Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

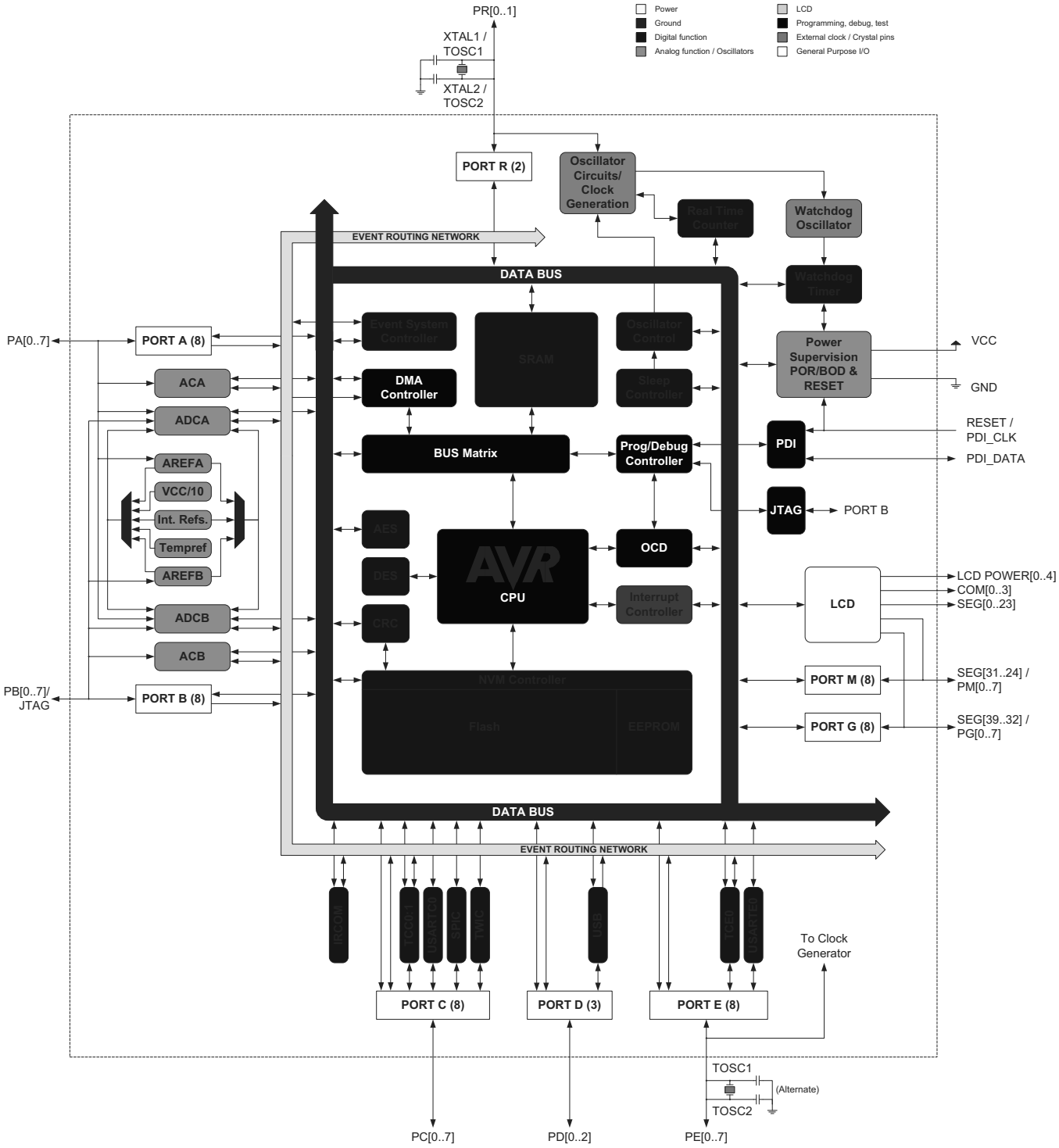
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	1.6V ~ 3.6V
Data Converters	A/D 16x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atxmega64b1-anr

3.1 Block Diagram

Figure 3-1. XMEGA B1 Block Diagram



6. AVR CPU

6.1 Features

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

6.2 Overview

All Atmel AVR XMEGA devices use the 8/16-bit AVR CPU. The main function of the CPU is to execute the code and perform all calculations. The CPU is able to access memories, perform calculations, control peripherals, and execute the program in the flash memory. Interrupt handling is described in a separate section, refer to “Interrupts and Programmable Multilevel Interrupt Controller” on page 28.

