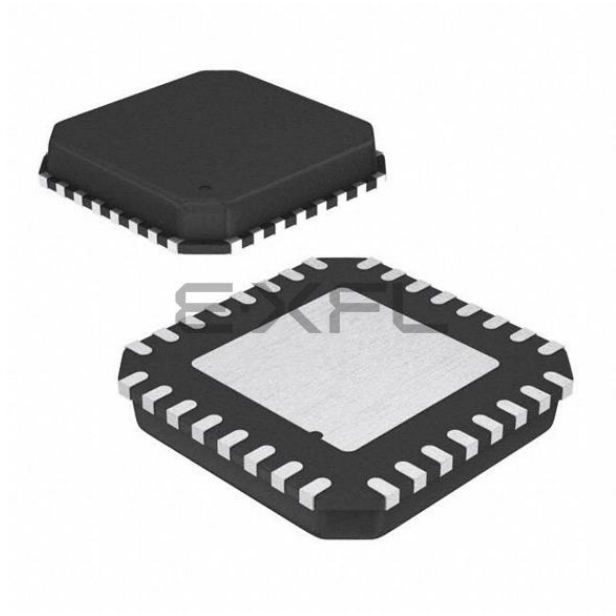


Welcome to E-XFL.COM

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	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	2K 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-VFQFN Exposed Pad
Supplier Device Package	32-VQFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atxmega16e5-mnr

1. Ordering Information

Ordering Code	Package[1][2][3]	Flash [Bytes]	EEPROM [Bytes]	SRAM [Bytes]	Speed [MHz]	Power supply [V]	Temp. [°C]
ATxmega8E5-AU	32A (7x7mm TQFP)	8K + 2K	512	1K	32	1.6 – 3.6	-40 – 85
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	512	2K	32	1.6 – 3.6	-40 – 85
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 – 3.6	-40 – 85
ATxmega32E5AUR[4]							
ATxmega32E5-MU	32Z (5x5mm VQFN)						
ATxmega32E5-MUR[4]							
ATxmega32E5-M4U	32MA (4x4mm UQFN)						
ATxmega32E5-M4UR[4]							
ATxmega8E5-AN	32A (7x7mm TQFP)	8K + 2K	512	1K	32	1.6 – 3.6	-40 – 105
ATxmega8E5-ANR[4]							
ATxmega8E5-MN	32Z (5x5mm VQFN)						
ATxmega8E5-MNR[4]							
ATxmega8E5-M4UN	32MA (4x4mm UQFN)						
ATxmega8E5-M4UNR[4]							
ATxmega16E5-AN	32A (7x7mm TQFP)	16K + 4K	512	2K	32	1.6 – 3.6	-40 – 105
ATxmega16E5-ANR[4]							
ATxmega16E5-MN	32Z (5x5mm VQFN)						
ATxmega16E5-MNR[4]							
ATxmega16E5-M4UN	32MA (4x4mm UQFN)						
ATxmega16E5-M4UNR[4]							

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

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.

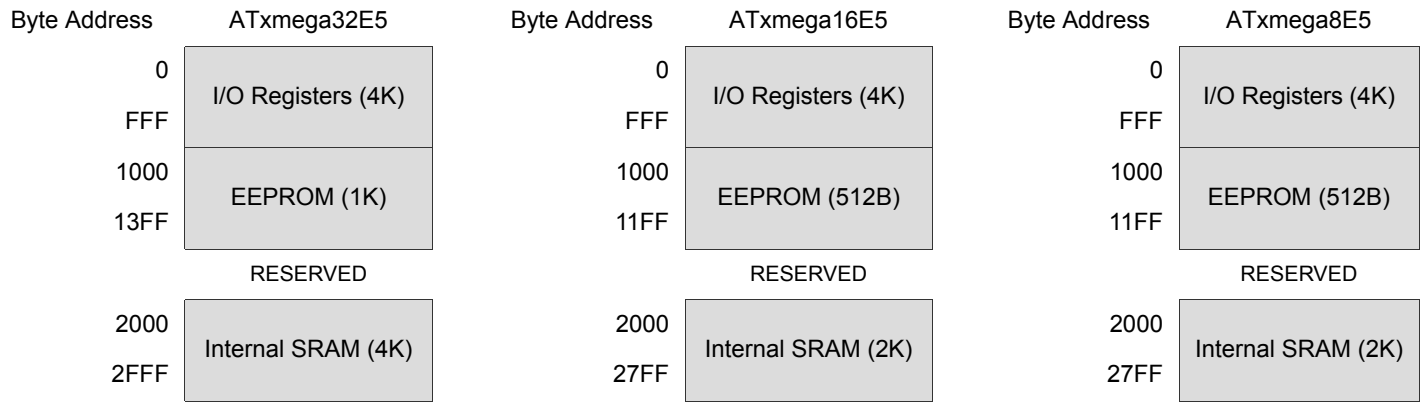
An un-programmed fuse or lock bit will have the value one, while a programmed fuse or lock bit will have the value zero.

