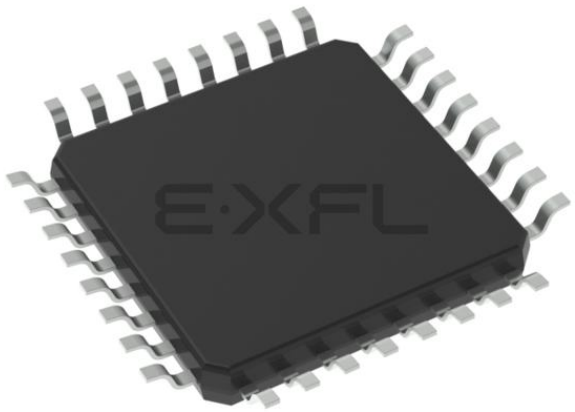


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	AVR
Core Size	8/16-Bit
Speed	32MHz
Connectivity	I²C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	26
Program Memory Size	8KB (4K x 16)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.6V ~ 3.6V
Data Converters	A/D 16x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	32-TQFP
Supplier Device Package	32-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atxmega8e5-an

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.

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.

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
0x0018	TCC4_INT_base	Timer/counter 4 on port C interrupt base

18. WeX – Waveform Extension

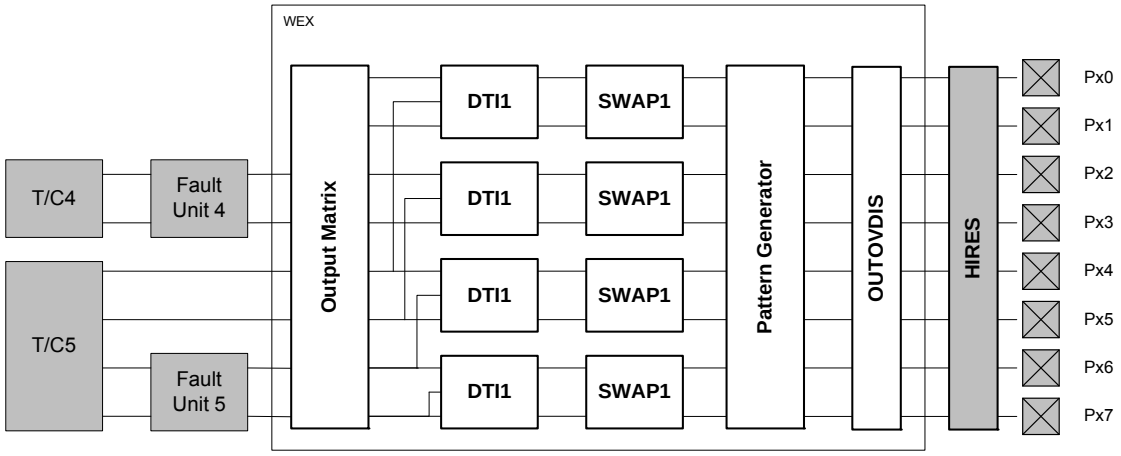
18.1 Features

- 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.