6.3 Architectural Overview

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

Figure 6-1. Block Diagram of the AVR CPU Architecture

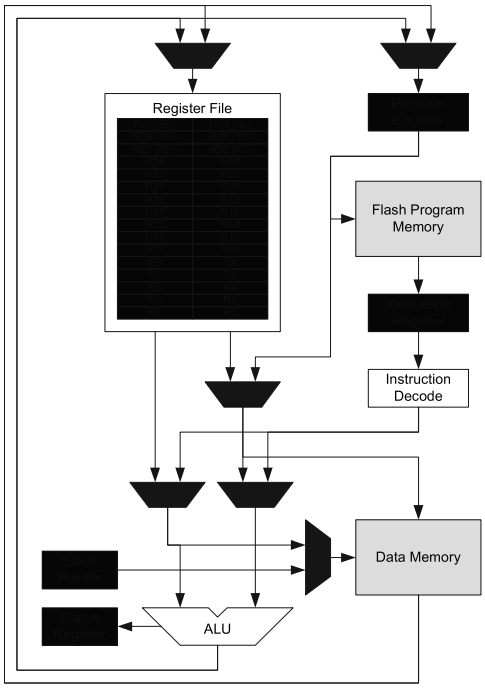
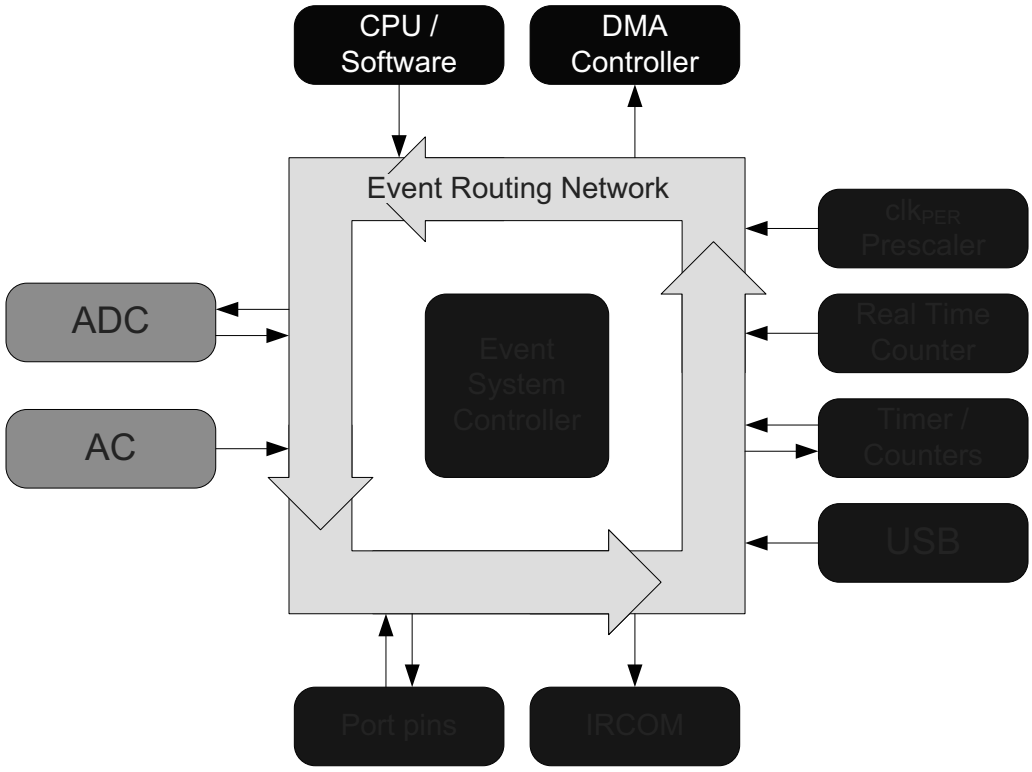


Figure 9-1. Event System Overview and Connected Peripherals



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

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.

27. CRC – Cyclic Redundancy Check Generator

27.1 Features

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

27.2 Overview

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

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

- **CRC-16:**

Polynomial:	$x^{16}+x^{12}+x^5+1$
Hex value:	0x1021

- **CRC-32:**

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

31. Programming and Debugging

31.1 Features

- Programming
  - External programming through PDI or JTAG interfaces
    - 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
  - Non-intrusive, 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
- JTAG interface
  - Four-pin, IEEE Std. 1149.1 compliant interface for programming and debugging
  - Boundary scan capabilities according to IEEE Std. 1149.1 (JTAG)

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 non-intrusive, 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 disassembler level.

Programming and debugging can be done through two physical interfaces. The primary one is the PDI physical layer, which is available on all devices. 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). A JTAG interface is also available on most devices, and this can be used for programming and debugging through the four-pin JTAG interface. The JTAG interface is IEEE Std. 1149.1 compliant, and supports boundary scan. Any external programmer or on-chip debugger/emulator can be directly connected to either of these interfaces. Unless otherwise stated, all references to the PDI assume access through the PDI physical layer.