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 15. 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 four 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.

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.

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 two and four 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 on page 21 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 23.

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.

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.

Program address (base address)	Source	Interrupt description
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

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.

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.

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
  - AV_{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.

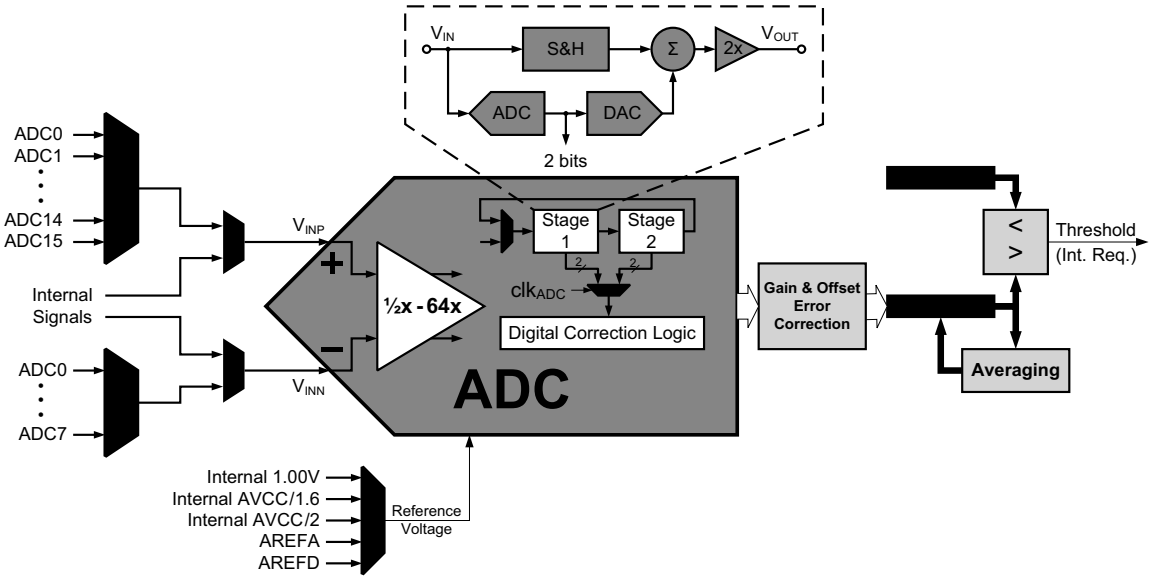
The ADC measurements can either be started by application software or an incoming event from another peripheral in the device. The ADC measurements can be started with predictable timing, and without software intervention. It is possible to use EDMA to move ADC results directly to memory or peripherals 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, $AV_{CC}/10$, 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.

