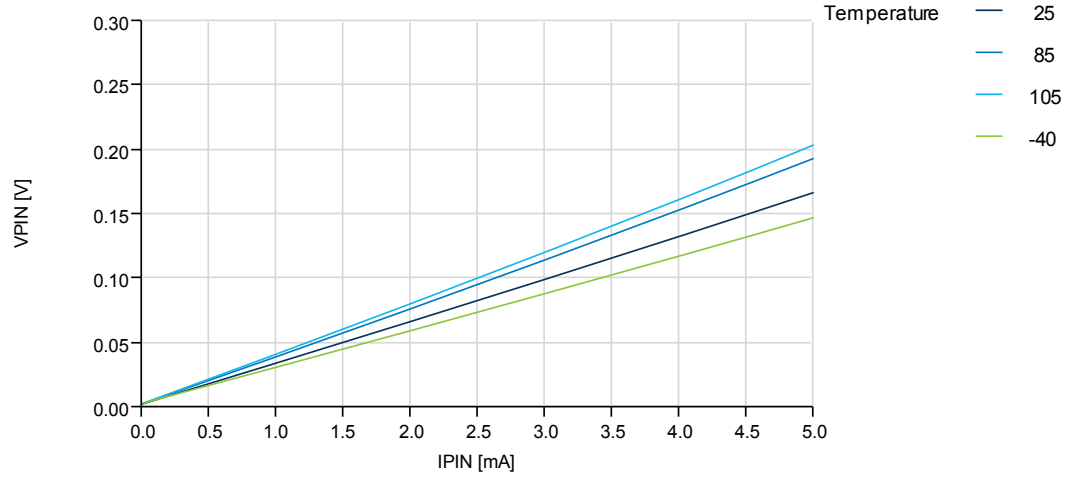


Figure 37-31.I/O pin output voltage vs. sink current.

V_{CC} = 3.0V.

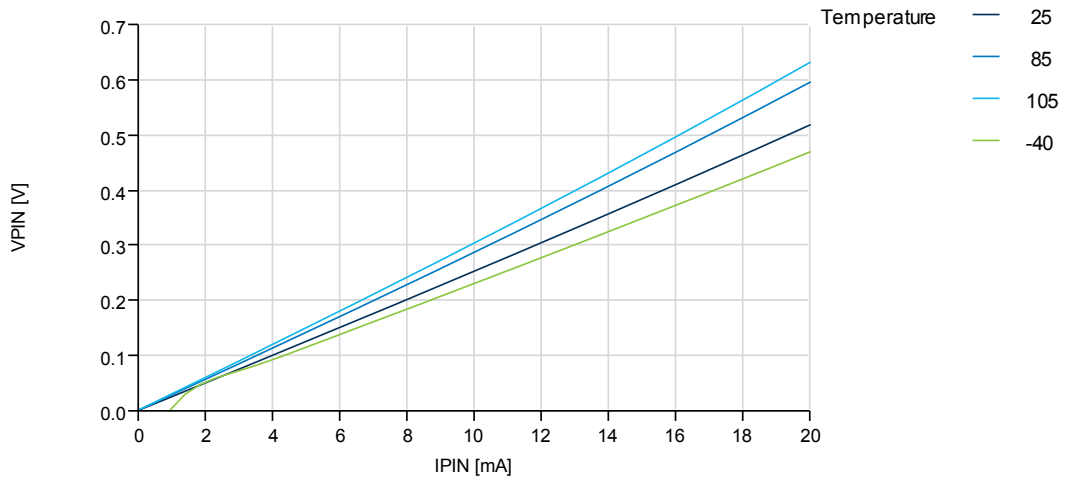


Figure 37-32.I/O pin output voltage vs. sink current.

V_{CC} = 3.3V.

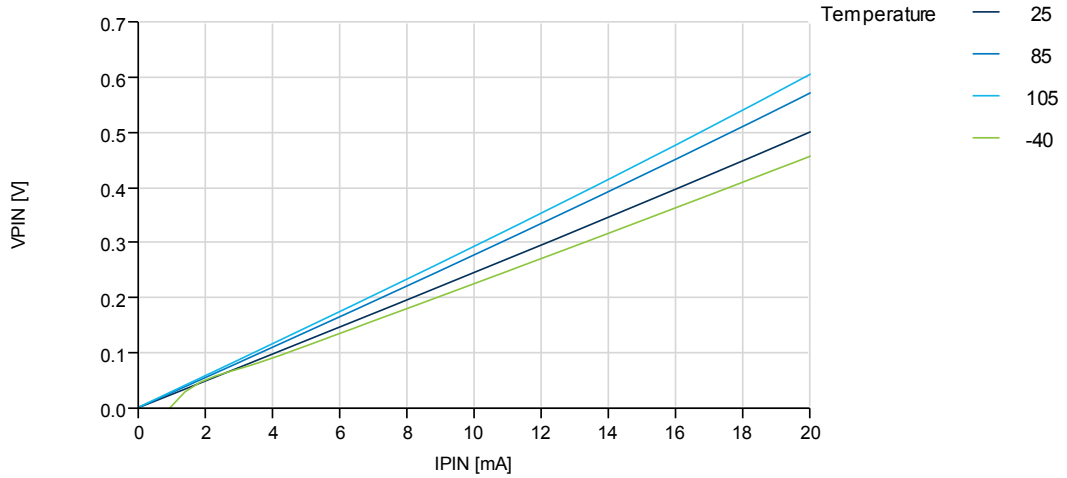
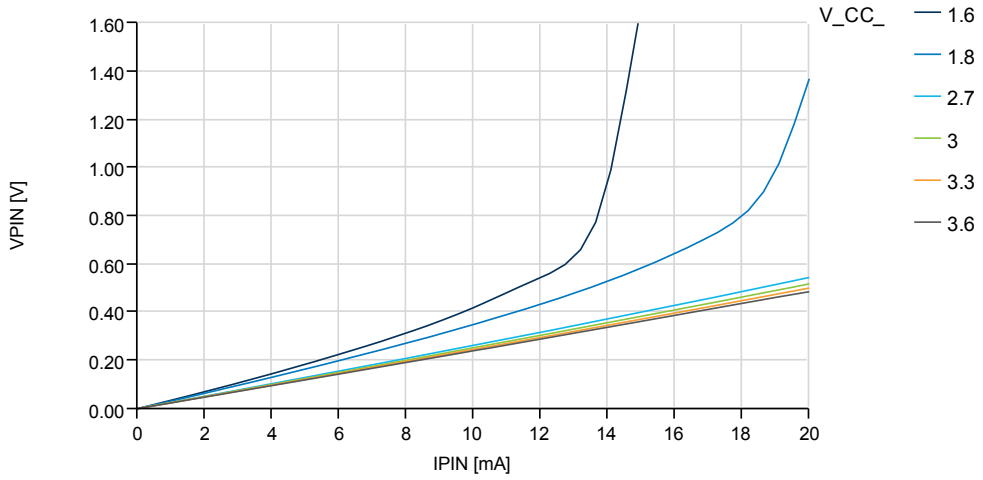


Figure 37-33.I/O pin output voltage vs. sink current.



37.2.3 Thresholds and Hysteresis

Figure 37-34.I/O pin input threshold voltage vs. V_{CC}.

T = 25°C.

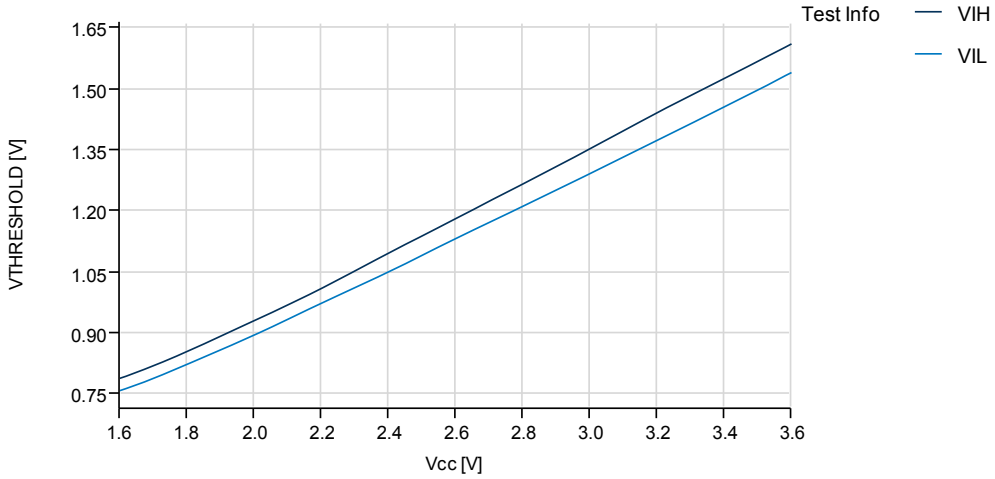


Figure 37-35.I/O pin input threshold voltage vs. V_{CC}.

V_{IH} I/O pin read as "1"

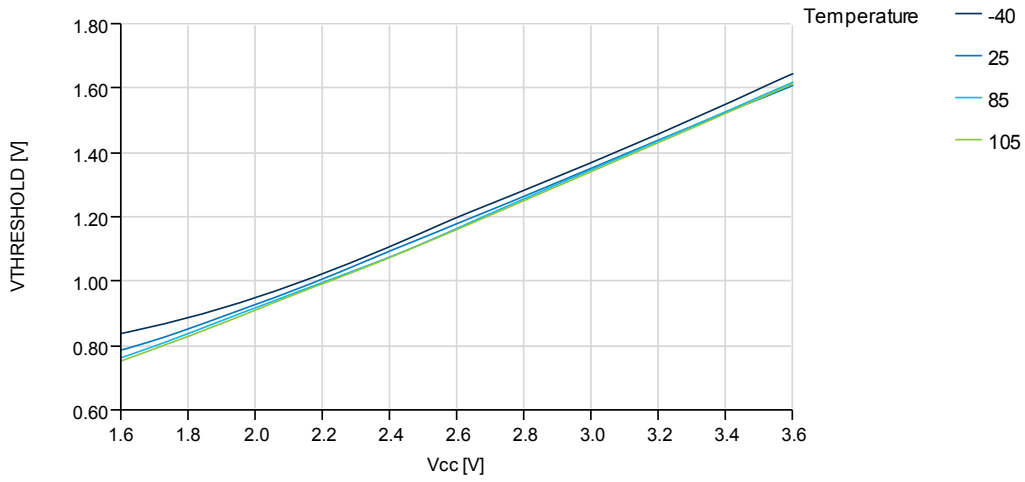


Figure 37-36.I/O pin input threshold voltage vs. V_{CC}.

V_{IL} I/O pin read as "0"

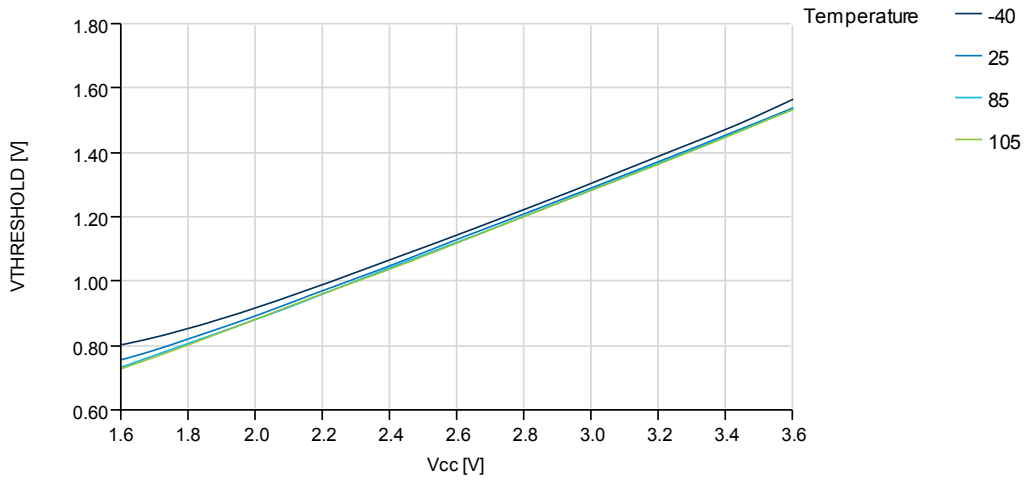
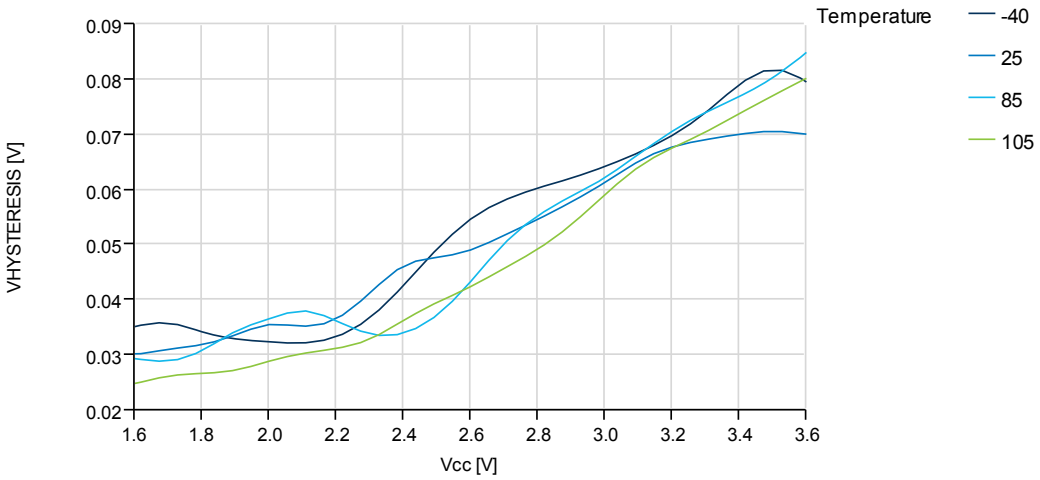


Figure 37-37.I/O pin input hysteresis vs. V_{CC}.



37.3 ADC Characteristics

Figure 37-38.ADC INL vs. V_{REF}.

T = 25°C, V_{CC} = 3.6V, external reference.

