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, USB
Peripherals	Brown-out Detect/Reset, DMA, LCD, POR, PWM, WDT
Number of I/O	53
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	1.6V ~ 3.6V
Data Converters	A/D 16x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TFBGA
Supplier Device Package	100-CBGA (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atxmega128b1-cu

- Operating voltage
  - 1.6 – 3.6V
- Operating frequency
  - 0 – 12MHz from 1.6V
  - 0 – 32MHz from 2.7V

1. Ordering Information

Ordering Code	Flash [Bytes]	EEPROM [Bytes]	SRAM [Bytes]	Speed [MHz]	Power supply	Package[1][2][3]	Temp.
ATxmega128B1-AU	128K + 8K	2K	8K	32	1.6 - 3.6V	100A	-40°C - 85°C
ATxmega128B1-AUR[4]							
ATxmega128B1-CU						7A1	
ATxmega128B1-CUR[4]							
ATxmega128B1-AN						100A	-40°C - 105°C
ATxmega128B1-ANR[4]							
ATxmega64B1-AU	64K + 4K	2K	4K	32	1.6 - 3.6V	100A	-40°C - 85°C
ATxmega64B1-AUR[4]							
ATxmega64B1-CU						7A1	
ATxmega64B1-CUR[4]							
ATxmega64B1-AN						100A	-40°C - 105°C
ATxmega64B1-ANR[4]							

Notes:
1. This device can also be supplied in wafer form. 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 “Errata” on page 136.
4. Tape and Reel.

Package Type	
100A	100-lead, 14 x 14mm body size, 1.0mm body thickness, 0.5mm lead pitch, thin profile plastic quad flat package (TQFP)
7A1	100-ball (10x10 array), 7 x 7 x 1.0mm body, ball pitch 0.65mm, very thin fine-pitch ball grid array (VFBGA)

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 and SRAM. In addition, the EEPROM can be memory mapped in the data memory.

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 memory mapping of 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.

6.4 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 are supported, and the instruction set allows for efficient implementation of 32-bit arithmetic. The hardware multiplier supports signed and unsigned multiplication and fractional format.

6.4.1 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. Memories

7.1 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
    - Optional memory mapping for direct load and store
  - I/O memory
    - Configuration and status registers for all peripherals and modules
    - Four bit-accessible general purpose registers for global variables or flags
  - Bus arbitration
    - Safe and deterministic handling of priority between CPU, DMA controller, and other bus masters
  - Separate buses for SRAM, EEPROM and I/O memory
    - Simultaneous bus access for CPU and DMA 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

7.2 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.

11. Power Management and Sleep Modes

11.1 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

11.2 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.

11.3 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.

11.3.1 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 DMA controller are kept running. Any enabled interrupt will wake the device.

11.3.2 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, asynchronous port interrupts, and the USB resume interrupt.

13. WDT – Watchdog Timer

13.1 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

13.2 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. I/O Ports

15.1 Features

- 53 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 synchronous and/or 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
- Asynchronous pin change sensing that can wake the device from all sleep modes
- Two port interrupts 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, SPI, and timer/counter input/output pin locations

15.2 Overview

One port consists of up to eight port pins: pin 0 to 7. Each port pin can be configured as input or output with configurable driver and pull settings. They also implement synchronous and asynchronous input sensing with interrupts and events for selectable pin change conditions. Asynchronous pin-change sensing means that a pin change can wake the device from all sleep modes, included 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.

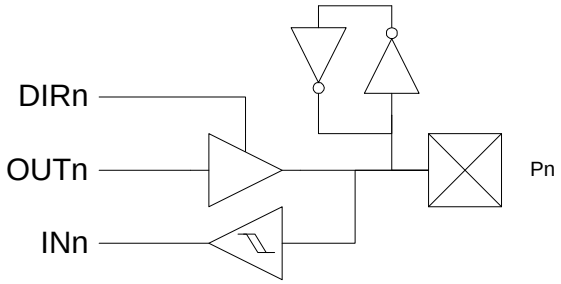
The port pin configuration also controls input and output selection of other device functions. It is possible to have both the peripheral clock and the real-time clock output to a port pin, and available for external use. The same applies to events from the event system that can be used to synchronize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

The notation of the ports are PORTA, PORTB, PORTC, PORTD, PORTE, PORTG, PORTM, and PORTR.