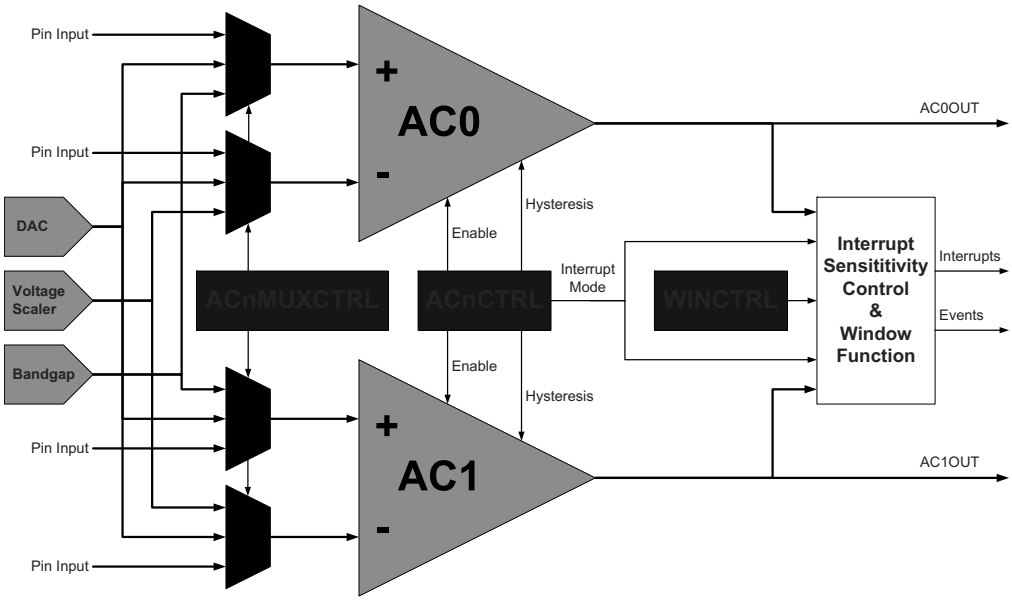
Figure 28-1. ADC Overview



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.

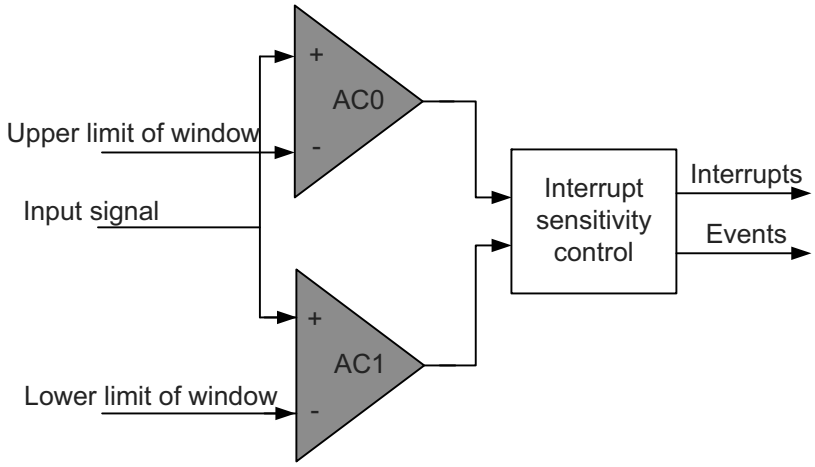
PORTA has one ADC. Notation of this peripheral is ADCA.

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.

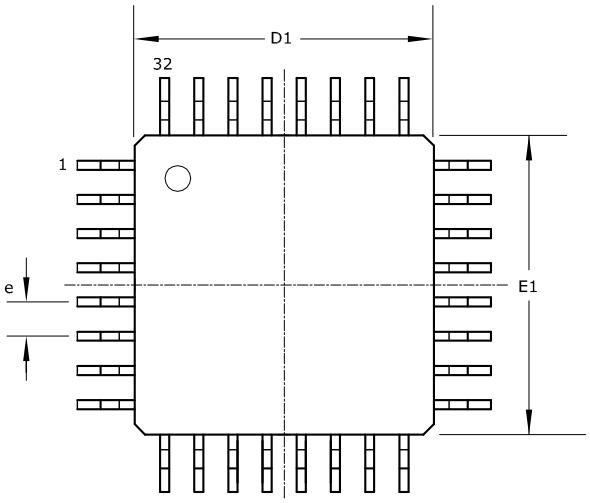
Figure 30-2. Analog Comparator Window Function