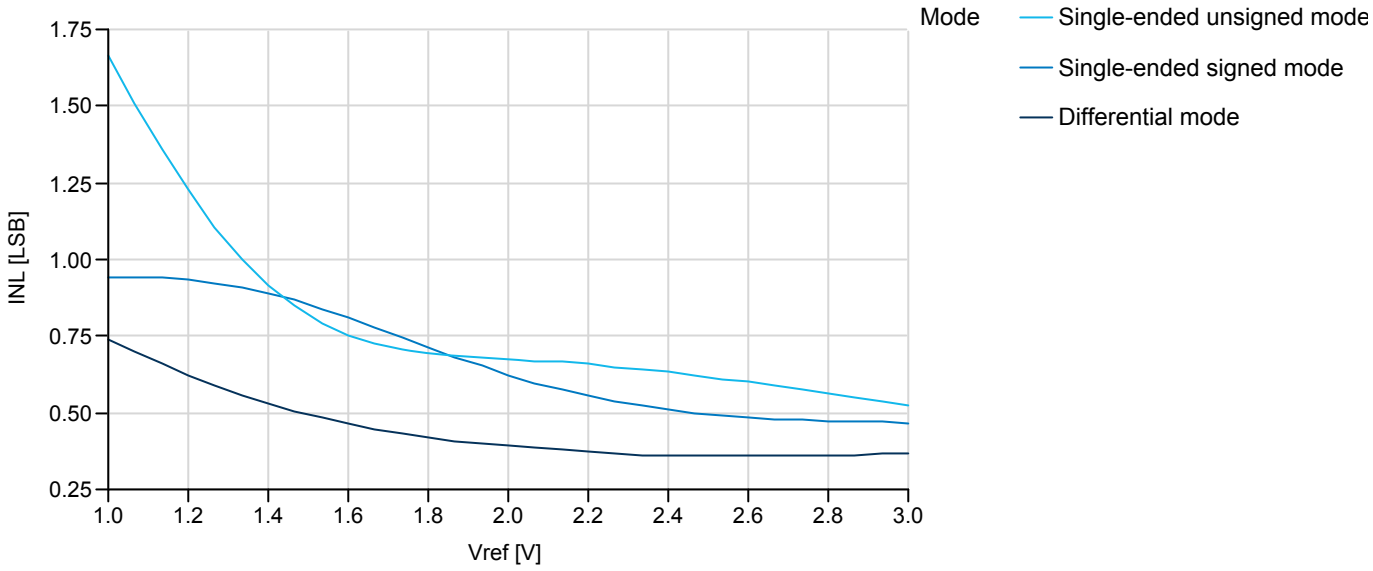


Figure 37-39.ADC INL error vs. V_{CC}.

T = 25°C, V_{REF} = 1.0V

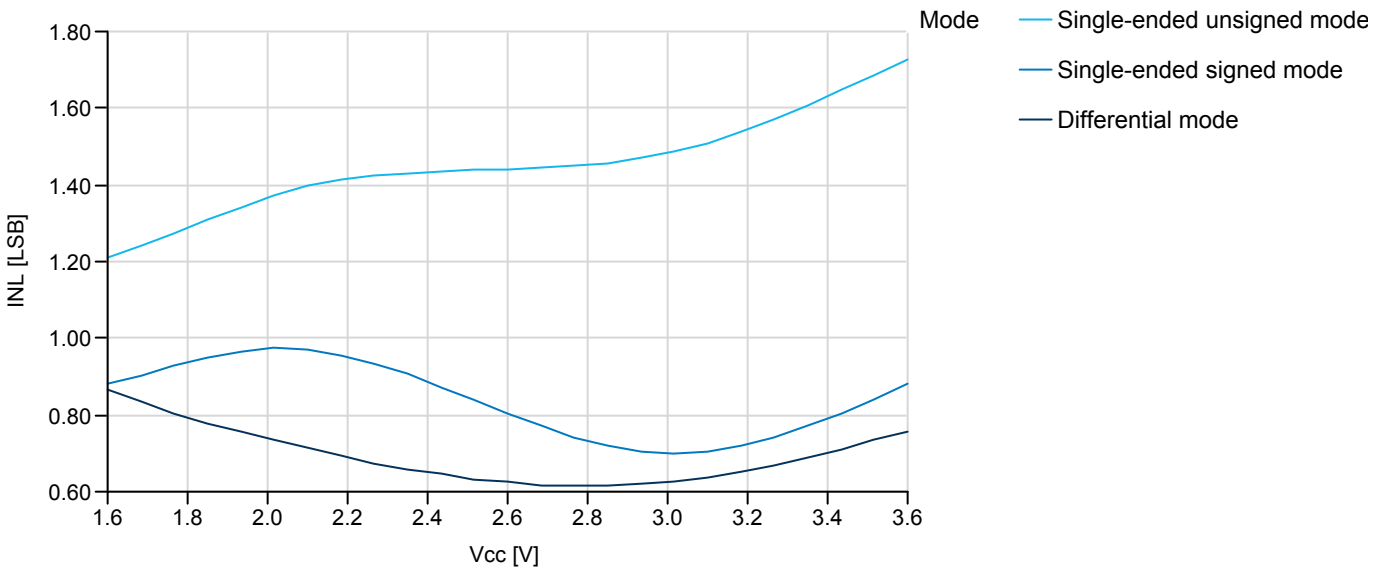


Figure 37-40.ADC DNL error vs. V_{REF}.

SE Unsigned mode, T=25°C, V_{CC} = 3.6V, external reference.

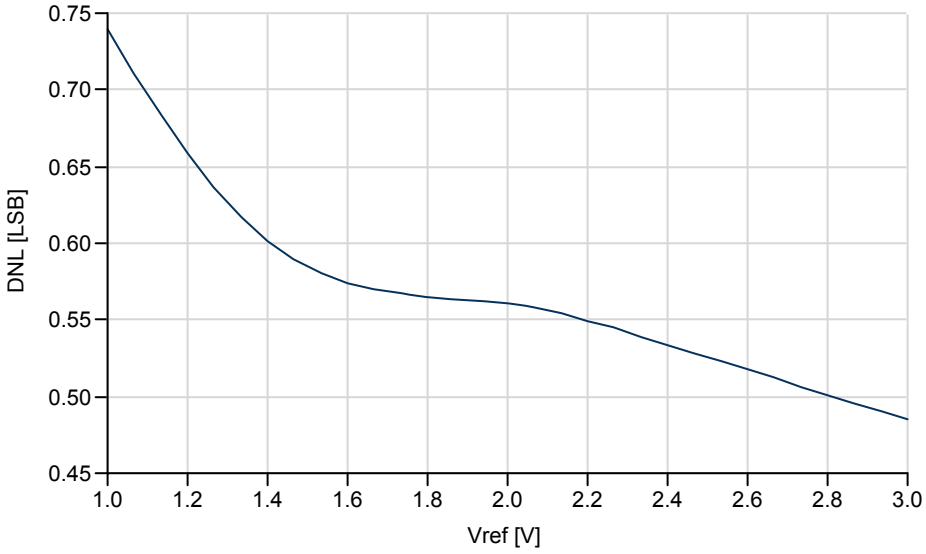


Figure 37-41. ADC gain error vs. V_{CC}.

T = 25°C, V_{REF} = 1.0V, ADC sample rate = 300ksps.

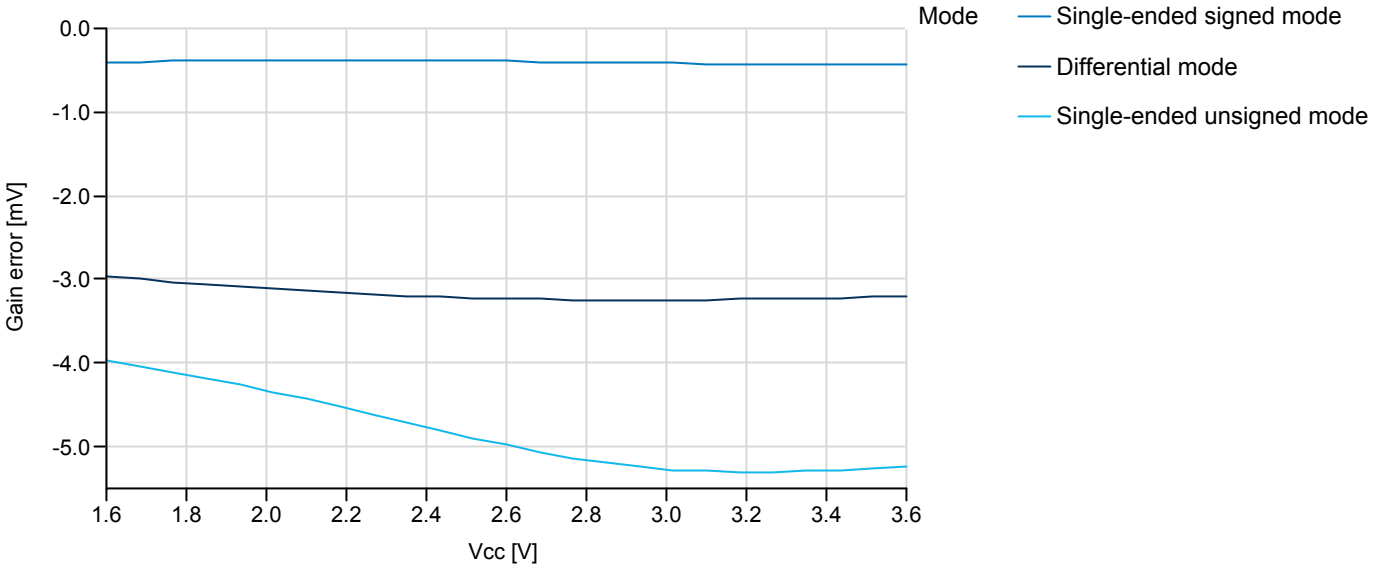


Figure 37-42. ADC gain error vs. V_{REF}.

T = 25°C, V_{CC} = 3.6V, ADC sample rate = 300ksps

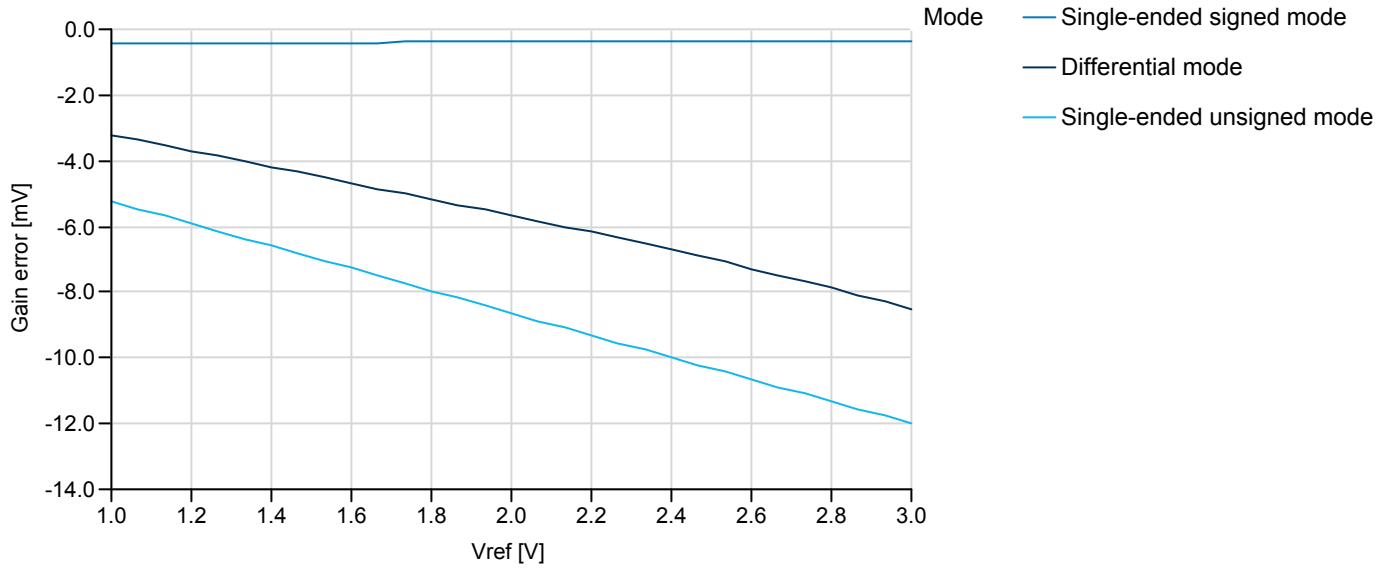


Figure 37-43. ADC gain error vs. temperature.

V_{CC} = 3.6V, V_{REF} = 1.0V, ADC sample rate = 300ksps

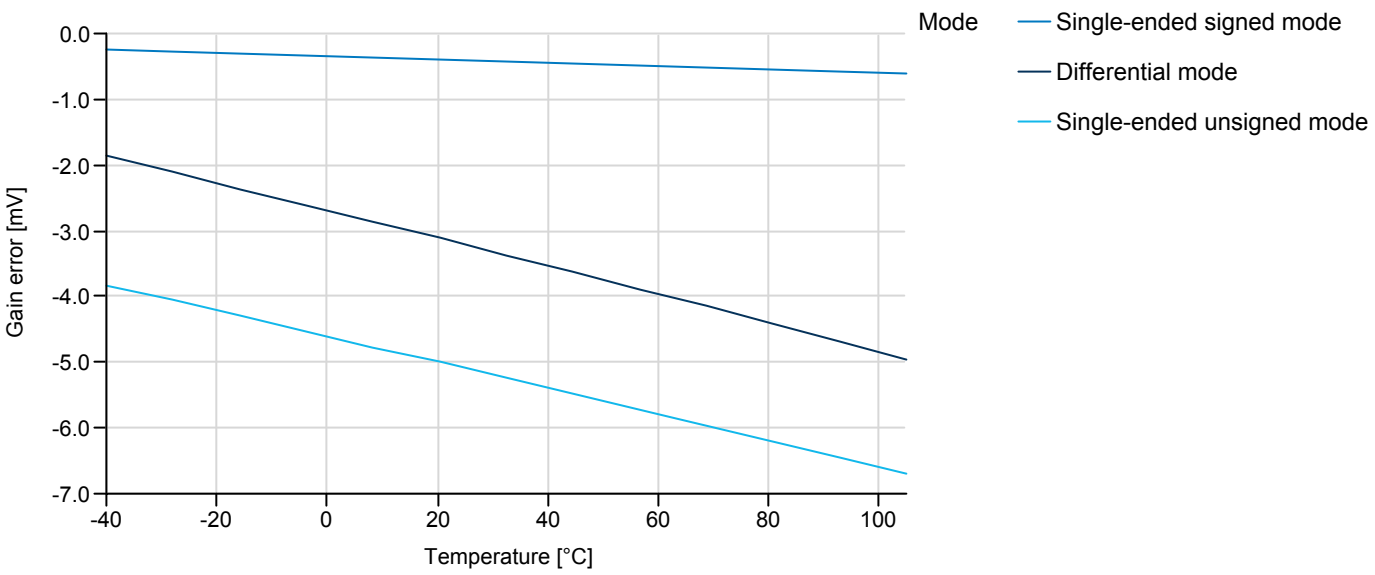


Figure 37-44. ADC offset error vs. V_{CC}.