15.3 Output Driver

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

Figure 15-4. I/O Configuration - Totem-pole with Bus-keeper



15.3.5 Others

Figure 15-5. Output Configuration - Wired-OR with Optional Pull-down

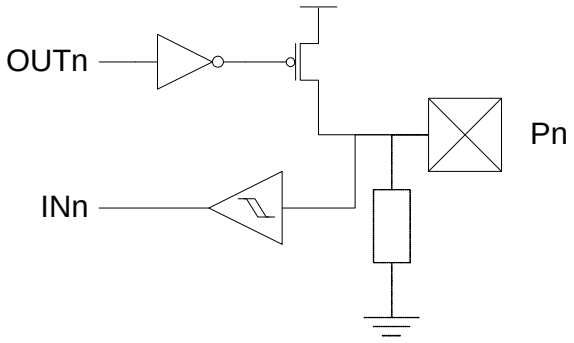
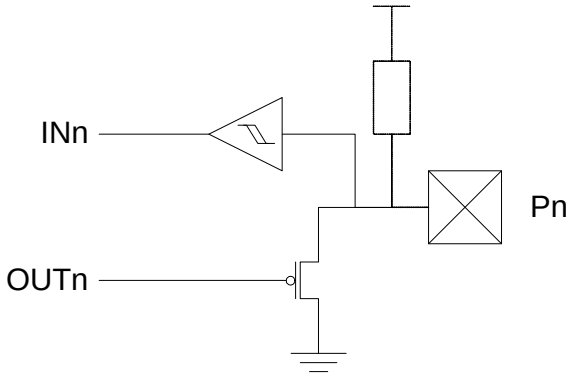


Figure 15-6. I/O Configuration - Wired-AND with Optional Pull-up



Multi-packet transfer enables a data payload exceeding the maximum packet size of an endpoint to be transferred as multiple packets without software intervention. This reduces the CPU intervention and the interrupts needed for USB transfers.

For low-power operation, the USB module can put the microcontroller into any sleep mode when the USB bus is idle and a suspend condition is given. Upon bus resumes, the USB module can wake up the microcontroller from any sleep mode.

PORTD has one USB. Notation of this is USB.

22. TWI – Two-wire Interface

22.1 Features

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

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

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

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

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

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

PORTC has one TWI. Notation of this peripheral is TWIC.

24. USART

24.1 Features

- Two Identical USART peripherals
- Full-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 and 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 multi-device bus
  - Enable unaddressed devices to automatically ignore all frames
- 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

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 communication and asynchronous and synchronous 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.

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.

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. Pin control and interrupt generation are identical in both modes. The registers are used in both modes, but their functionality differs for some control settings.

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.

PORTC and PORTE each has one USART. Notation of these peripherals are USARTC0 and USARTE0 respectively.

32. Pinout and Pin Functions

The device pinout is shown in “Pinout/Block Diagram” on page 4. 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
AGND	Analog 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
AREF	Analog Reference input pin

32.1.4 LCD Functions

SEGn	LCD Segment Drive Output n
COMn	LCD Common Drive Output n
VLCD	LCD Voltage Multiplier Output
BIAS2	LCD Intermediate Voltage 2 Output (VLCD * 2/3)
BIAS1	LCD Intermediate Voltage 1 Output (VLCD * 1/3)
CAPH	LCD High End Of Flying Capacitor
CAPL	LCD Low End Of Flying Capacitor