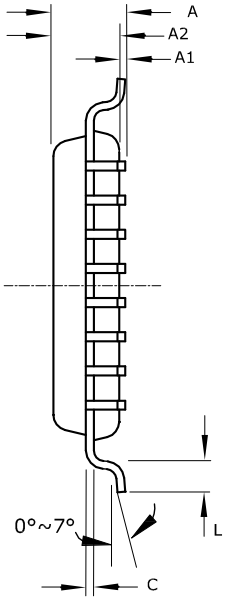
35. Packaging Information

35.1 32A

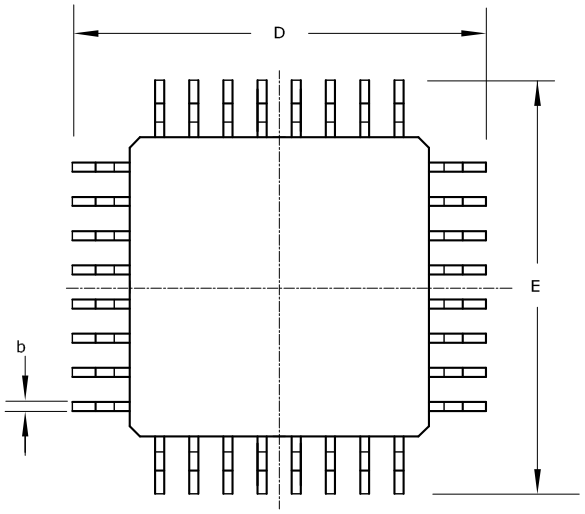
DRAWINGS NOT SCALED



TOP VIEW



SIDE VIEW



BOTTOM VIEW

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

37.4 DAC Characteristics

Figure 37-47.DAC INL Error vs. External V_{REF}