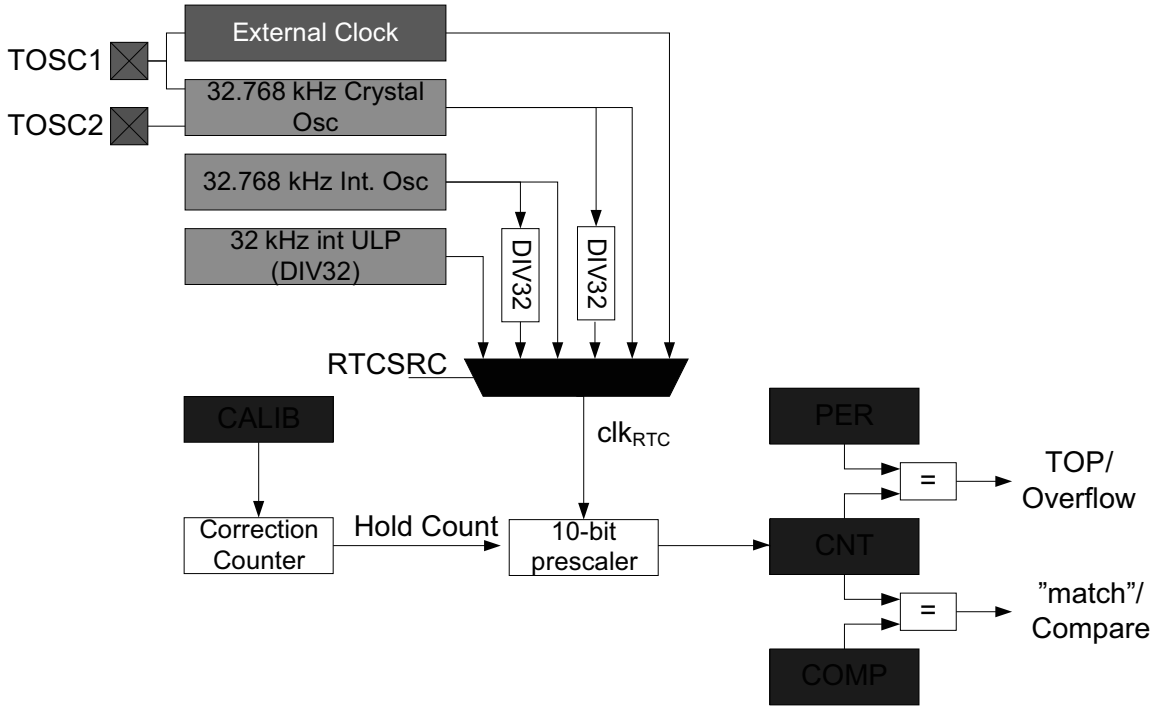
Figure 18-1. Waveform Extension and Closely Related Peripherals



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.

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 ±0.5ppm. 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.

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.

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.

29. DAC – Digital to Analog Converter

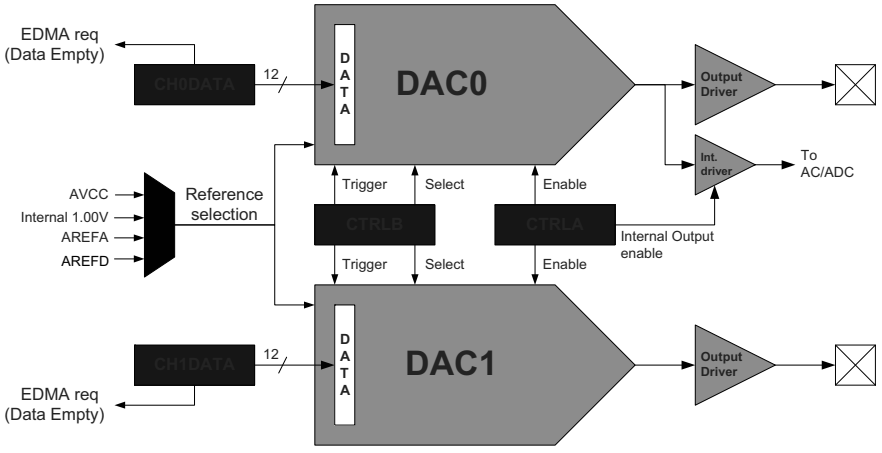
29.1 Features

- 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.

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.

XCKn	Transfer Clock for USART n
RXDn	Receiver Data for USART n
TXDn	Transmitter Data for USART n
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

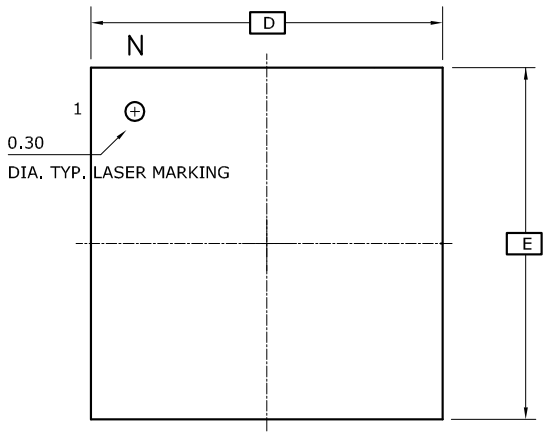
RESET	Reset pin
PDI_CLK	Program and Debug Interface Clock pin
PDI_DATA	Program and Debug Interface Data pin