32.1.5 Timer/Counter and AWEX Functions

OCnxLS	Output Compare Channel x Low Side for Timer/Counter n
OCnxHS	Output Compare Channel x High Side for Timer/Counter n

32.1.6 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
SS	Slave Select for SPI
MOSI	Master Out Slave In for SPI
MISO	Master In Slave Out for SPI
SCK	Serial Clock for SPI
D-	Data- for USB
D+	Data+ for USB

32.1.7 Oscillators, Clock, and Event

TOSCn	Timer Oscillator pin n
XTALn	Input/Output for Oscillator pin n
CLKOUT	Peripheral Clock Output
EVOUT	Event Channel 0 Output
RTCOUT	RTC Clock Source Output

32.1.8 Debug/System Functions

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

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
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	
	ADC settling time	After changing reference or input mode		7	7	

Table 36-11. 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, 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	
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

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			
		V_{CC} = 2.7 - 3.6V	2.4			
t_{CL}	Clock low time	V_{CC} = 1.6 - 1.8V	4.5			
		V_{CC} = 2.7 - 3.6V	2.4			
t_{CR}	Rise time (for maximum frequency)	V_{CC} = 1.6 - 1.8V			1.5	
		V_{CC} = 2.7 - 3.6V			1.0	
t_{CF}	Fall time (for maximum frequency)	V_{CC} = 1.6 - 1.8V			1.5	
		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 1.8V and 2.7V, and the same applies for all other parameters with supply voltage conditions.

36.14.7 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		0		ns
		XOSCPWR=0, FRQRANGE=1, 2, or 3		0		
		XOSCPWR=1		0		
	Long term jitter	XOSCPWR=0, FRQRANGE=0		0		
		XOSCPWR=0, FRQRANGE=1, 2, or 3		0		
		XOSCPWR=1		0		
	Frequency error	XOSCPWR=0, FRQRANGE=0		0.03		%
		XOSCPWR=0, FRQRANGE=1		0.03		
		XOSCPWR=0, FRQRANGE=2 or 3		0.03		
		XOSCPWR=1		0.03		
	Duty cycle	XOSCPWR=0, FRQRANGE=0		50		
		XOSCPWR=0, FRQRANGE=1		50		
		XOSCPWR=0, FRQRANGE=2 or 3		50		
		XOSCPWR=1		50		

Symbol	Parameter	Condition			Min.	Typ.	Max.	Units
C_{XTAL1}	Parasitic capacitance					5.9		pF
C_{XTAL2}	Parasitic capacitance					8.3		
C_{LOAD}	Parasitic capacitance load					3.5		

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

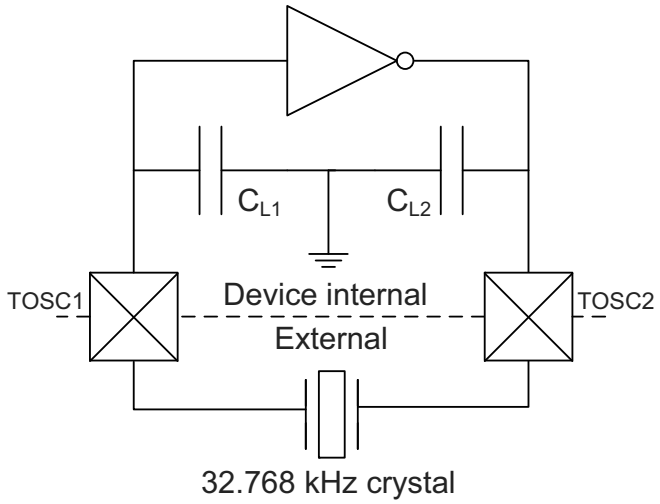
36.14.8 External 32.768kHz Crystal Oscillator and TOSC Characteristics