T = 25°C, V_{CC} = 3.6V

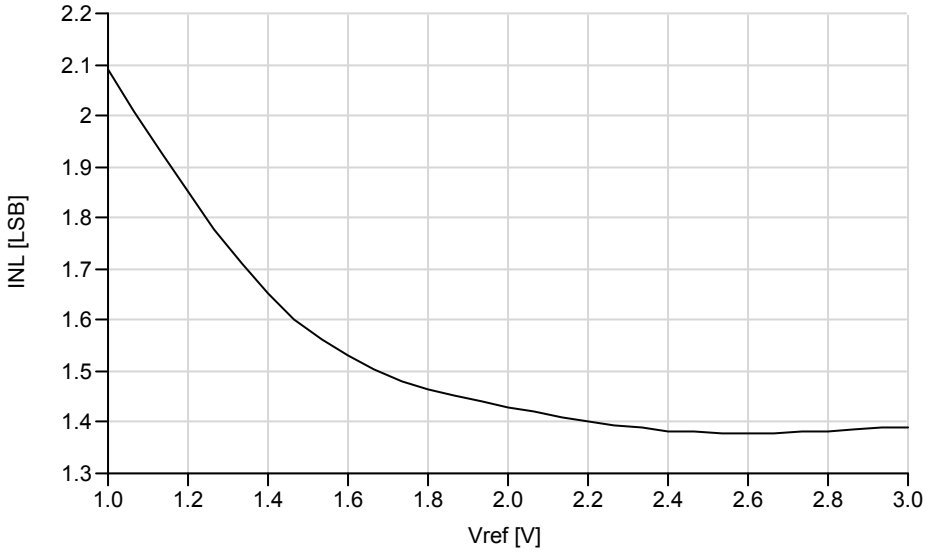


Figure 37-48.DNL Error vs. V_{REF}

T = 25°C, V_{CC} = 3.6V

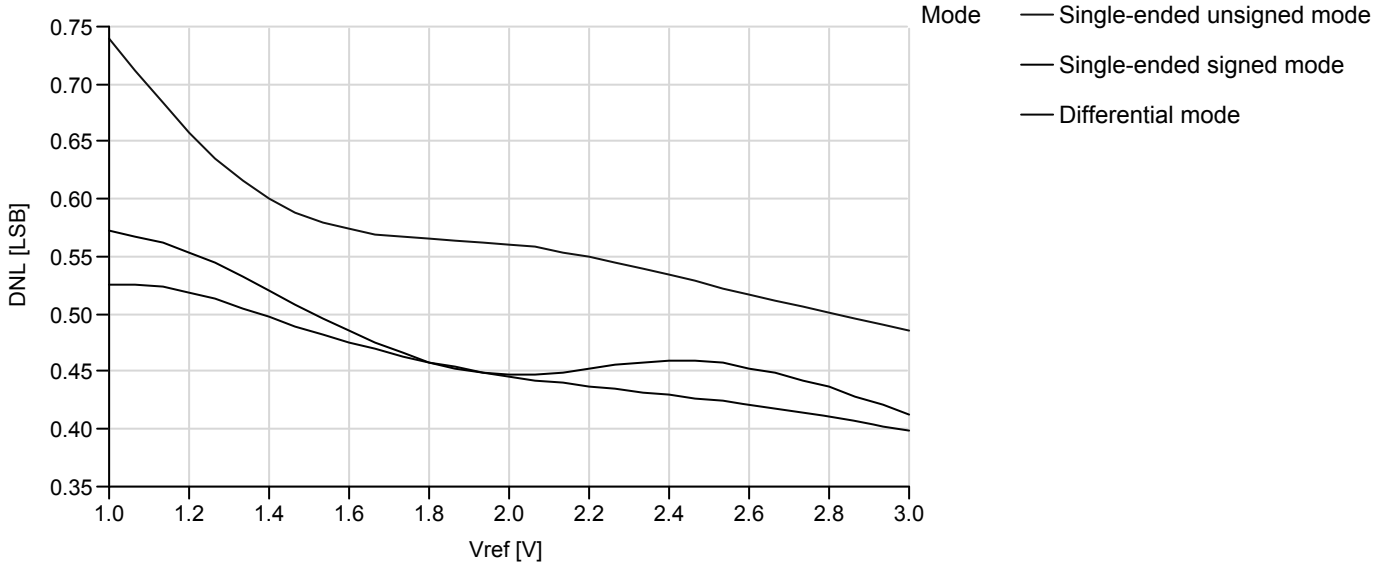


Figure 37-72. 32.768kHz Internal Oscillator Frequency vs. Calibration Value

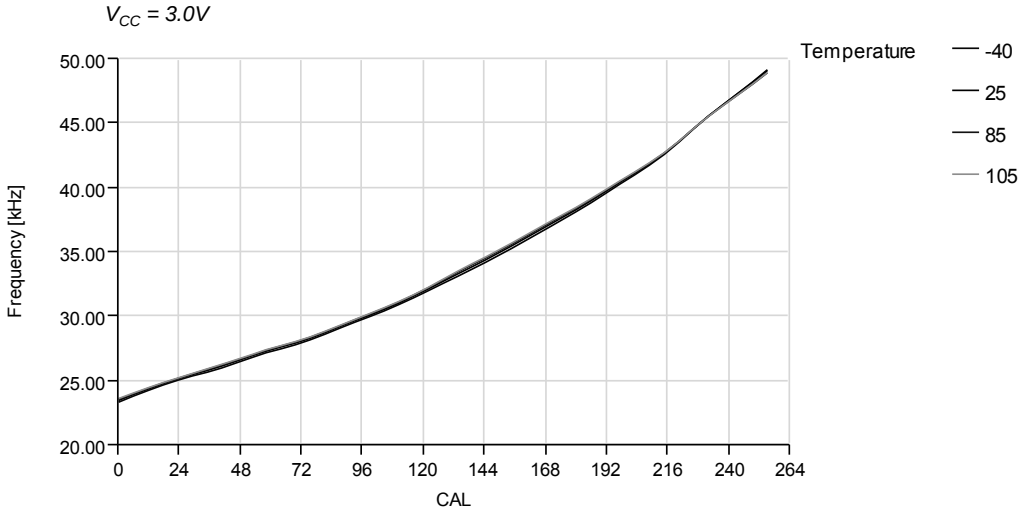
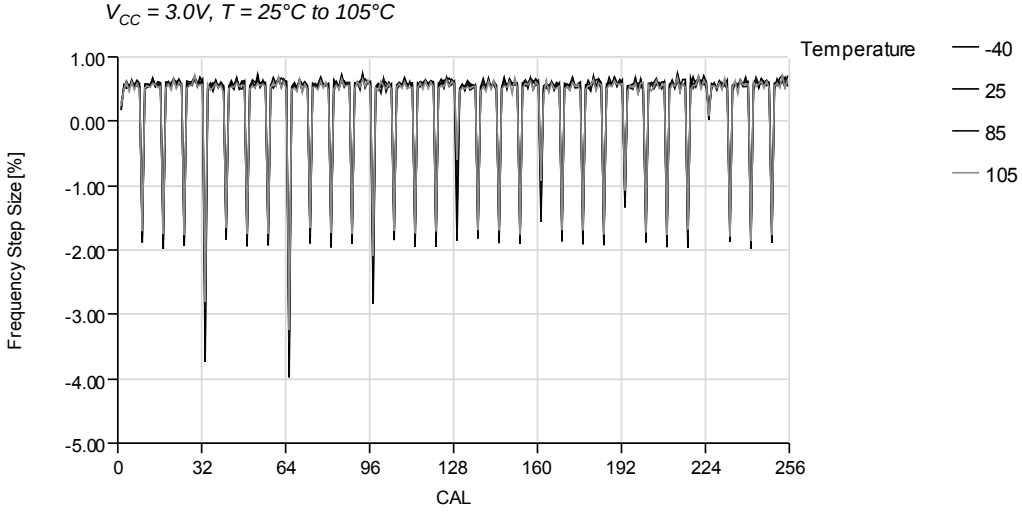