T = 25°C, V_{REF} = 1.0V, ADC sample rate = 300ksps

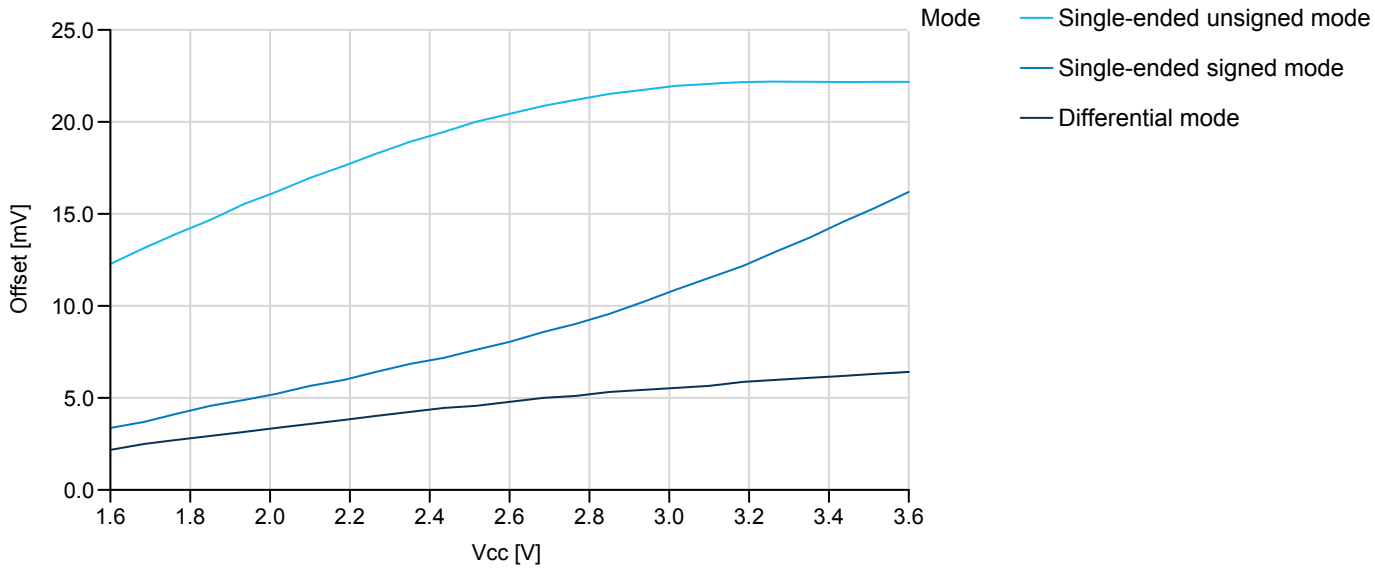


Figure 37-45. ADC offset error vs. V_{REF}.

T = 25°C, V_{CC} = 3.6V, ADC sample rate = 300ksps

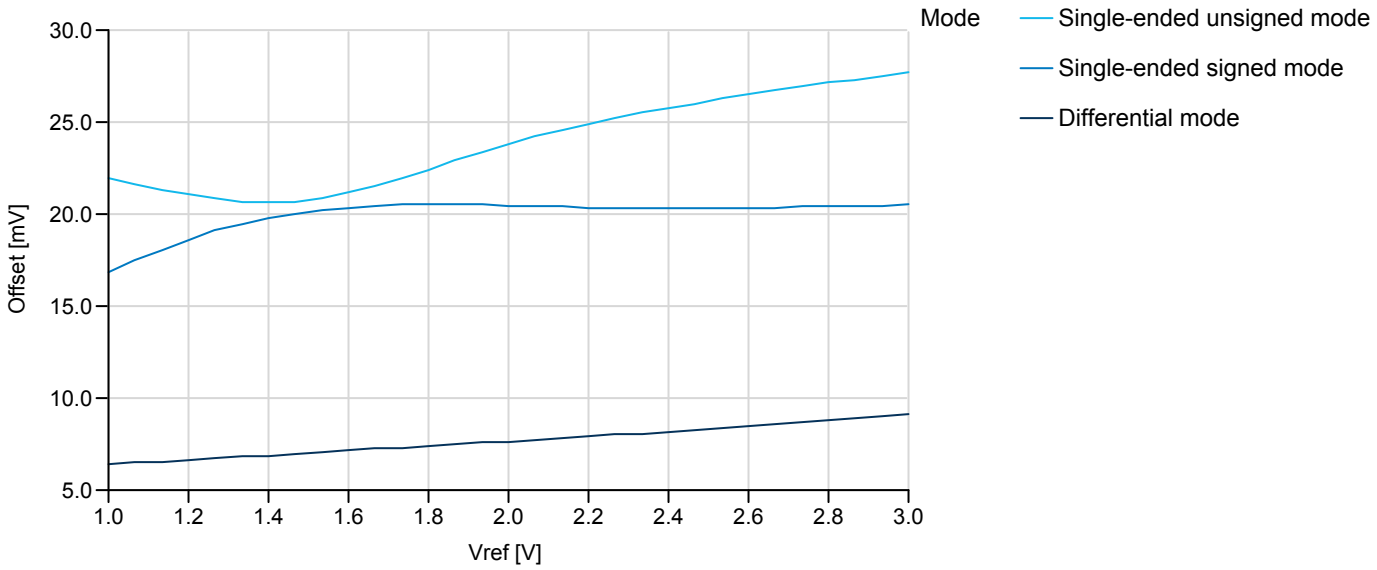
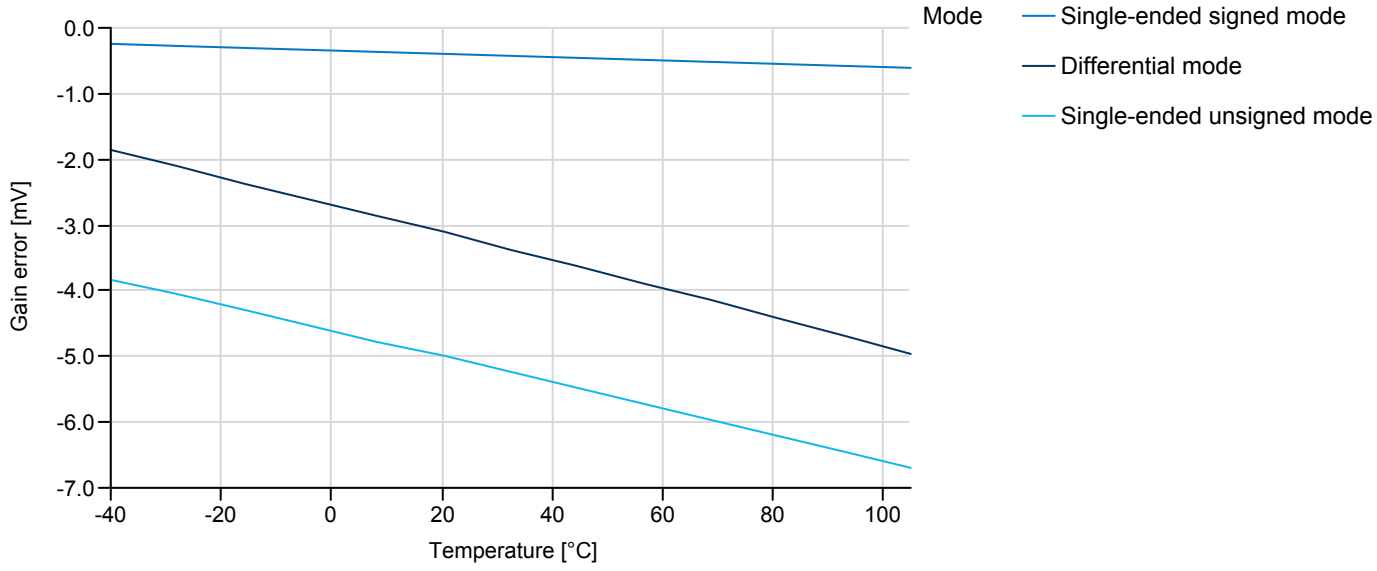


Figure 37-46.ADC gain error vs. temperature.

VCC = 3.6V, VREF = external 1.0V, sample rate = 300ksps



37.4 DAC Characteristics

Figure 37-47.DAC INL error vs. external V_{REF}.

T = 25°C, V_{CC} = 3.6V

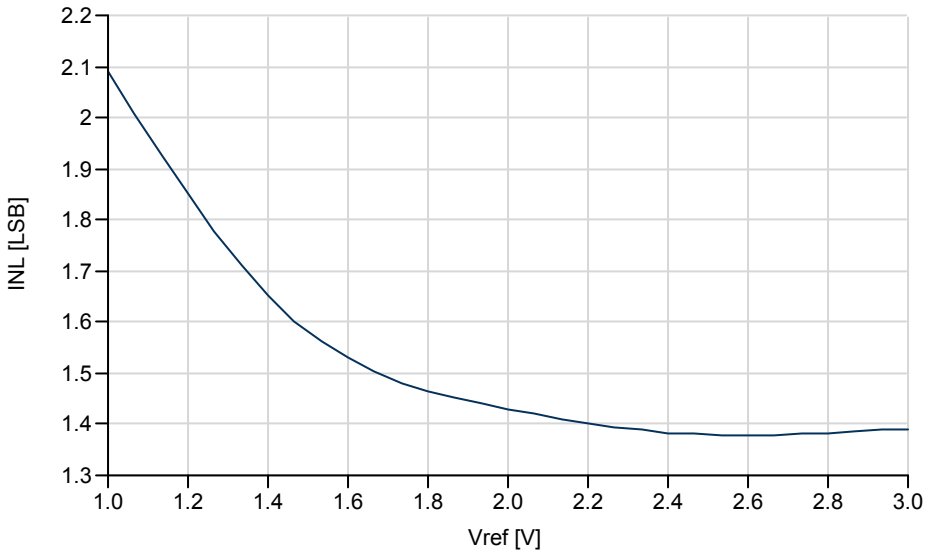


Figure 37-48.DNL error vs. V_{REF}.

T = 25°C, V_{CC} = 3.6V

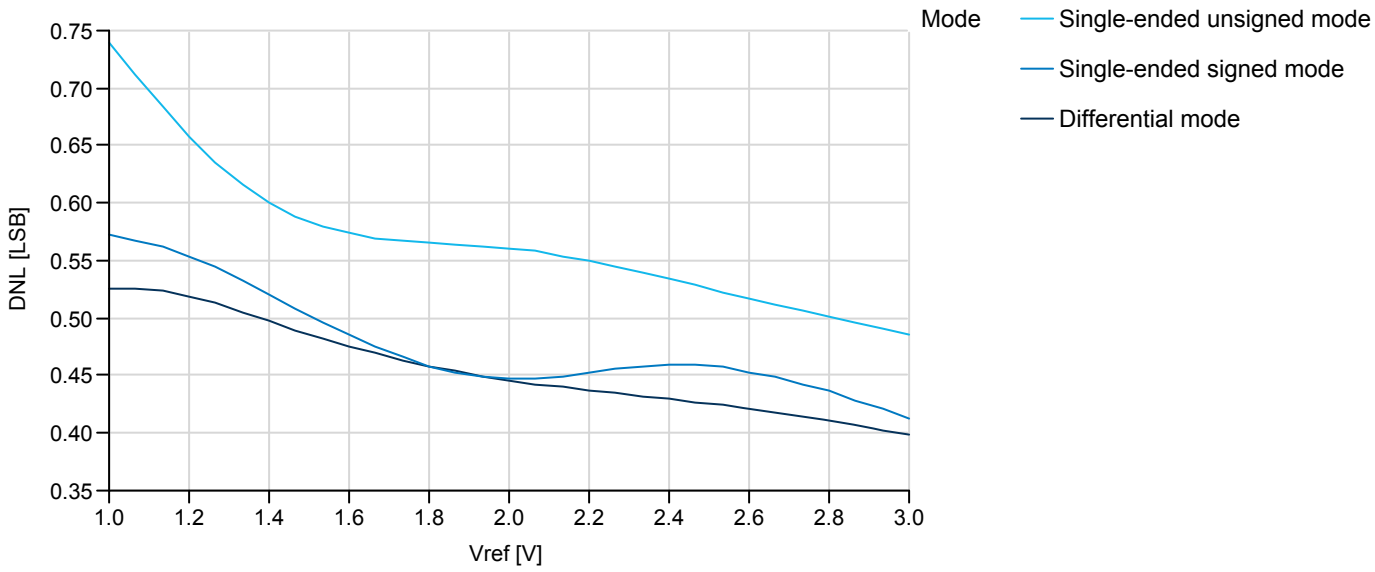
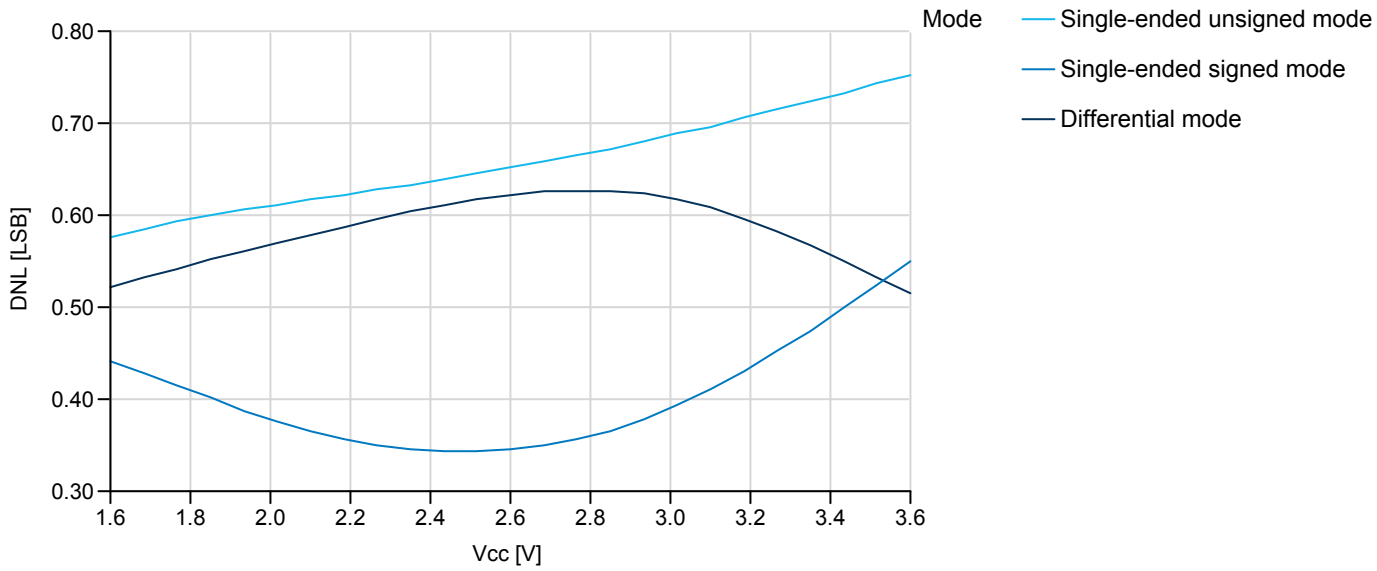


Figure 37-49.DNL error vs. V_{CC}.