34. Instruction Set Summary

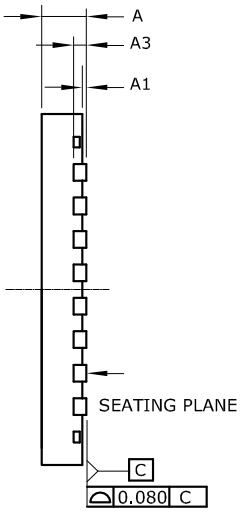
Mnemonics	Operands	Description	Operation	Flags	#Clocks
Arithmetic and Logic Instructions					
ADD	Rd, Rr	Add without Carry	Rd ← Rd + Rr	Z,C,N,V,S,H	1
ADC	Rd, Rr	Add with Carry	Rd ← Rd + Rr + C	Z,C,N,V,S,H	1
ADIW	Rd, K	Add Immediate to Word	Rd ← Rd + 1:Rd + K	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract without Carry	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)

35.2 32Z

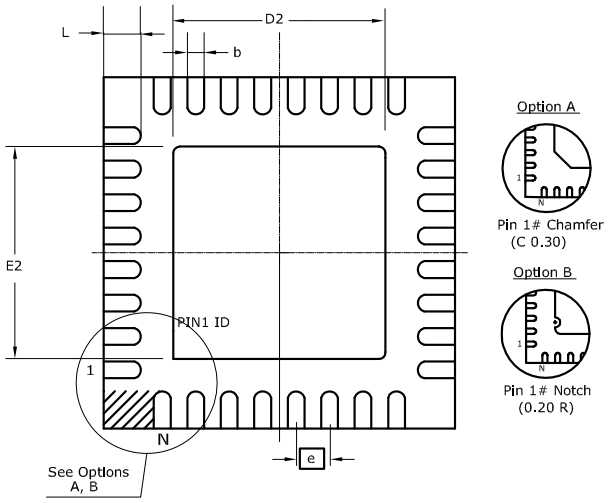
DRAWINGS NOT SCALED



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

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

Symbol	Parameter	Condition[2]		Min.	Typ.	Max.	Units
DNL[1]	Differential non-linearity	Differential mode	16ksps, V_{REF} = 3V		1		lsb
			16ksps, V_{REF} = 1V		2		
			300ksps, V_{REF} = 3V		1		
			300ksps, V_{REF} = 1V		2		
		Single ended unsigned mode	16ksps, V_{REF} = 3.0V		1	1.5	
			16ksps, V_{REF} = 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
			AV_{CC}/1.6		-5		
			AV_{CC}/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
			AV_{CC}/1.6		-8		
			AV_{CC}/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 V_{REF} 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

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		
	Page write		4		
	Atomic page erase and write		8		
EEPROM	Page erase		4		
	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.