Figure 37-73. 32.768kHz Internal Oscillator Calibration Step Size



37.10.4 32MHz Internal Oscillator

Figure 37-78. 32MHz Internal Oscillator Frequency vs. Temperature

DFLL disabled

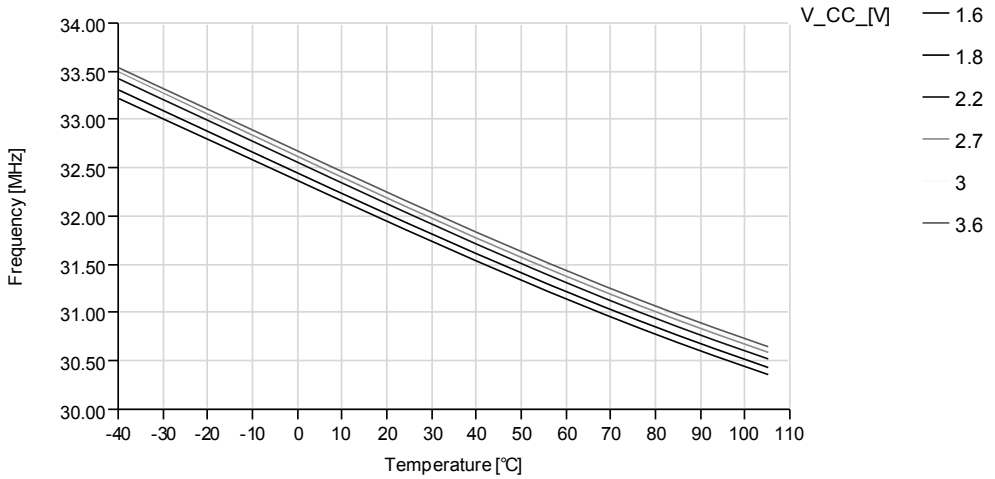


Figure 37-79. 32MHz Internal Oscillator Frequency vs. Temperature

DFLL enabled, from the 32.768kHz internal oscillator

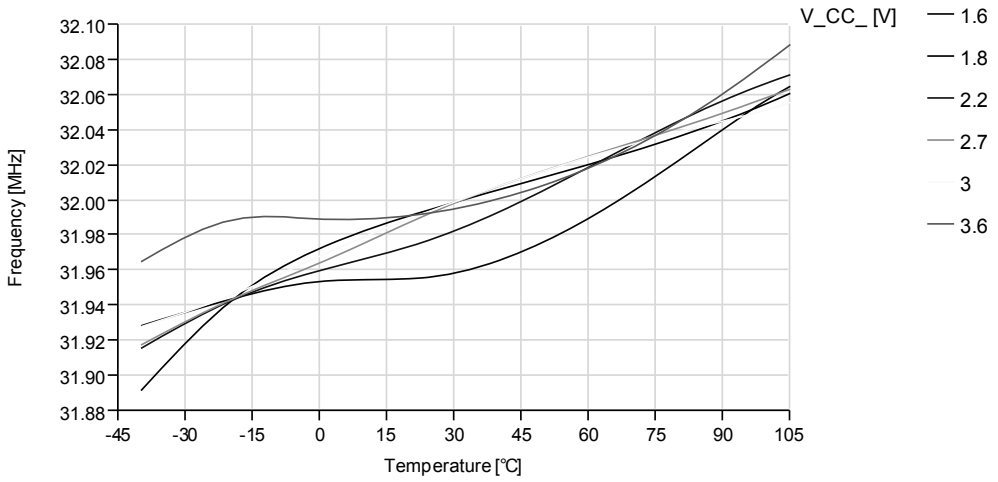