T = 25°C, V_{REF} = 1.0V



37.5 AC Characteristics

Figure 37-50.Analog comparator hysteresis vs. V_{CC}.

Small hysteresis

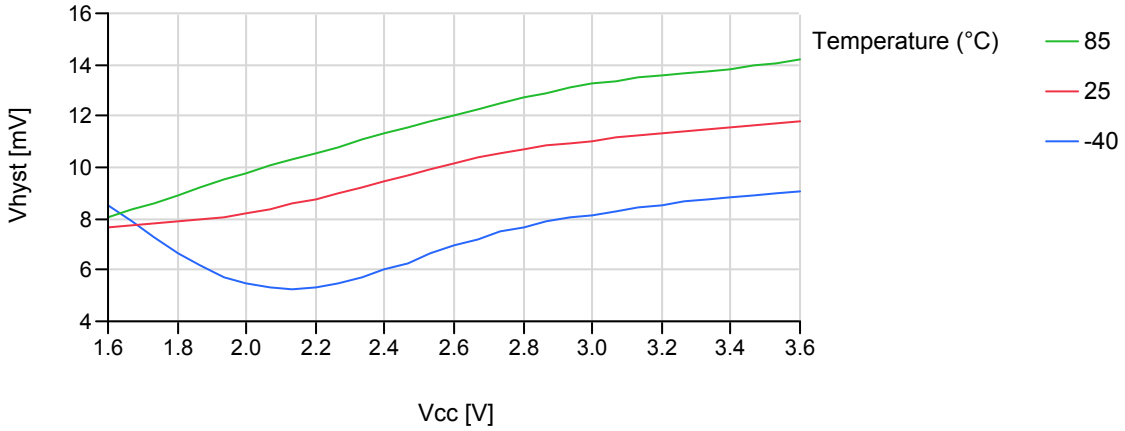


Figure 37-51.Analog comparator hysteresis vs. V_{CC}.

Large hysteresis

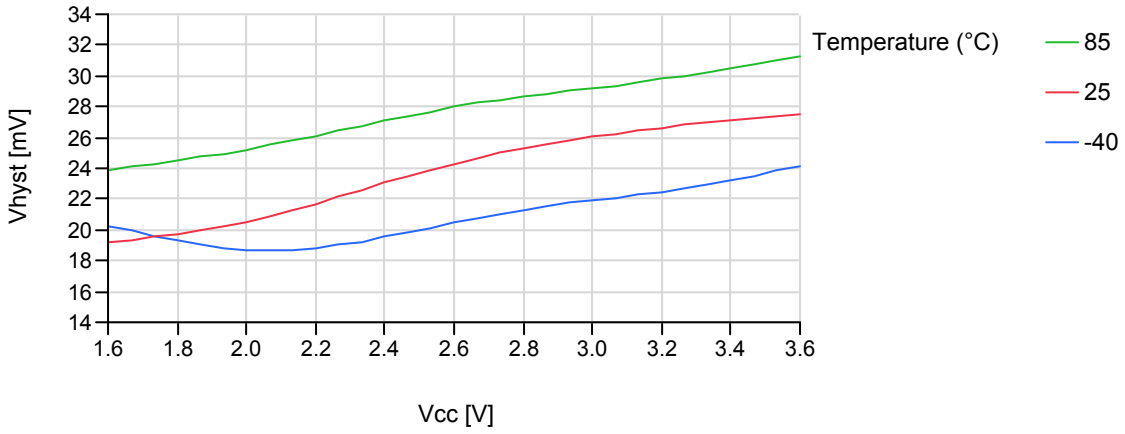


Figure 37-52.Analog comparator propagation delay vs. V_{CC}.

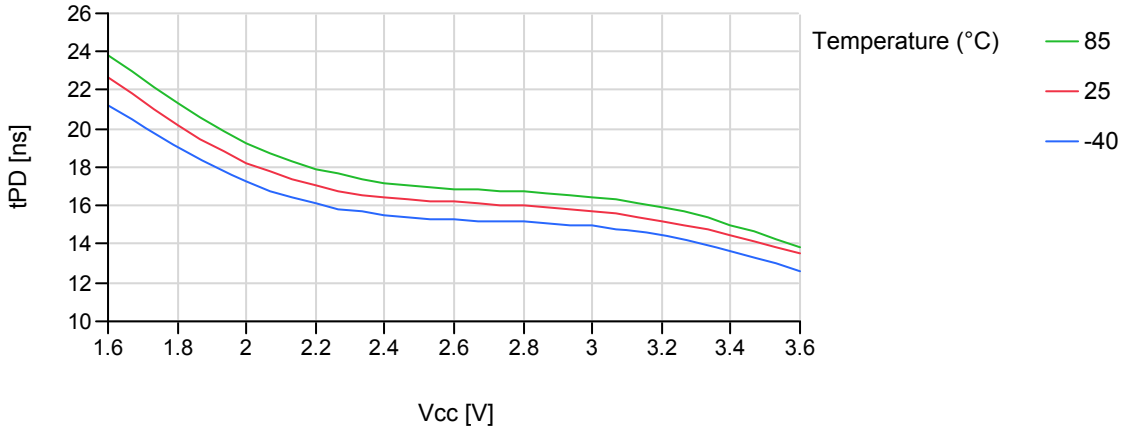


Figure 37-53.Analog comparator propagation delay vs. temperature.

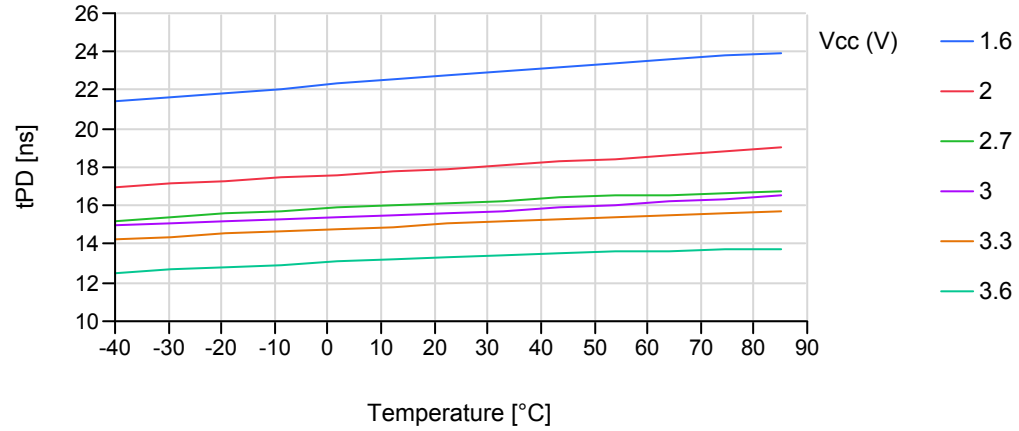


Figure 37-54.Analog comparator current consumption vs. V_{CC}.

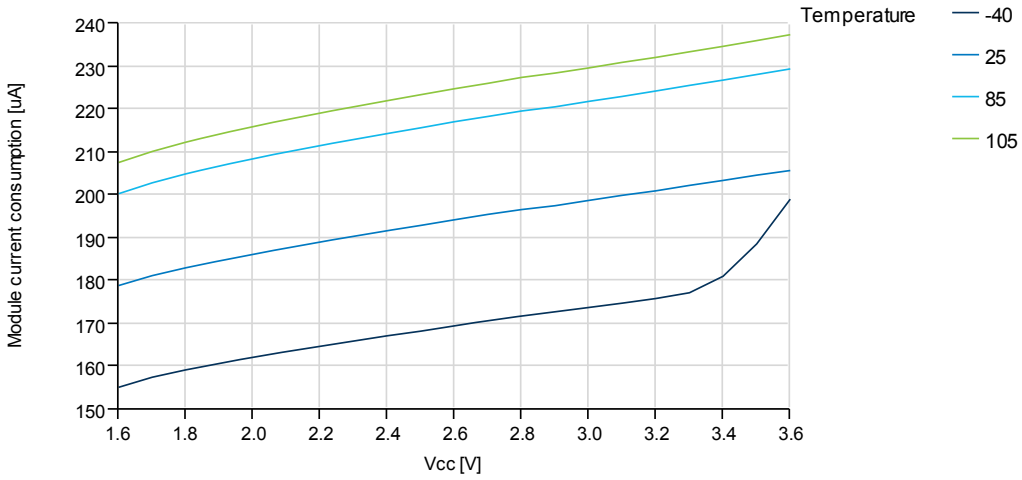


Figure 37-55.Analog comparator voltage scaler vs. SCALEFAC.

T = 25°C, V_{CC} = 3.0V

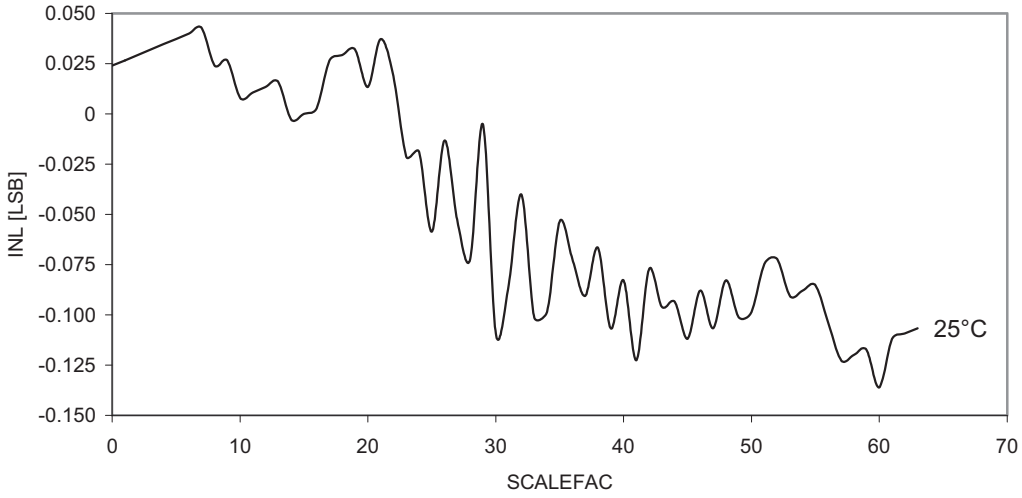


Figure 37-56.Analog comparator offset voltage vs. common mode voltage.

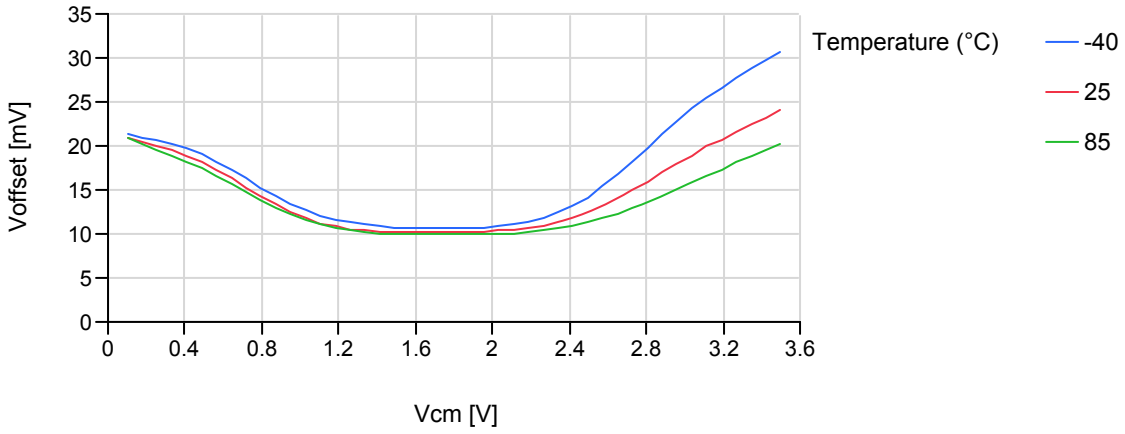


Figure 37-57.Analog comparator source vs. calibration value.