36.15 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

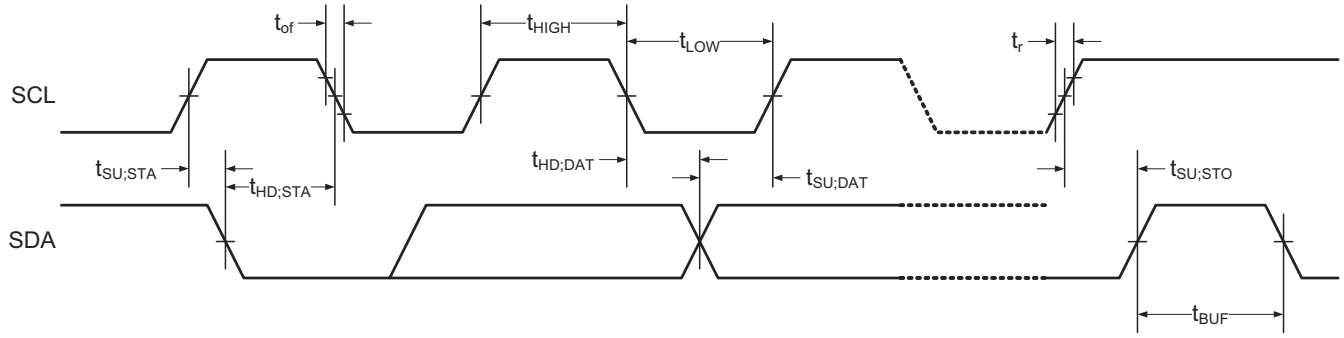


Table 36-30. Two-wire Interface Characteristics

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_{hys}	Hysteresis of Schmitt trigger inputs			$0.05V_{CC}$(1)			
V_{OL}	Output low voltage	3mA, sink current		0		0.4	
I_{OL}	Low level output current	f_{SCL} ≤ 400kHz	V_{OL} = 0.4V	3			mA
		f_{SCL} ≤ 1MHz		20			
t_r	Rise time for both SDA and SCL	f_{SCL} ≤ 400kHz		$20+0.1C_b$(1)(2)		300	ns
		f_{SCL} ≤ 1MHz				120	
t_{of}	Output fall time from V_{IHmin} to V_{ILmax}	10pF< C_b<400pF(2)	f_{SCL} ≤ 400kHz	$20+0.1C_b$(1)(2)		250	
			f_{SCL} ≤ 1MHz			120	
t_{SP}	Spikes suppressed by Input filter			0		50	
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} ≤ 100kHz		(V_{CC}-0.4V)/I_{OL}		100ns/C_b	Ω
		f_{SCL} ≤ 400kHz				300ns/C_b	
		f_{SCL} ≤ 1MHz				550ns/C_b	
$t_{HD;STA}$	Hold time (repeated) START condition	f_{SCL} ≤ 100kHz		4			µs
		f_{SCL} ≤ 400kHz		0.6			
		f_{SCL} ≤ 1MHz		0.26			