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	
		Crystal load capacitance 12.0pF			28	
C_{IN_TOSC}	Input capacitance between TOSC pins	Normal mode		3.5		pF
		Low power mode		3.5		
	Recommended safety factor	capacitance load matched to crystal specification	3			
	Long term Jitter (SIT)			0		%

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

Figure 36-4. TOSC Input Capacitance



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

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

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

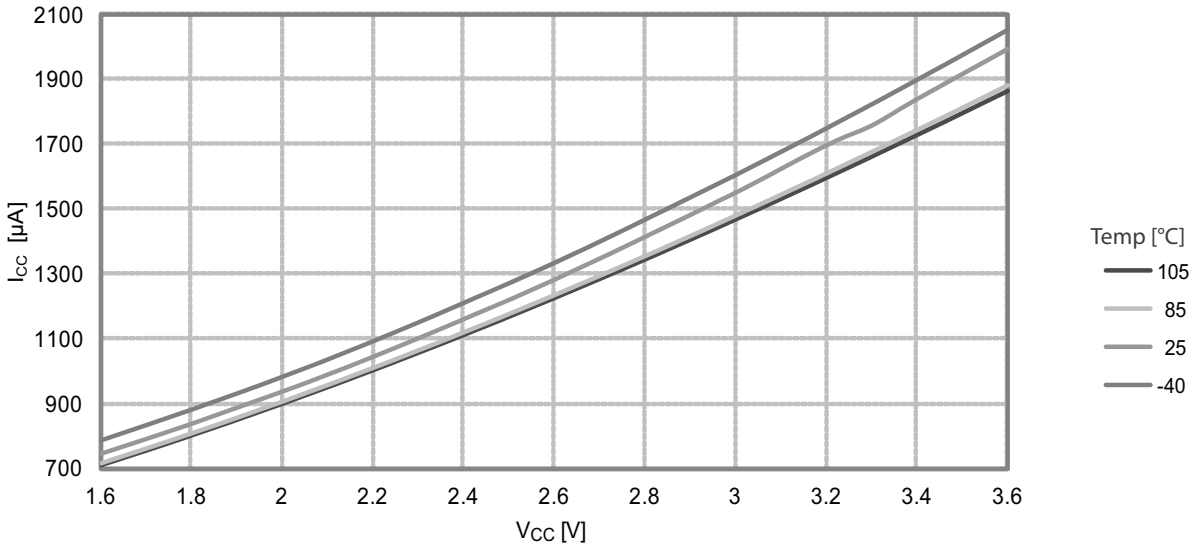


Figure 37-12.Idle Mode Current vs. V_{CC}

***f_{SYS} = 32MHz internal oscillator*.**

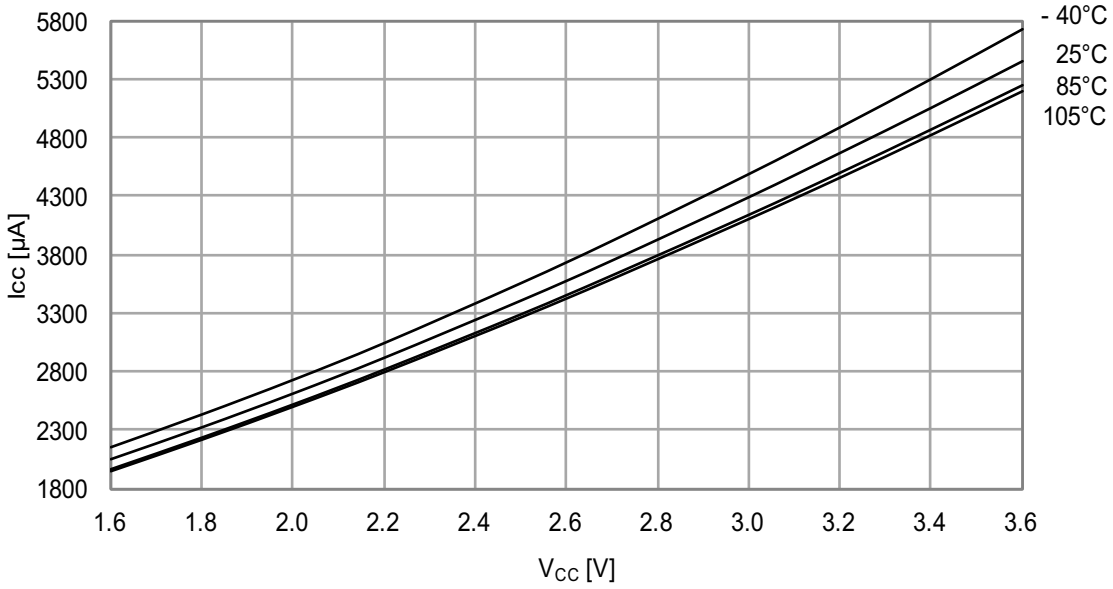
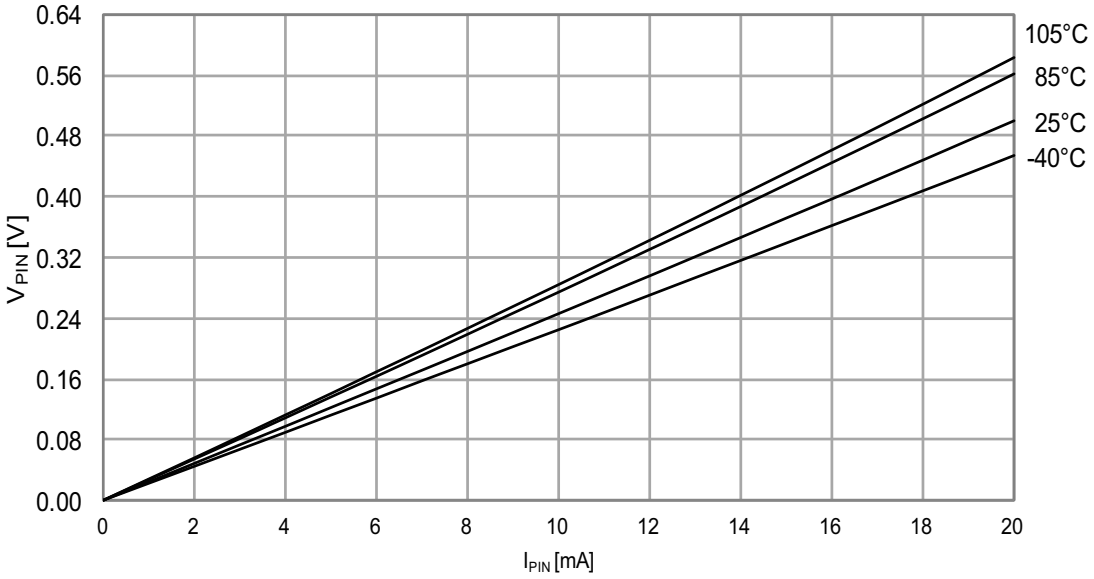


Figure 37-25. I/O Pin Output Voltage vs. Sink Current

V_{CC} = 3.3V.