V_{CC} = 3.0V

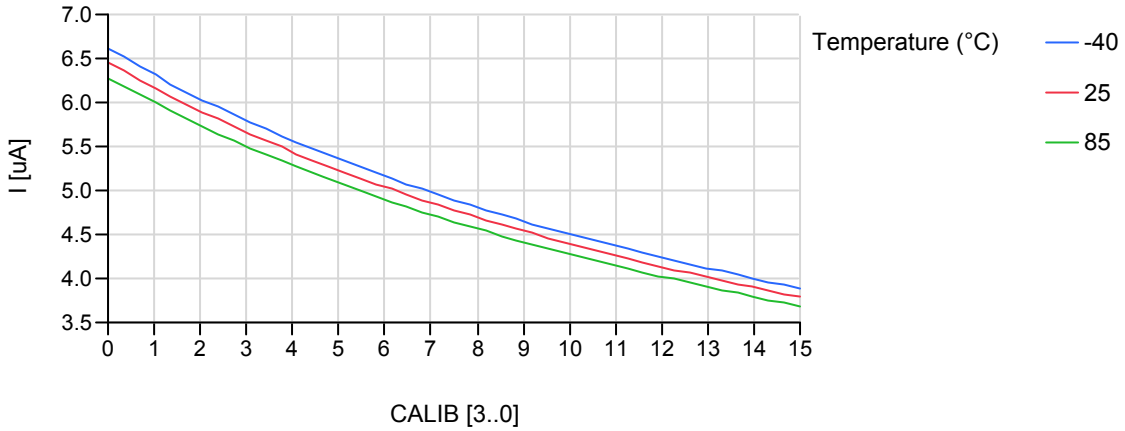
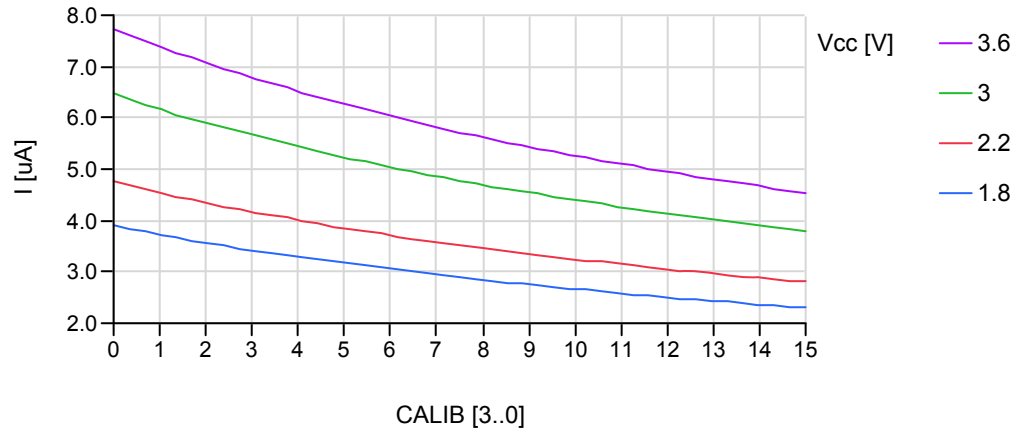


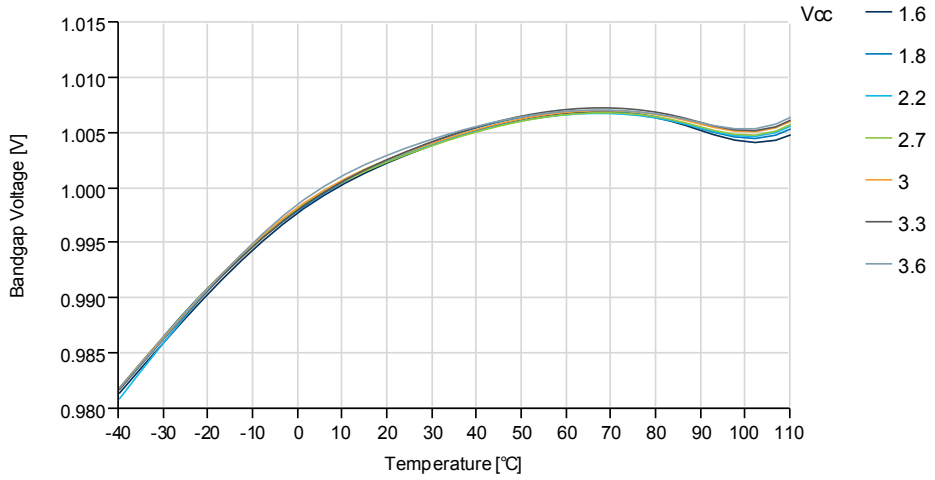
Figure 37-58.Analog comparator source vs. calibration value.

T = 25°C



37.6 Internal 1.0V Reference Characteristics

Figure 37-59.ADC/DAC internal 1.0V reference vs. temperature.



37.7 BOD Characteristics

Figure 37-60.BOD thresholds vs. temperature.

BOD level = 1.6V

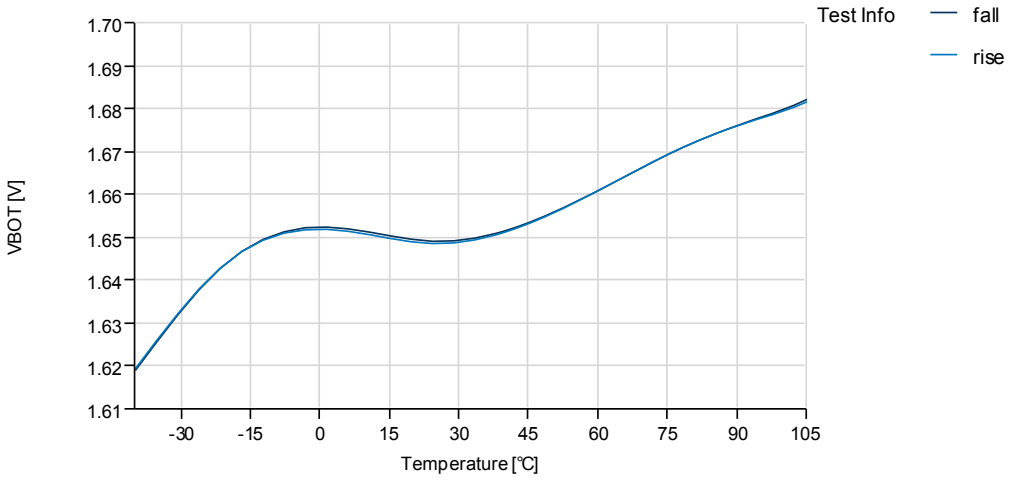
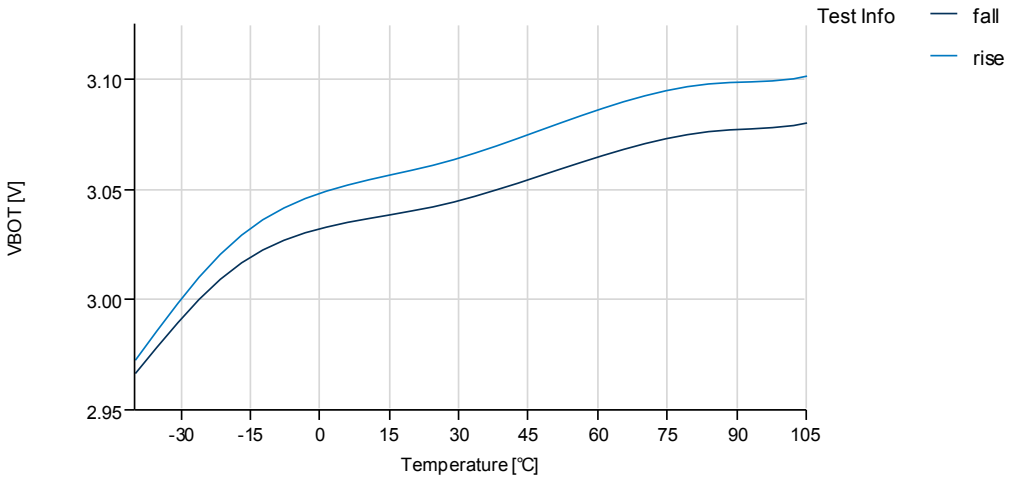


Figure 37-61.BOD thresholds vs. temperature.

BOD level = 3.0V



37.8 External Reset Characteristics

Figure 37-62.Minimum reset pin pulse width vs. V_{CC}.

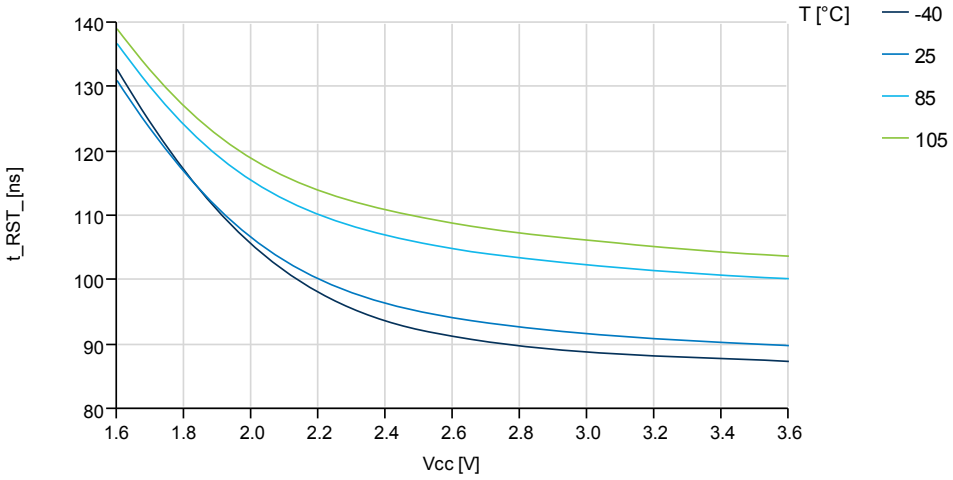


Figure 37-63.Reset pin pull-up resistor current vs. reset pin voltage.

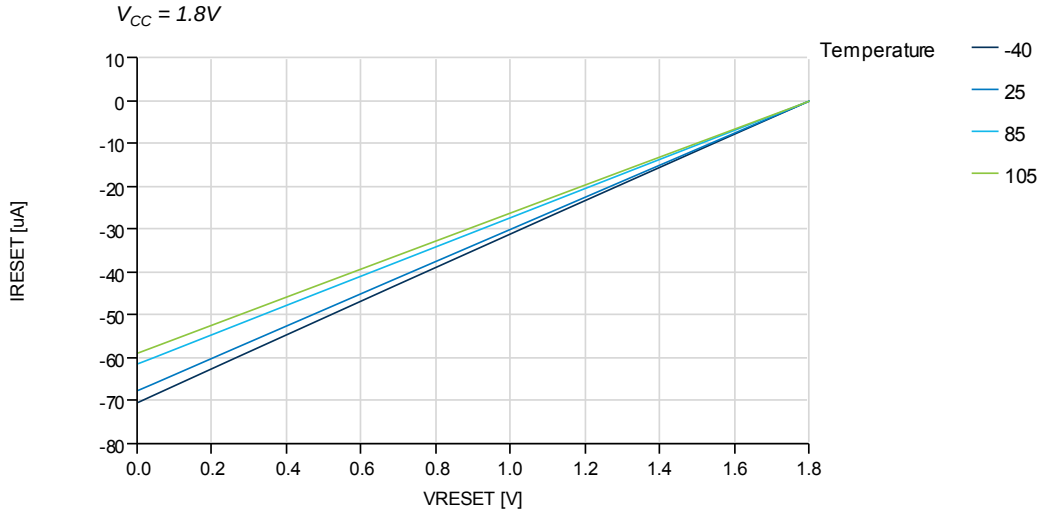


Figure 37-64.Reset pin pull-up resistor current vs. reset pin voltage.

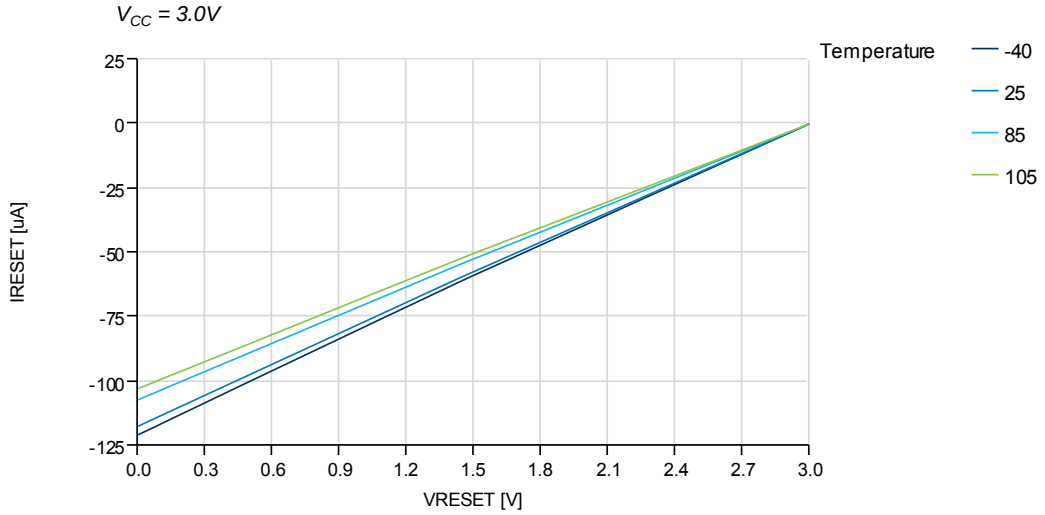


Figure 37-65.Reset pin pull-up resistor current vs. reset pin voltage.

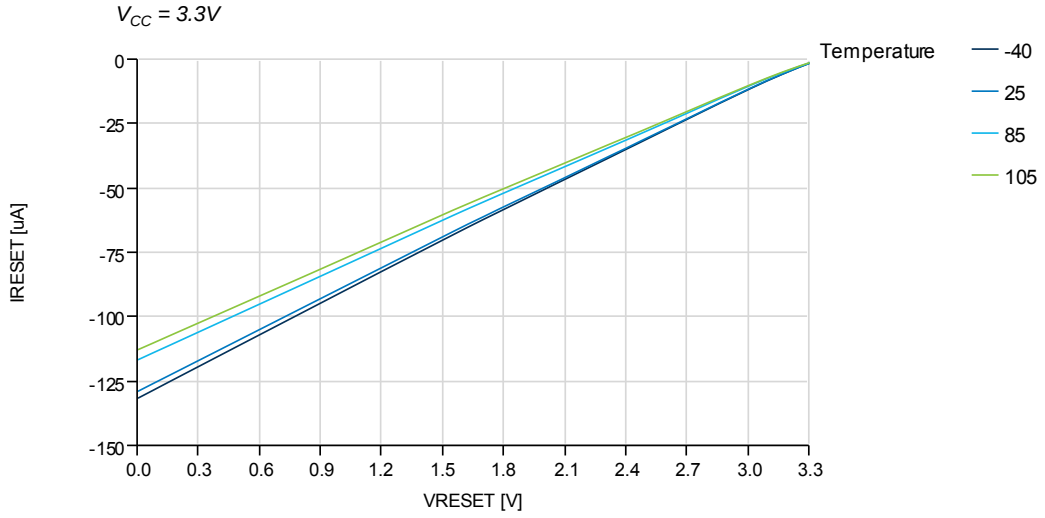


Figure 37-66.Reset pin input threshold voltage vs. V_{CC}.