Figure 37-11.Idle Mode Supply Current vs. V_{CC}

f_{SYS} = 32.768kHz internal oscillator

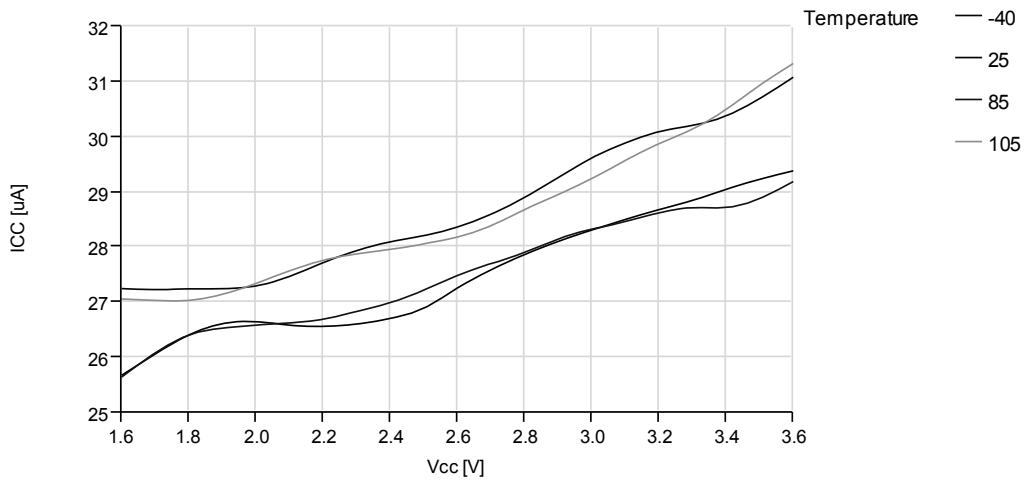


Figure 37-12.Idle Mode Supply Current vs. V_{CC}

f_{SYS} = 1MHz external clock

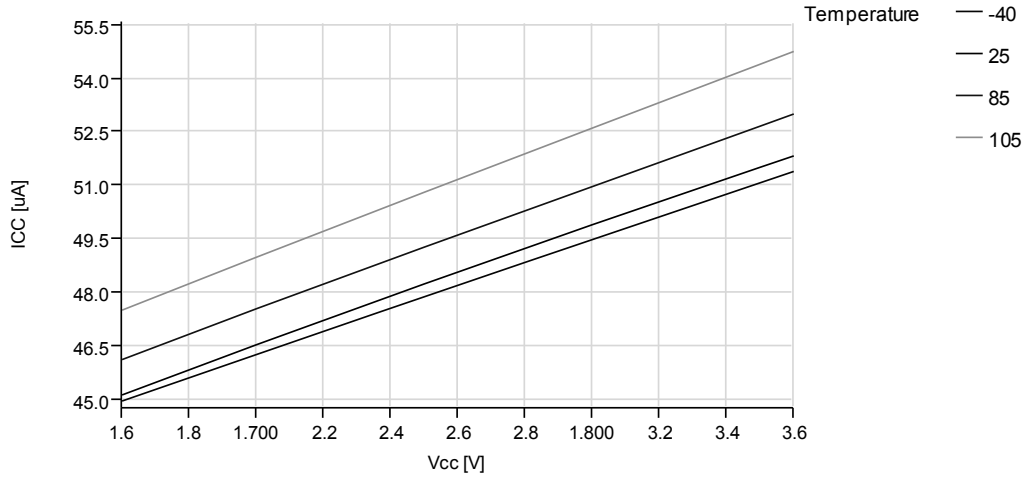