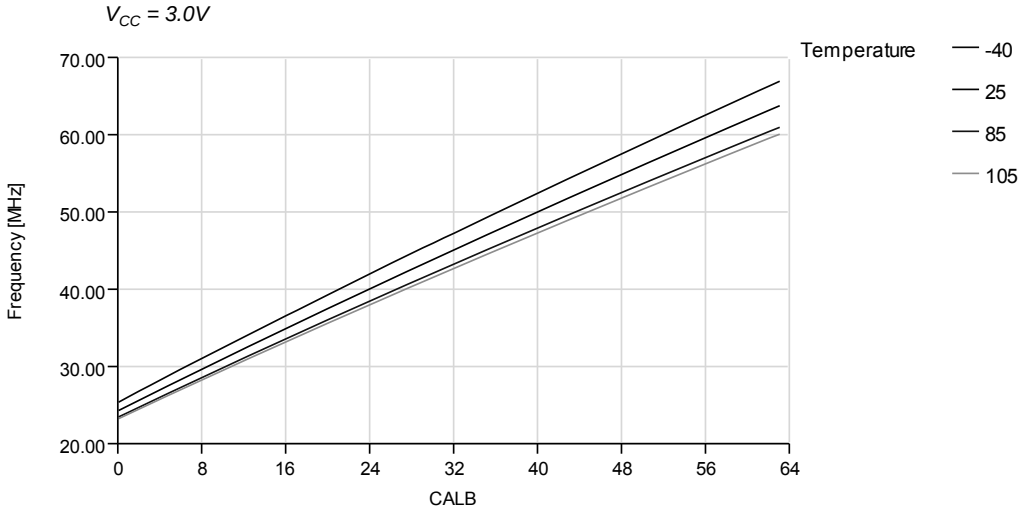
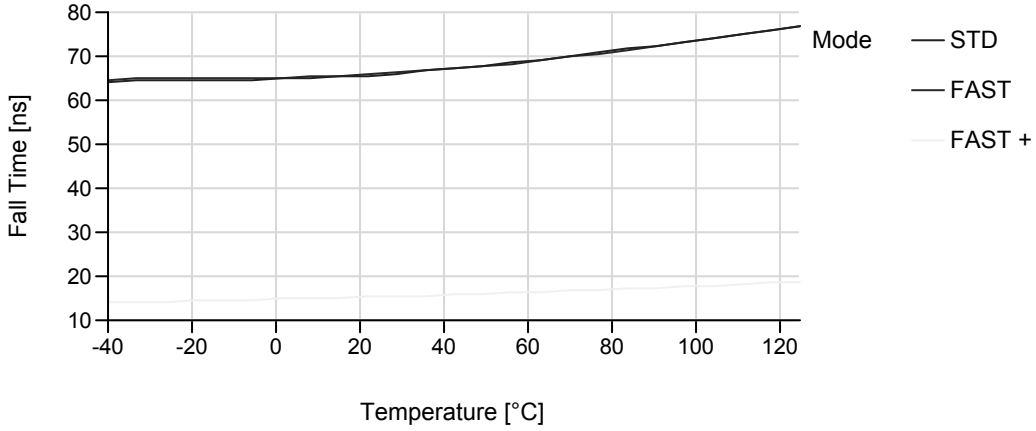


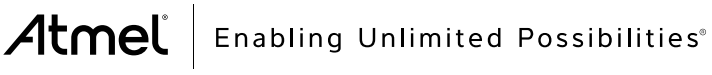
Figure 37-82. 32MHz internal Oscillator Frequency vs. CALB Calibration Value



37.11 Two-wire Interface Characteristics

Figure 37-83. SDA Fall Time vs. Temperature





Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA**T:** (+1)(408) 441.0311**F:** (+1)(408) 436.4200 **|www.atmel.com**