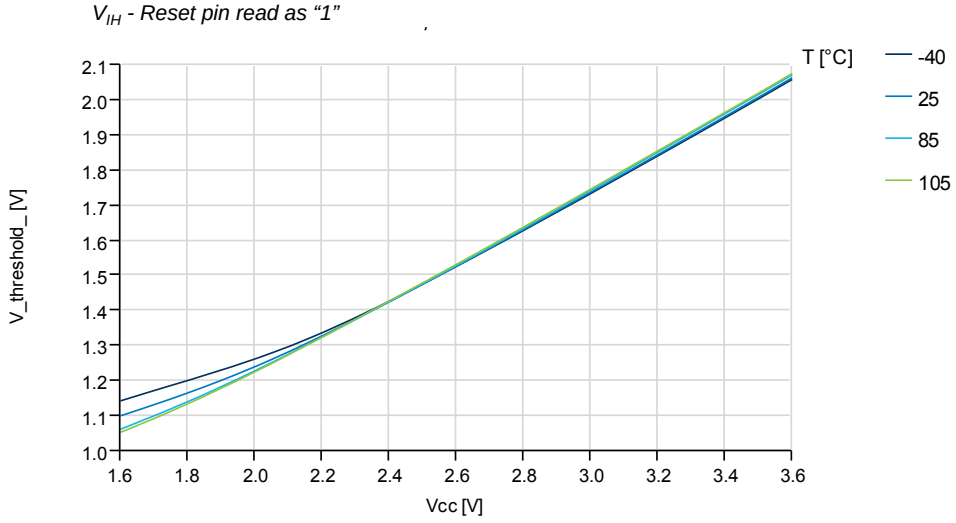
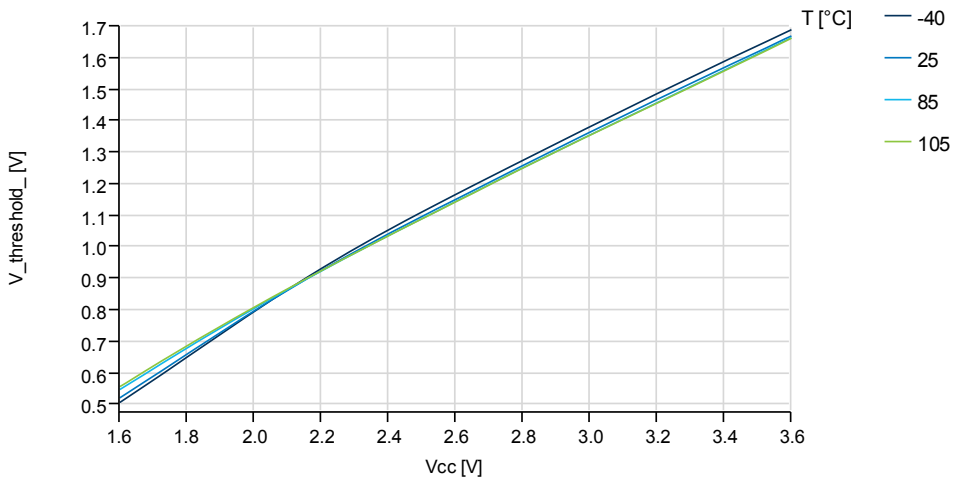


Figure 37-67.Reset pin input threshold voltage vs. V_{CC}.

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



37.9 Power-on Reset Characteristics

Figure 37-68.Power-on reset current consumption vs. V_{CC}.

BOD level = 3.0V, enabled in continuous mode

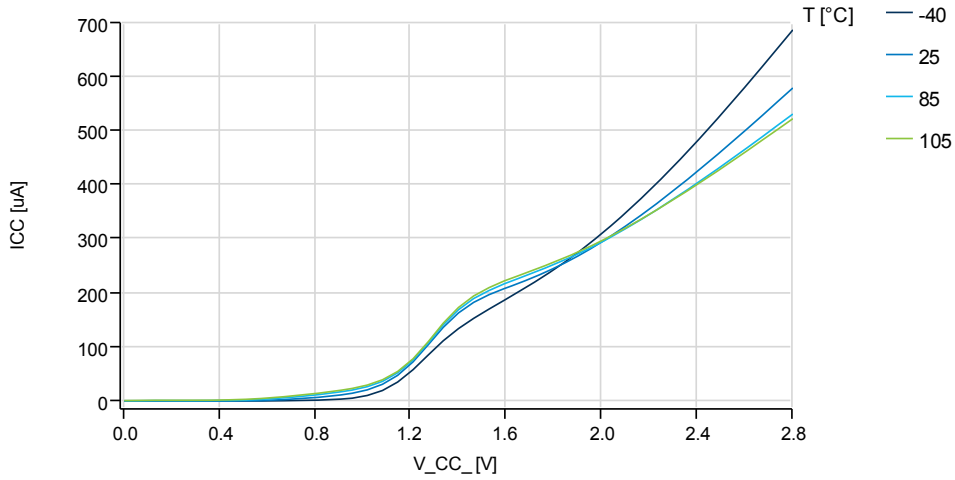
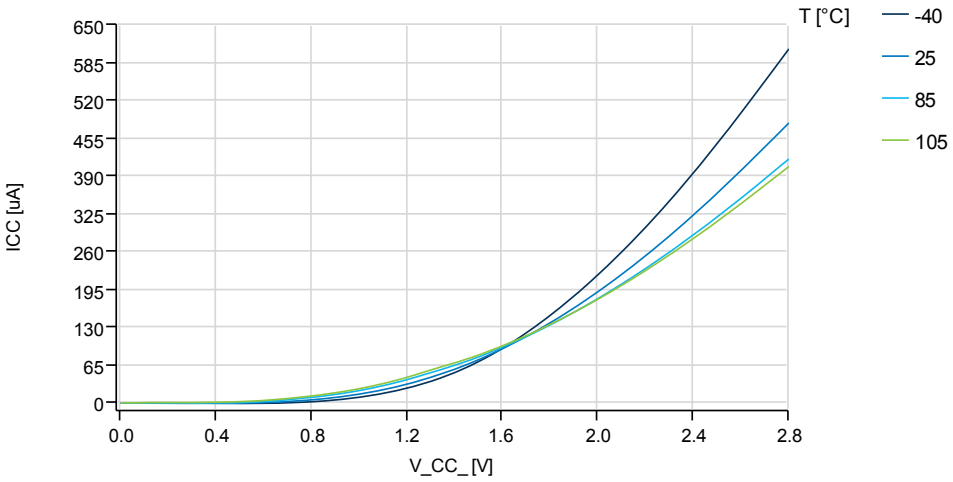


Figure 37-69.Power-on reset current consumption vs. V_{CC}.

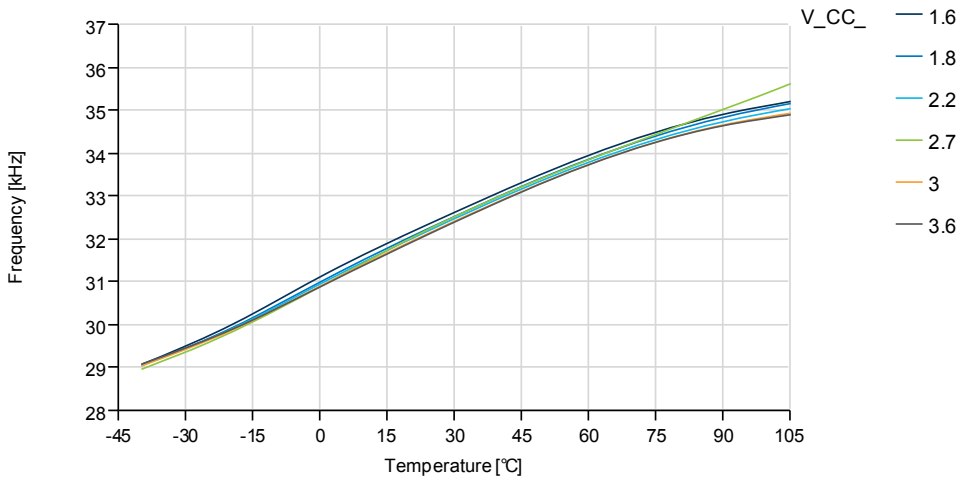
BOD level = 3.0V, enabled in sampled mode



37.10 Oscillator Characteristics

37.10.1 Ultra Low-Power Internal Oscillator

Figure 37-70. Ultra Low-Power internal oscillator frequency vs. temperature.



37.10.2 32.768KHz Internal Oscillator

Figure 37-71. 32.768kHz internal oscillator frequency vs. temperature.

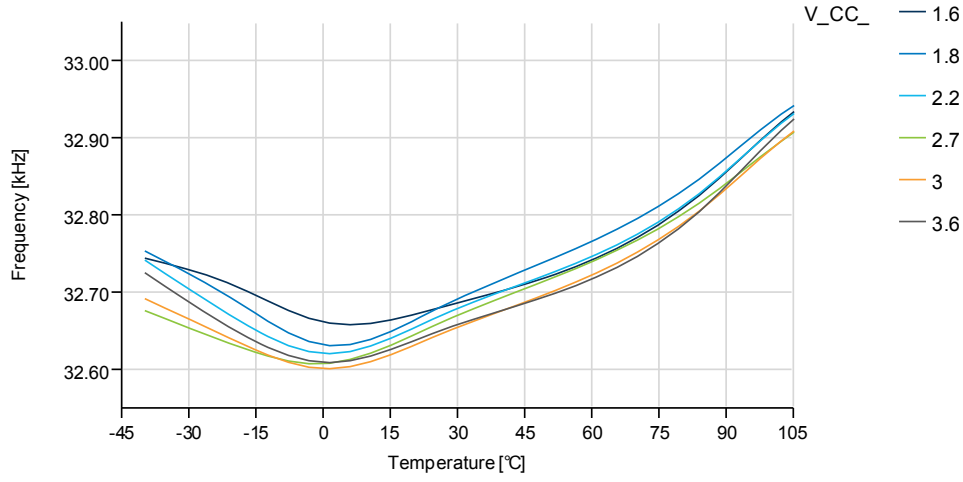


Figure 37-72. 32.768kHz internal oscillator frequency vs. calibration value.

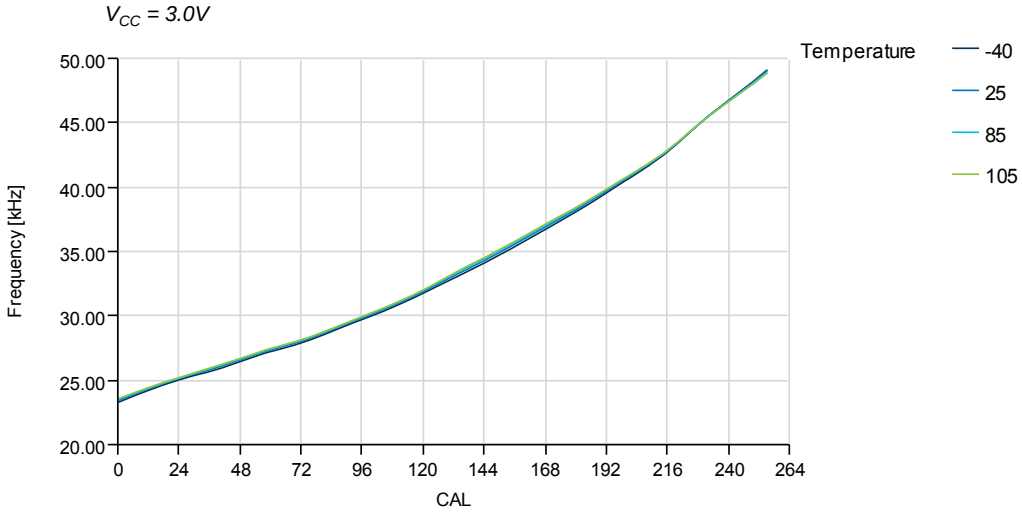
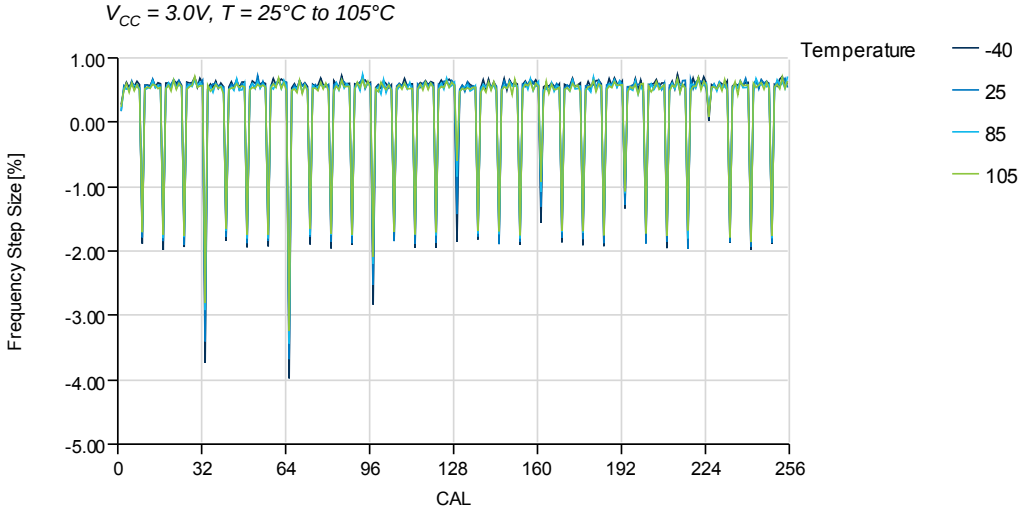


Figure 37-73. 32.768kHz internal oscillator calibration step size.