37.2.3 Thresholds and Hysteresis

Figure 37-26. I/O Pin Input Threshold Voltage vs. V_{CC}

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

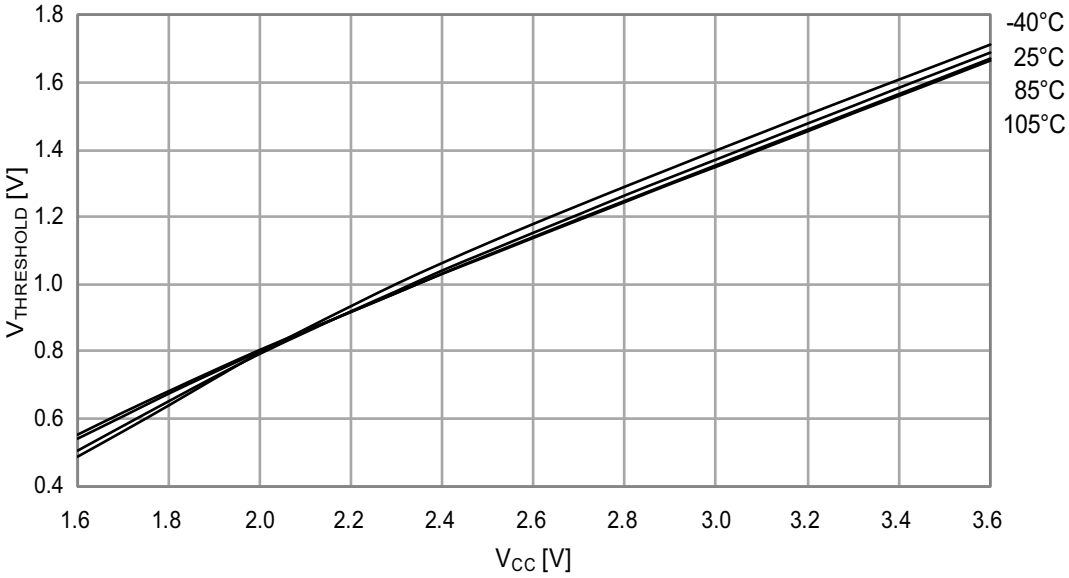


Figure 37-27. I/O Pin Input Threshold Voltage vs. V_{CC}

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

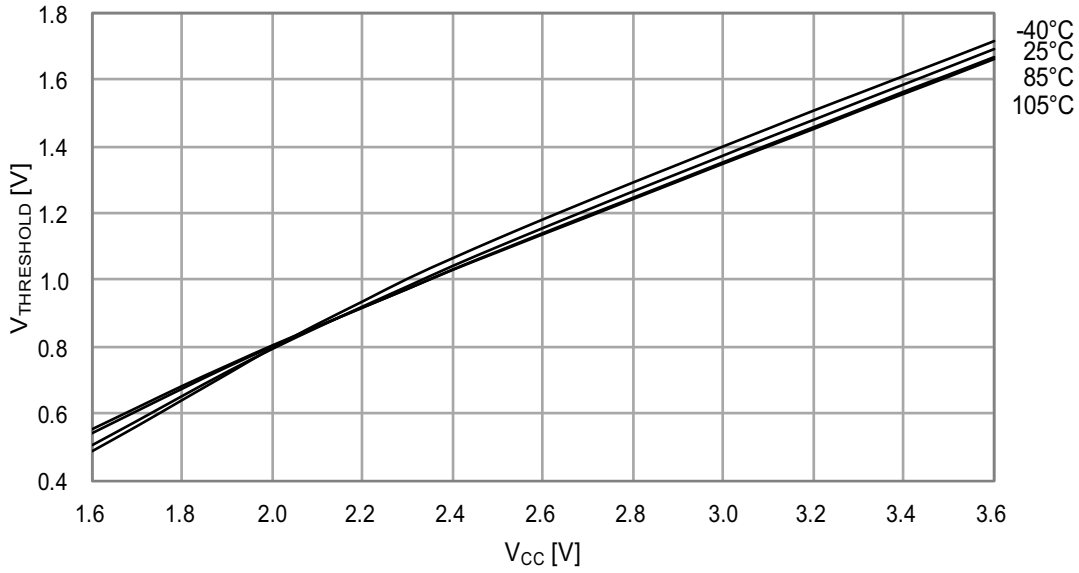


Figure 37-28. I/O Pin Input Hysteresis vs. V_{CC}.