Figure 37-31.I/O Pin Output Voltage vs. Sink Current

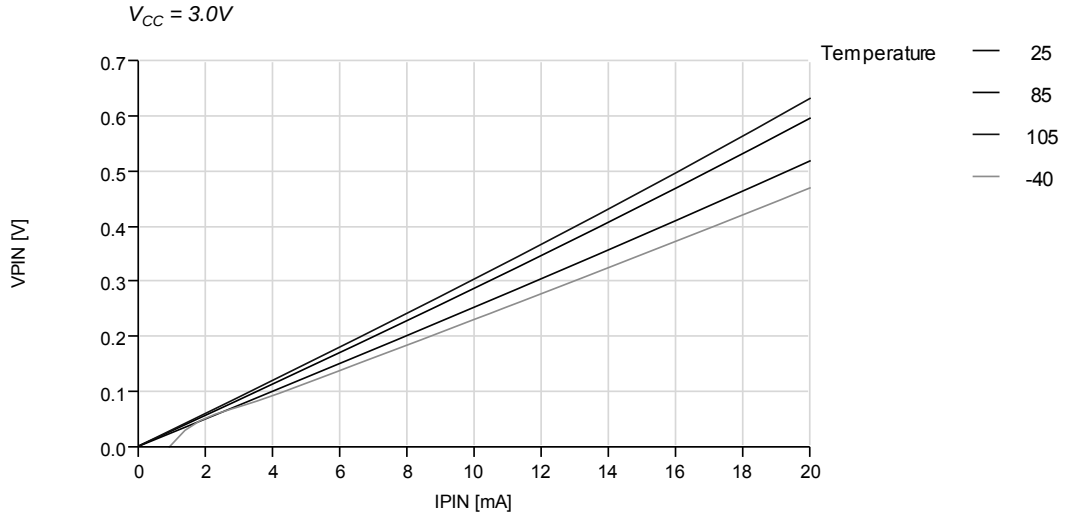


Figure 37-32.I/O Pin Output Voltage vs. Sink Current

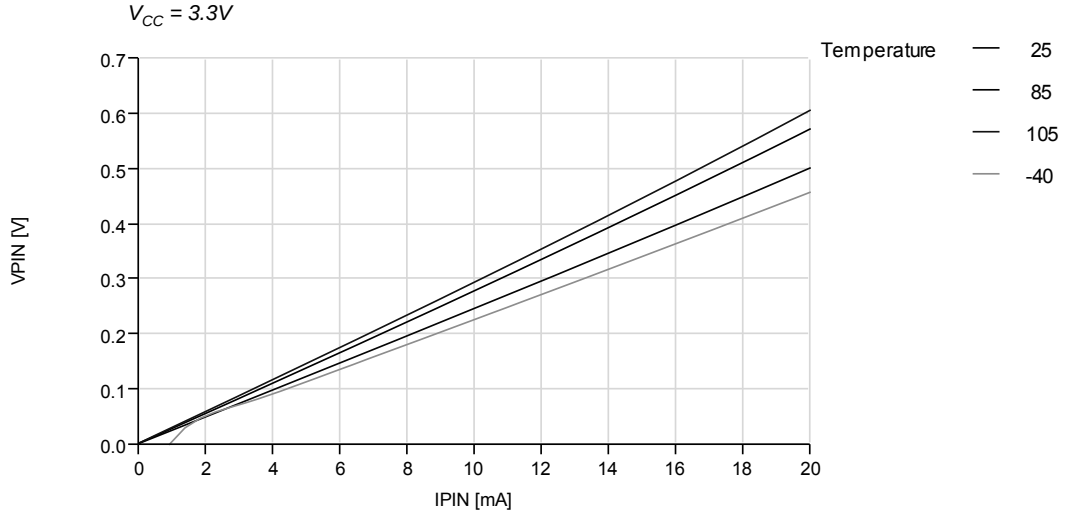
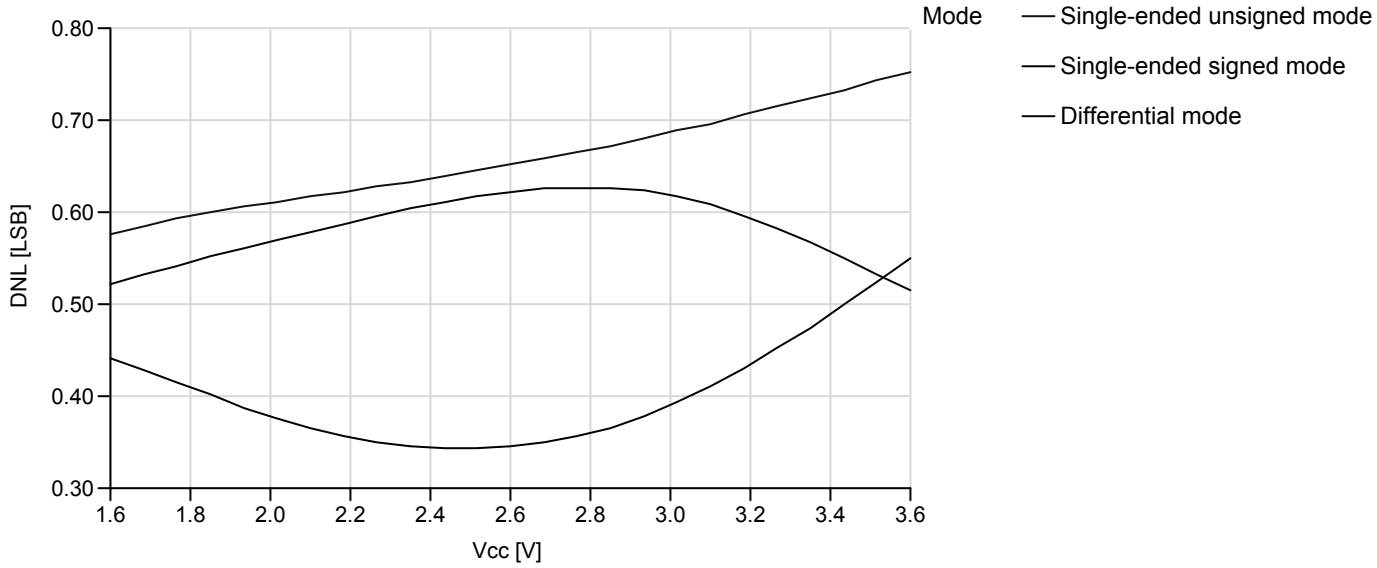