37.10.3 8MHz Internal Oscillator

Figure 37-74. 8MHz internal oscillator frequency vs. temperature.

Normal mode

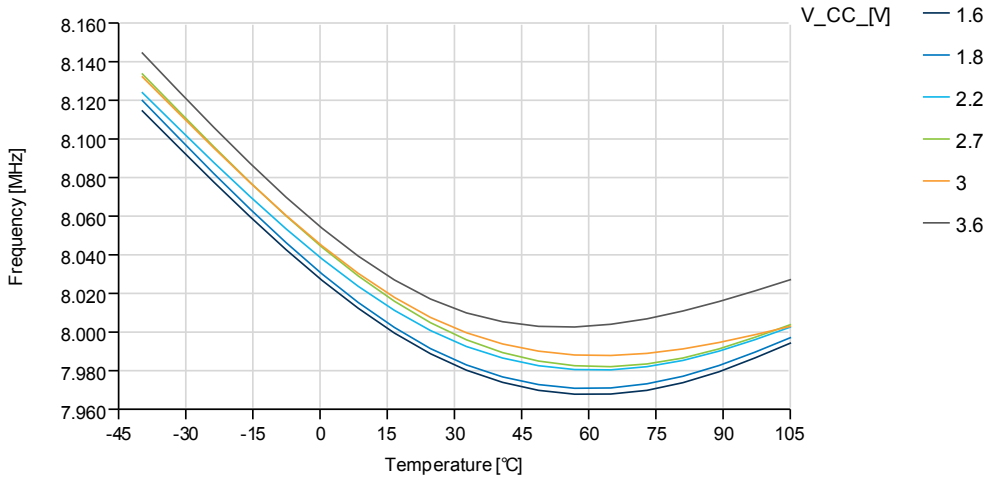


Figure 37-75. 8MHz internal oscillator frequency vs. temperature.

Low power mode

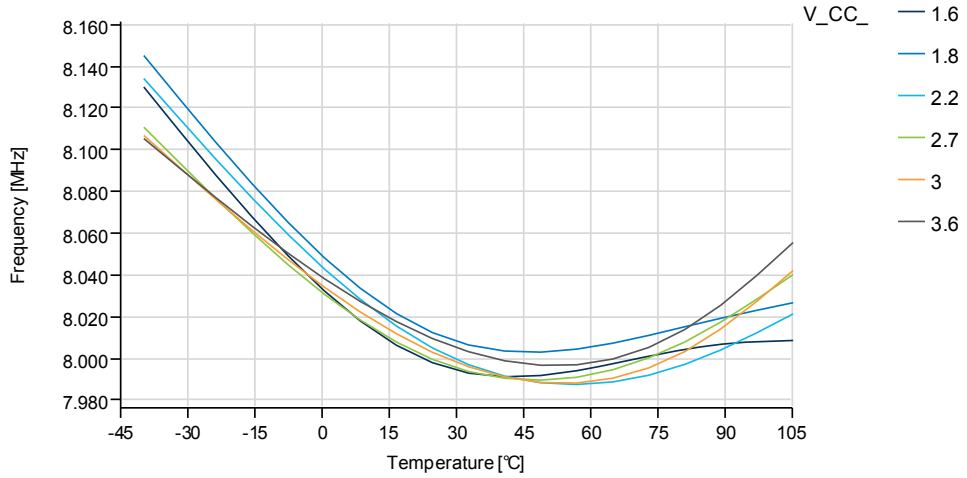


Figure 37-76. 8MHz internal oscillator CAL calibration step size.

V_{CC} = 3.0V

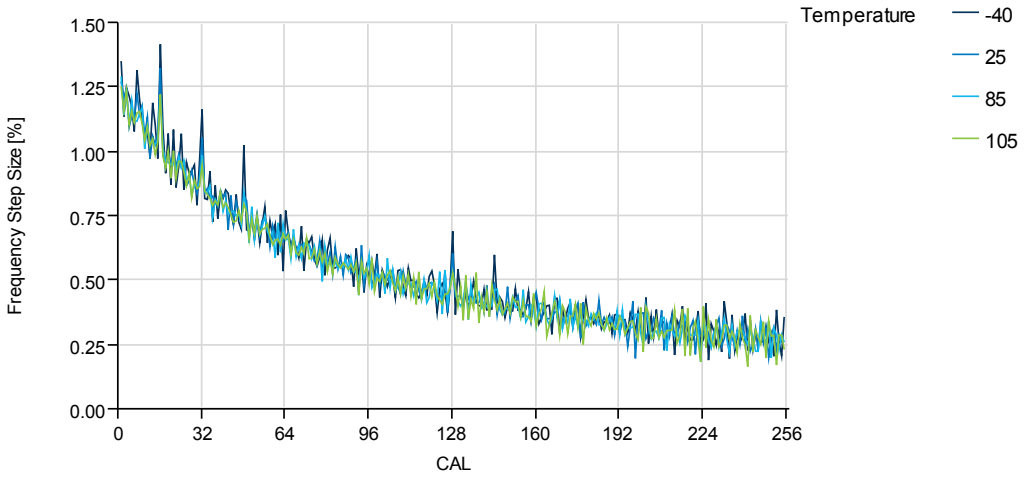
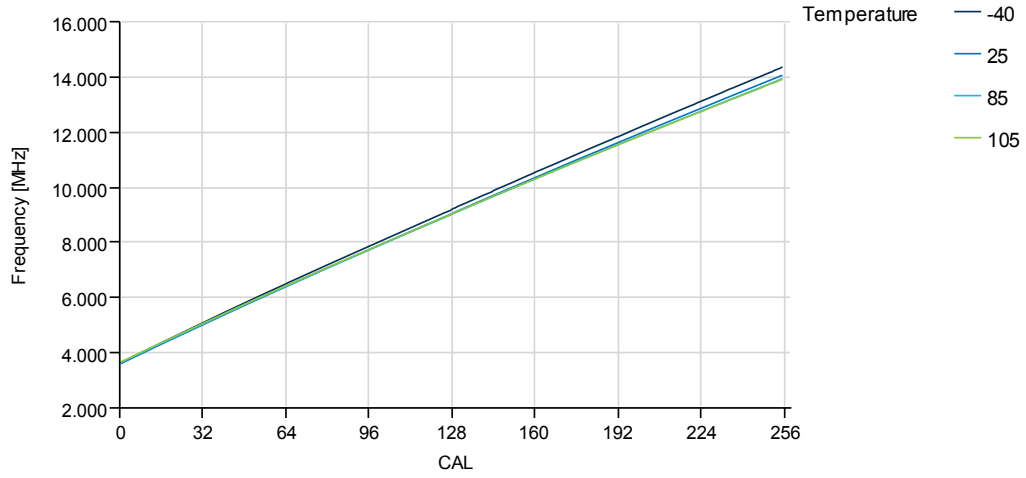


Figure 37-77. 8MHz internal oscillator frequency vs. calibration.

V_{CC} = 3.0V, Normal mode



37.10.4 32MHz Internal Oscillator

Figure 37-78. 32MHz internal oscillator frequency vs. temperature.

DFLL disabled

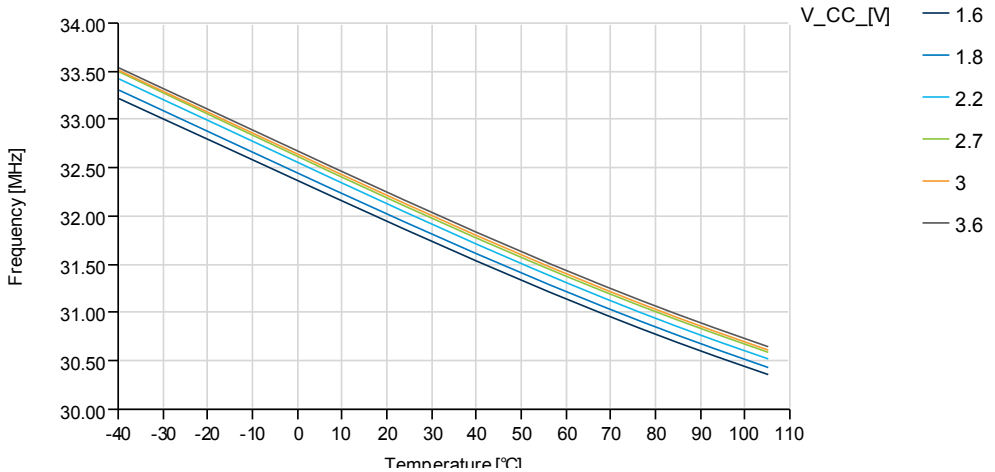


Figure 37-79. 32MHz internal oscillator frequency vs. temperature.

DFLL enabled, from the 32.768kHz internal oscillator

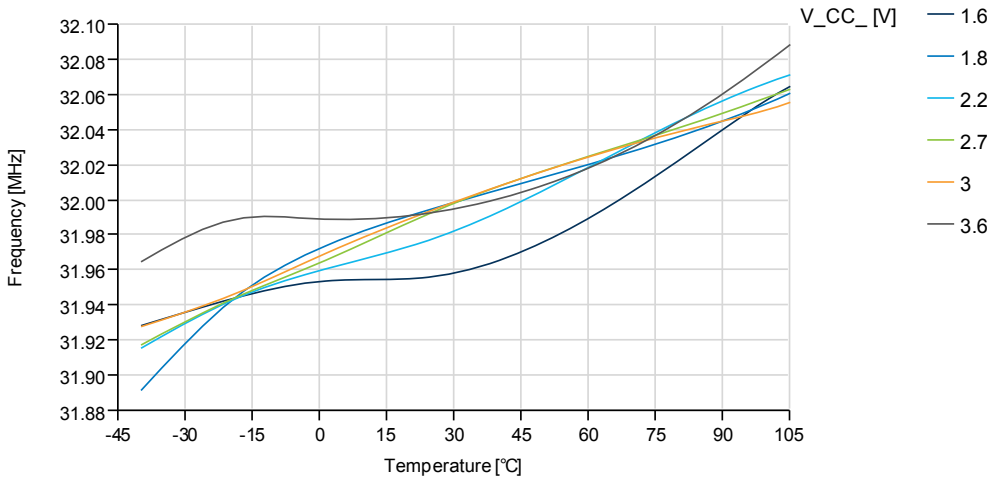


Figure 37-80. 32MHz internal oscillator CALA calibration step size.

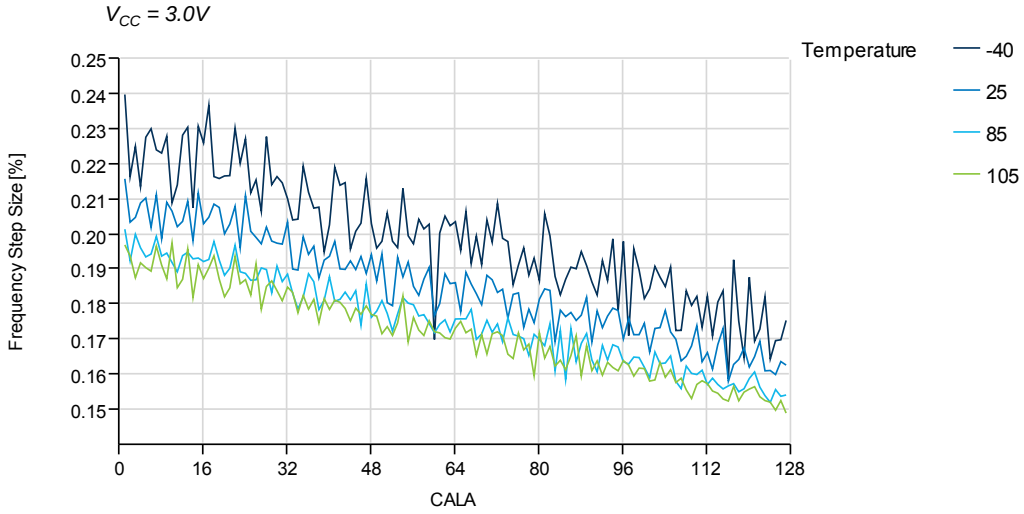


Figure 37-81. 32MHz internal oscillator frequency vs. CALA calibration value.

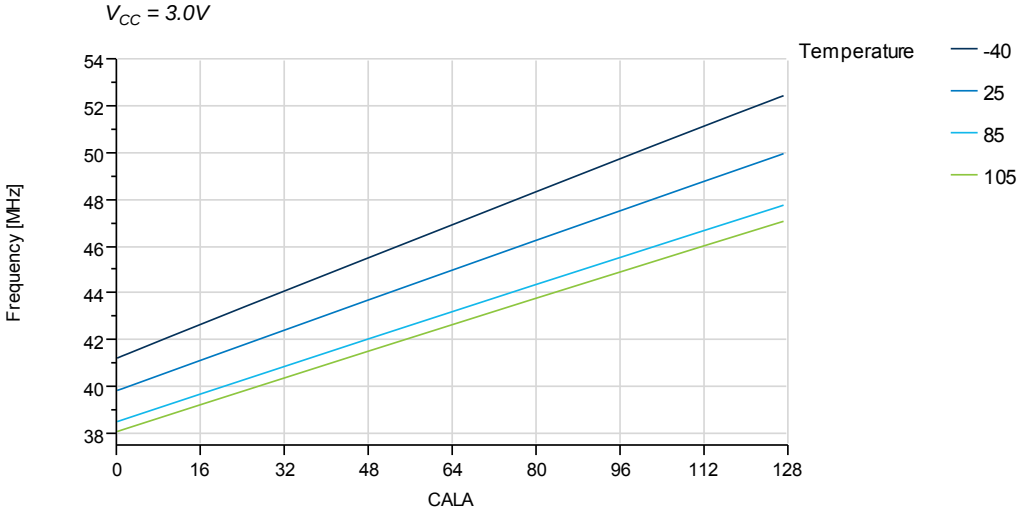
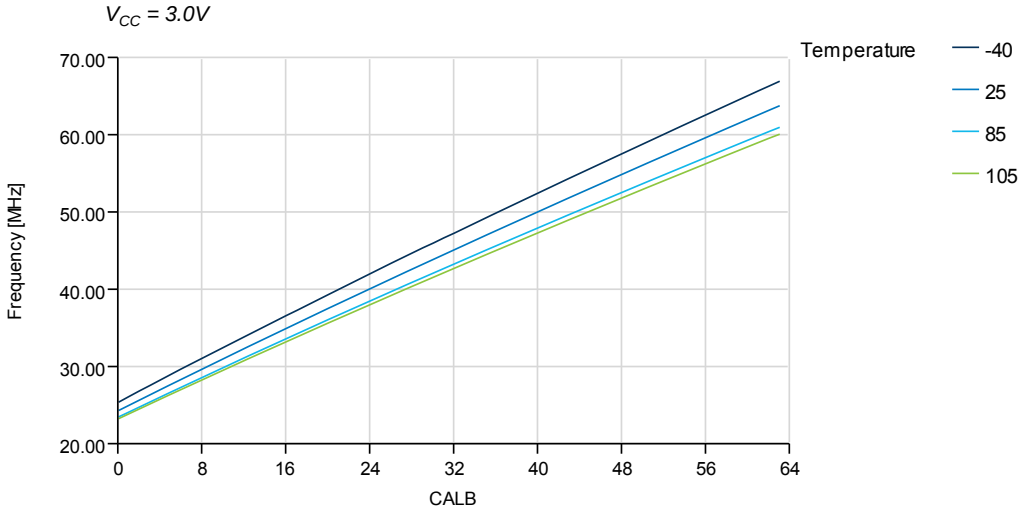


Figure 37-82. 32MHz internal oscillator frequency vs. CALB calibration value.