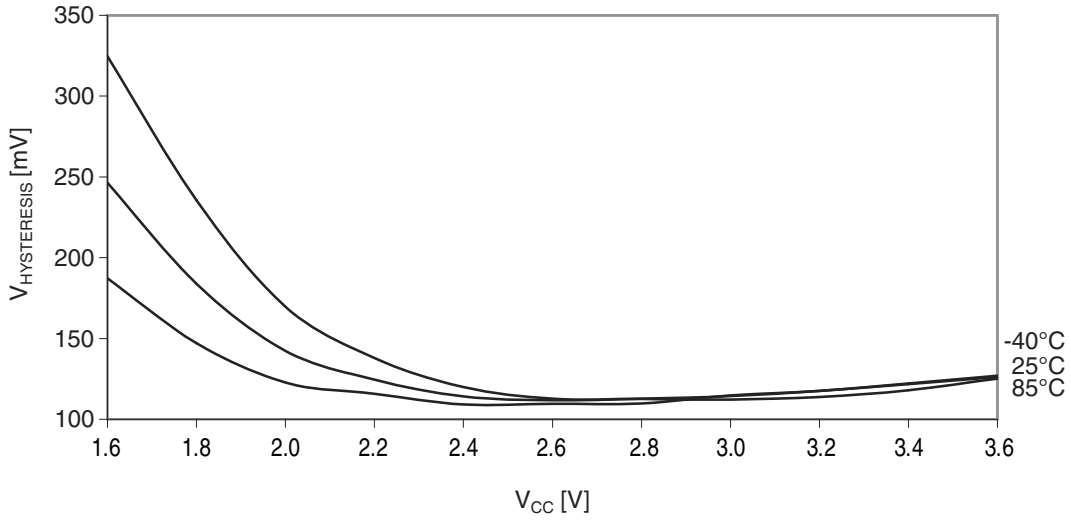


Figure 37-87. P_{SAVE} LCD LP 32Hz vs. Temperature
RTC, WDT, BOD sampled.

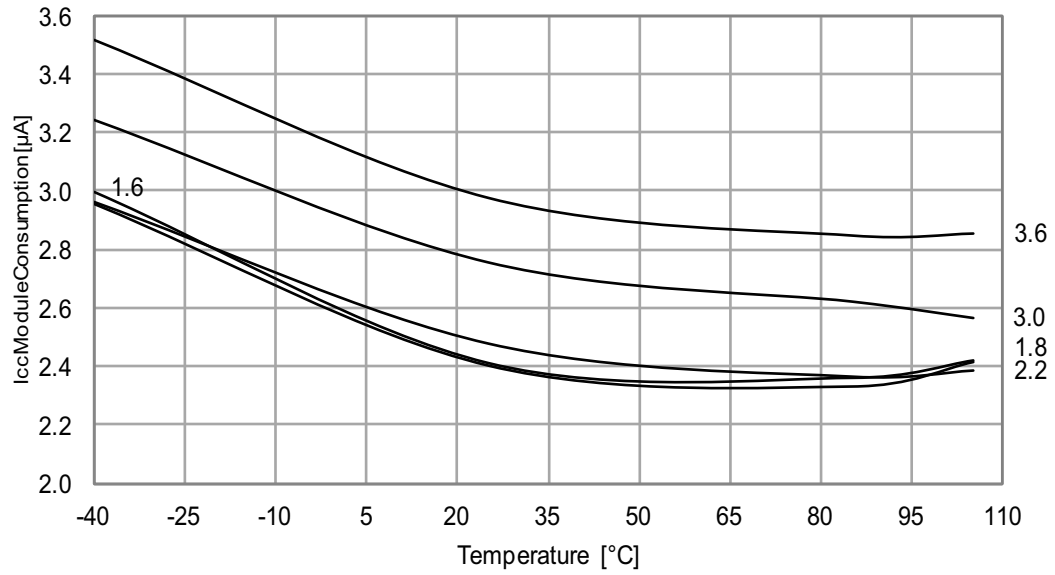
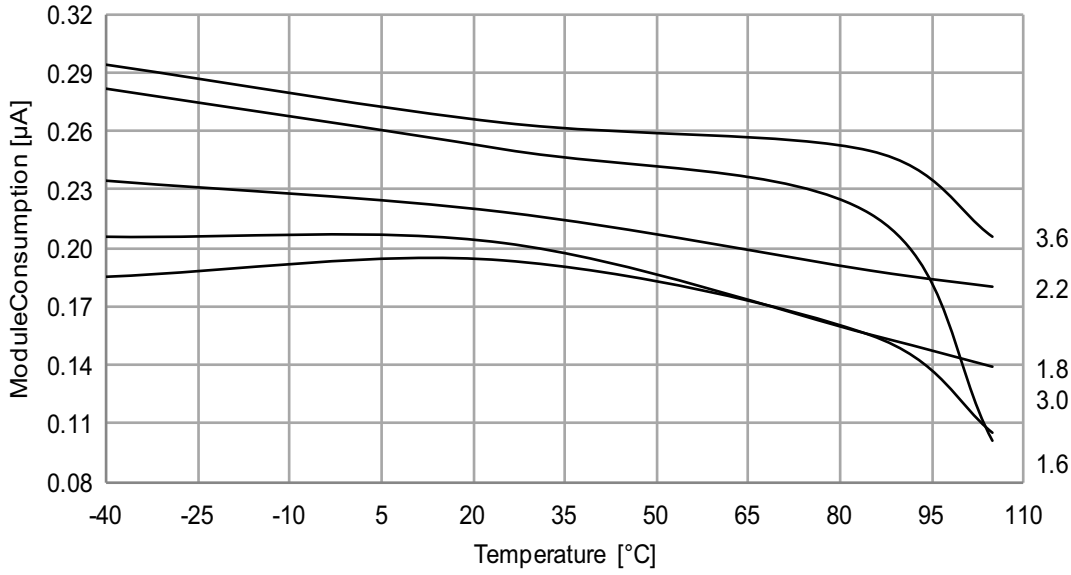