Figure 37-49.DNL Error vs. V_{CC}

T = 25°C, V_{REF} = 1.0V



37.5 AC Characteristics

Figure 37-50.Analog Comparator Hysteresis vs. V_{CC}

Small hysteresis

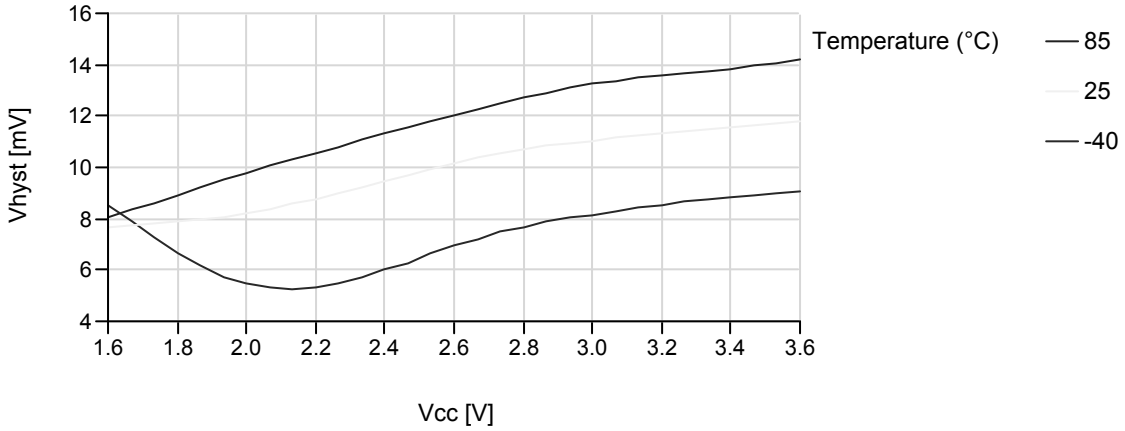
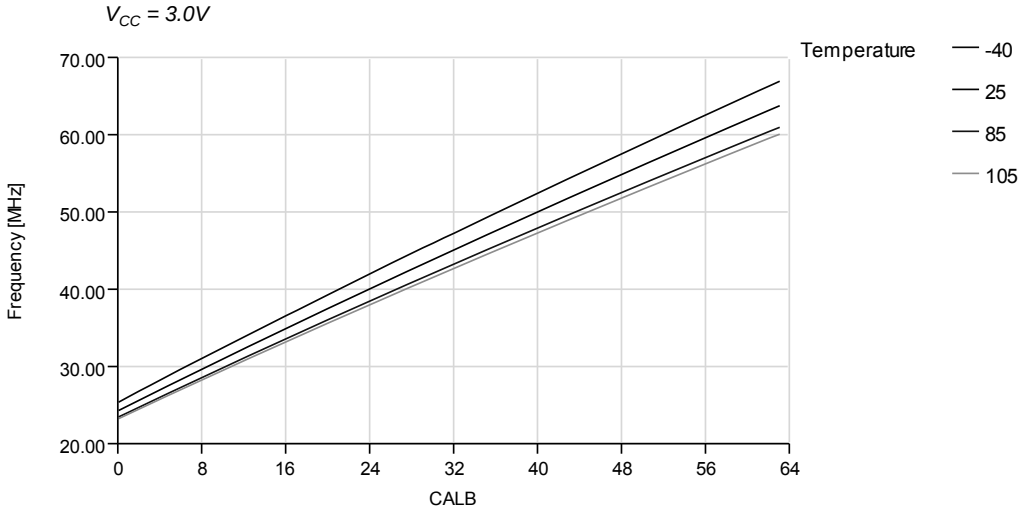
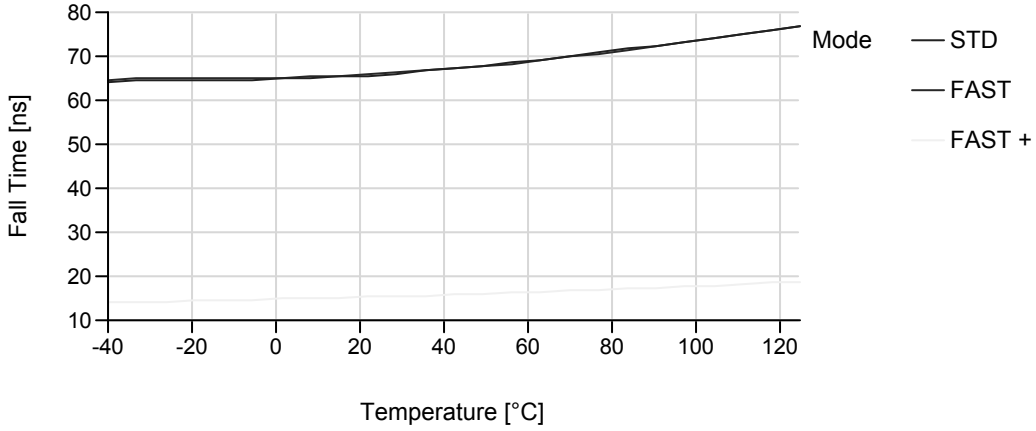


Figure 37-82. 32MHz internal Oscillator Frequency vs. CALB Calibration Value



37.11 Two-wire Interface Characteristics

Figure 37-83. SDA Fall Time vs. Temperature



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
- Automatic port override on PORT C
- Sext timer is not implemented in slave mode

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.