37.11 Two-wire Interface Characteristics

Figure 37-83. SDA fall time vs. temperature.

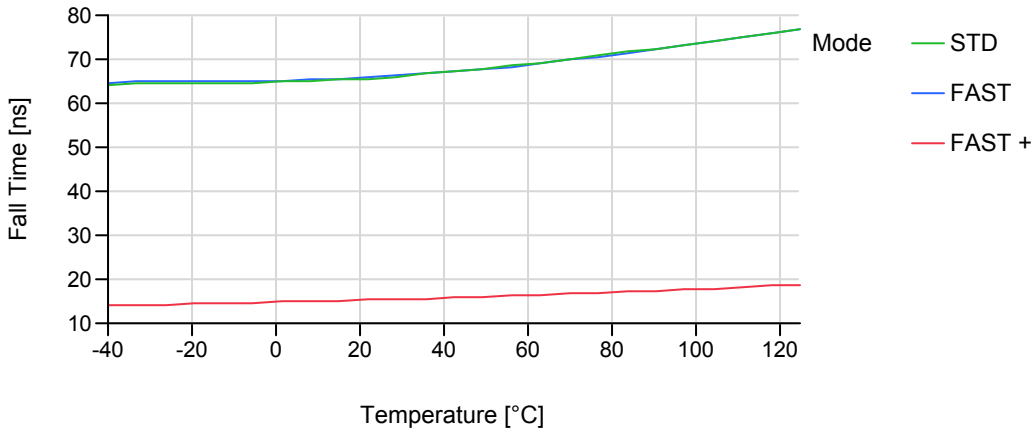
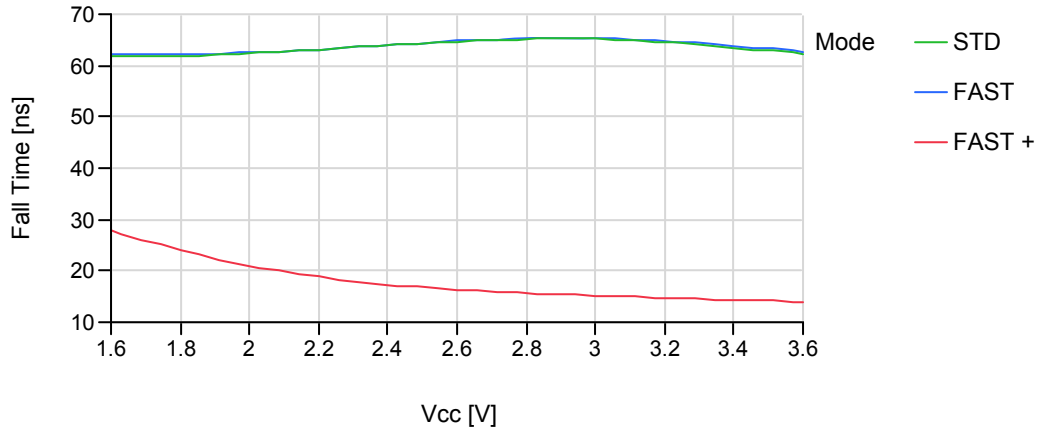
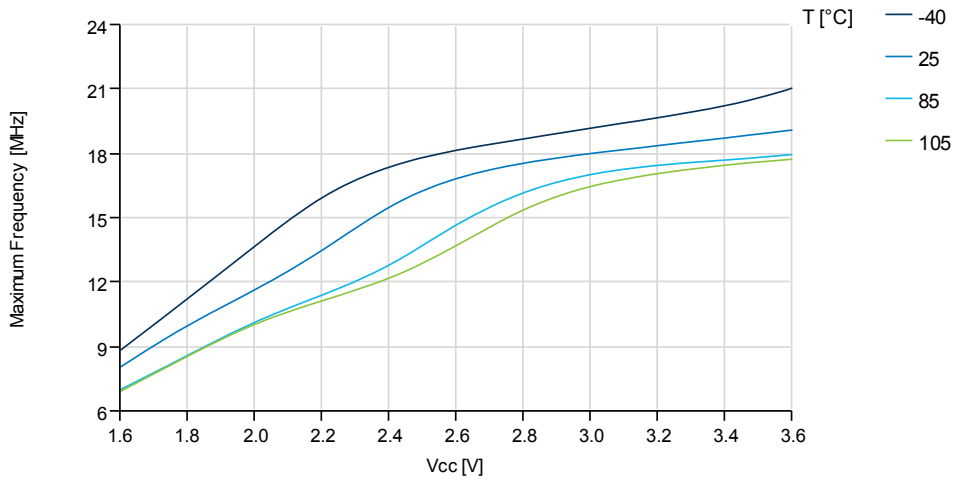


Figure 37-84. SDA fall time vs. V_{CC}.



37.12 PDI Characteristics

Figure 37-85. Maximum PDI frequency vs. V_{CC}.



38. Errata – ATxmega32E5 / ATxmega16E5 / ATxmega8E5

38.1 Rev. B

- DAC: AREF on PD0 is not available for the DAC
- ADC: Offset correction fails in unsigned mode
- EEPROM write and Flash write operations fails under 2.0V
- TWI Master or slave remembering data
- TWI SM bus level one Master or slave remembering data
- Temperature Sensor not calibrated

Issue: DAC: AREF on PD0 is not available for the DAC

The AREF external reference input on pin PD0 is not available for the DAC.

Workaround:

No workaround. Only AREF on pin PA0 can be used as external reference input for the DAC.

Issue: ADC: Offset correction fails in unsigned mode

In single ended, unsigned mode, a problem appears in low saturation (zero) when the offset correction is activated. The offset is removed from result and when a negative result appears, the result is not correct.

Workaround:

No workaround, but avoid using this correction method to cancel ΔV effect.

Issue: EEPROM write and Flash write operations fails under 2.0V

EEPROM write and Flash write operations are limited from 2.0V to 3.6V. Other functionalities operates from 1.6V to 3.6V.

Workaround:

None.

Issue: TWI master or slave remembering data

If a write is made to Data register, prior to Address register, the TWI design sends the data as soon as the write to Address register is made. But the send data will be always 0x00.

Workaround:

None.

Issue: **TWI SM bus level one Master or slave remembering data**

If a write is made to Data register, prior to Address register, the TWI design sends the data as soon as the write to Address register is made. But the send data will be always 0x00.

Workaround:

Since single interrupt line is shared by both timeout interrupt and other TWI interrupt sources, there is a possibility in software that data register will be written after timeout is detected but before timeout interrupt routine is executed. To avoid this, in software, before writing data register, always ensure that timeout status flag is not set.

Issue: **Temperature Sensor not calibrated**

Temperature sensor factory calibration is not implemented on devices before date code 1324.

Workaround:

None.

38.2 Rev. A

- DAC: AREF on PD0 is not available for the DAC
- EDMA: Channel transfer never stops when double buffering is enabled on sub-sequent channels
- ADC: Offset correction fails in unsigned mode
- ADC: Averaging is failing when channel scan is enabled
- ADC: Averaging in single conversion requires multiple conversion triggers
- ADC accumulator sign extends the result in unsigned mode averaging
- ADC: Free running average mode issue
- ADC: Event triggered conversion in averaging mode
- AC: Flag can not be cleared if the module is not enabled
- USART: Receiver not functional when variable data length and start frame detector are enabled
- T/C: Counter does not start when CLKSEL is written
- EEPROM write and Flash write operations fails under 2.0V
- TWI master or slave remembering data
- Temperature Sensor not calibrated

Issue: DAC: AREF on PD0 is not available for the DAC

The AREF external reference input on pin PD0 is not available for the DAC.

Workaround:

No workaround. Only AREF on pin PA0 can be used as external reference input for the DAC.

Issue: EDMA: Channel transfer never stops when double buffering is enabled on sub-sequent channels

When the double buffering is enabled on two channels, the channels which are not set in double buffering mode are never disabled at the end of the transfer. A new transfer can start if the channel is not disabled by software.

Workaround:

- CHMODE = 00

  Enable double buffering on all channels or do not use channels which are not set the double buffering mode.
- CHMODE = 01 or 10

  Do not use the channel which is not supporting the double buffering mode.

Issue: ADC: Offset correction fails in unsigned mode

In single ended, unsigned mode, a problem appears in low saturation (zero) when the offset correction is activated. The offset is removed from result and when a negative result appears, the result is not correct.

Workaround:

No workaround, but avoid using this correction method to cancel ΔV effect.

Issue: **ADC: Averaging is failing when channel scan is enabled**

For a correct operation, the averaging must complete on the on-going channel before incrementing the input offset. In the current implementation, the input offset is incremented after the ADC sampling is done.

Workaround:

None.

Issue: **ADC: Averaging in single conversion requires multiple conversion triggers**

For a normal operation, an unique start of conversion trigger starts a complete average operation. Then, for N-samples average operation, we should have:

- One start of conversion
- N conversions + average
- Optional interrupt when the Nth conversion/last average is completed

On silicon we need:

- N start of conversion

The two additional steps are well done.

Workaround:

- Set averaging configuration
- N starts of conversion by polling the reset of START bit
- Wait for interrupt flag (end of averaging)

Issue: **ADC accumulator sign extends the result in unsigned mode averaging**

In unsigned mode averaging, when the MSB is going high(1), measurements are considered as negative when right shift is used. This sets the unused most significant bits once the shift is done.

Workaround:

Mask to zero the unused most significant bits once shift is done.

Issue: **ADC: Free running average mode issue**

In free running mode the ADC stops the ongoing averaging as soon as free running bit is disabled. This creates the need to flush the ADC before starting the next conversion since one or two conversions might have taken place in the internal accumulator.

Workaround:

Disable and re-enable the ADC before the start of next conversion in free running average mode.

Issue: **ADC: Event triggered conversion in averaging mode**

If the ADC is configured as event triggered in averaging mode, then a single event does not complete the entire averaging as it should be.

Workaround:

In the current revision, N events are needed for completing averaging on N samples.

Issue: **AC: Flag can not be cleared if the module is not enabled**

It is not possible to clear the AC interrupt flags without enabling either of the analog comparators.

Workaround:

Clear the interrupt flags before disabling the module.

Issue: **USART: Receiver not functional when variable data length and start frame detector are enabled**

When using USART in variable frame length with XCL PEC01 configuration and start frame detection activated, the USART receiver is not functional.

Workaround:

Use XCL BTC0PCE2 configuration instead of PEC01.

Issue: **T/C: Counter does not start when CLKSEL is written**

When STOP bit is cleared (CTRLGCLR.STOP) before the timer/counter is enabled (CTRLA.CLKSEL != OFF), the T/C doesn't start operation.

Workaround:

Do not write CTRLGCLR.STOP bit before writing CTRLA.CLKSEL bits.

Issue: **EEPROM write and Flash write operations fails under 2.0V**

EEPROM write and Flash write operations are limited from 2.0V to 3.6V. Other functionalities operates from 1.6V to 3.6V.

Workaround:

None.

Issue: **TWI master or Slave remembering data**

If a write is made to Data register, prior to Address register, the TWI design sends the data as soon as the write to Address register is made. But the send data will be always 0x00.

Workaround:

None.

Issue: **Temperature Sensor not calibrated**

Temperature sensor factory calibration is not implemented.

Workaround:

None.

39. Revision History

Please note that referring page numbers in this section are referred to this document. The referring revision in this document section are referring to the document revision.

39.1 8135F – 08/2013

1.	TWI characteristics: Units of Data setup time ($t_{SU;DAT}$) changed from µs to ns in Table 36-30 on page 91.

39.2 8135E – 06/2013

1.	Errata “Rev. B” : Updated date code from 1318 to 1324 in “Temperature Sensor not calibrated” on page 136.

39.3 8135D – 06/2013

1.	Analog Comparator Characteristics: Updated minimum and maximum values of Input Voltage Range, Table 36-14 on page 81.

39.4 8135C – 05/2013

1.	Electrical Characteristics, Table 36-3 on page 73: Updated typical value from 7mA to 6mA for Active Current Consumption, 32MHz, V_{CC}=3.0V.
2.	Errata “Rev. A” and “Rev. B” : Added DAC errata: AREF on PORT C0.

39.5 8153B – 04/2013

1.	“Rev. B” on page 135: Removed the “EDMA: Channel transfer never stops when double buffering is enabled on sub-sequent channels” errata.

39.6 8153A – 04/2013

1.	Initial revision.

Table of Contents

Atmel | Enabling Unlimited Possibilities®

Atmel Corporation
1600 Technology Drive
San Jose, CA 95110
USA
Tel: (+1) (408) 441-0311
Fax: (+1) (408) 487-2600
www.atmel.com

Atmel Asia Limited
Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Roa
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

Atmel Munich GmbH
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan G.K.
16F Shin-Osaki Kangyo Bldg
1-6-4 Osaki, Shinagawa-ku
Tokyo 141-0032
JAPAN
Tel: (+81) (3) 6417-0300
Fax: (+81) (3) 6417-0370