Figure 37-88. P_{SAVE} vs. Temperature
***RTC, WDT, BOD sampled*.**



38. Errata

38.1 ATxmega64B1, ATxmega128B1

38.1.1 Rev. C

- **Device revision number**
- **AWeX fault protection restore is not done correct in Pattern Generation Mode**

1. Device revision number is unchanged between rev. B and rev. C

2. AWeX fault protection restore is not done correctly in Pattern Generation Mode

When a fault is detected the OUTOVEN register is cleared, and when fault condition is cleared, OUTOVEN is restored according to the corresponding enabled DTI channels. For Common Waveform Channel Mode (CWCM), this has no effect as the OUTOVEN is correct after restoring from fault. For Pattern Generation Mode (PGM), OUTOVEN should instead have been restored according to the DTILSBUF register.

Problem fix/Workaround

For CWCM no workaround is required.

For PGM in latched mode, disable the DTI channels before returning from the fault condition. Then, set correct OUTOVEN value and enable the DTI channels, before the direction (DIR) register is written to enable the correct outputs again.

For PGM in cycle-by-cycle mode there is no workaround.

38.1.2 Rev. B

Not sampled.

38.1.3 Rev. A

- **Power down consumption**
- **ADC conversion error when x0.5 gain is used**
- **Disabling of USART transmitter does not automatically set the TxD pin direction to input**

1. Power Down consumption

After reset, when system enters in power down or when ADC is disabled, extra power consumption is drawn.

Problem fix/Workaround

Set ADC to a configuration different from differential mode.

2. ADC conversion error when x0.5 gain is used

When the gain is set to x0.5, the conversion result is similar to the gain setting x1.

Problem fix/Workaround

There is no workaround.

3. Disabling of USART transmitter does not automatically set the TxD pin direction to input

If the USART transmitter is idle with no frames to transmit, setting TXEN to zero will not automatically set the TxD pin direction to input.