Table 32-3. Port C - Alternate Functions

PORT C	PIN #	INTERRUPT	TCC0 (1)	AWEXC	TCC1	USARTC0 (2)	SPIC (3)	TWIC	EXTCLK	CLOCKOUT (4)	EVENTOUT (5)
GND	100										
VCC	1										
PC0	2	SYNC	OC0A	$\overline{\text{OC0ALS}}$				SDA	EXTCLKC0		
PC1	3	SYNC	OC0B	OC0AHS		XCK0		SCL	EXTCLKC1		
PC2	4	SYNC/ASYNC	OC0C	$\overline{\text{OC0BLS}}$		RXD0			EXTCLKC2		
PC3	5	SYNC	OC0D	OC0BHS		TXD0			EXTCLKC3		
PC4	6	SYNC		$\overline{\text{OC0CLS}}$	OC1A		$\overline{\text{SS}}$		EXTCLKC4		
PC5	7	SYNC		OC0CHS	OC1B		MOSI		EXTCLKC5		
PC6	8	SYNC		$\overline{\text{OC0DLS}}$			MISO		EXTCLKC6	RTCOUT	
PC7	9	SYNC		OC0DHS			SCK		EXTCLKC7	clk_{PER}	EVOUT

Notes:
1. Pin mapping of all TC0 can optionally be moved to high nibble of port.
2. Pin mapping of all USART0 can optionally be moved to high nibble of port.
3. Pins MOSI and SCK for all SPI can optionally be swapped.
4. CLKOUT can optionally be moved between port C and E and between pin 4 and 7.
5. EVOUT can optionally be moved between port C and E and between pin 4 and 7.

Table 32-4. Port D - Alternate Functions

PORT D	PIN #	INTERRUPT	USBD
GND	10		
VCC	11		
PD0	12	SYNC	D-
PD1	13	SYNC	D+
PD2	14	SYNC/ASYNC	

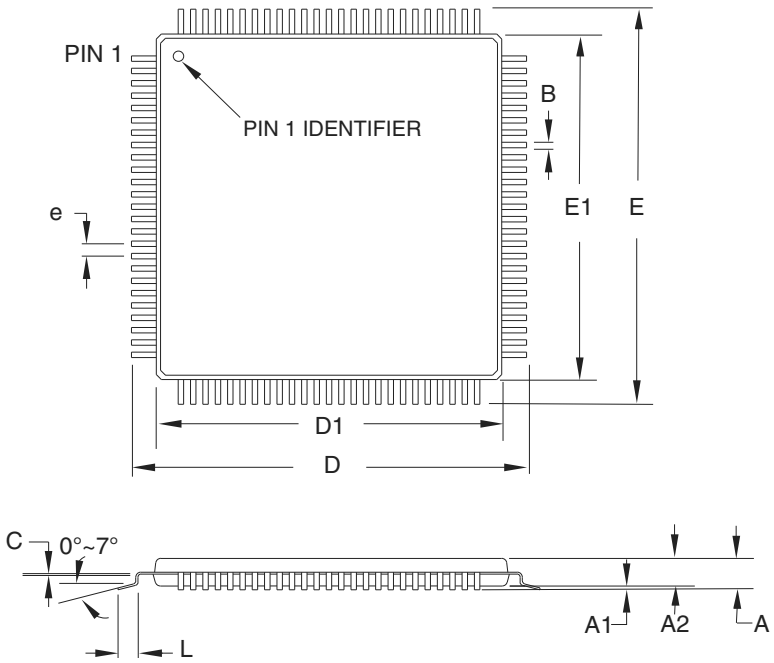
Table 32-5. Program and Debug Functions

PROG	PIN #	INTERRUPT	PROG
$\overline{\text{RESET}}$	15		PDI_CLK
PDI	16		PDI_DATA
GND	17		
VCC	18		

Mnemonics	Operands	Description	Operation	Flags	#Clocks
ELPM	Rd, Z	Extended Load Program Memory	Rd ← (RAMPZ:Z)	None	3
ELPM	Rd, Z+	Extended Load Program Memory and Post-Increment	Rd ← (RAMPZ:Z), Z ← Z + 1	None	3
SPM		Store Program Memory	(RAMPZ:Z) ← R1:R0	None	-
SPM	Z+	Store Program Memory and Post-Increment by 2	(RAMPZ:Z) ← R1:R0, Z ← Z + 2	None	-
IN	Rd, A	In From I/O Location	Rd ← I/O(A)	None	1
OUT	A, Rr	Out To I/O Location	I/O(A) ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	1 (1)
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2 (1)
XCH	Z, Rd	Exchange RAM location	Temp ← Rd, Rd ← (Z), (Z) ← Temp	None	2
LAS	Z, Rd	Load and Set RAM location	Temp ← Rd, Rd ← (Z), (Z) ← Temp v (Z)	None	2
LAC	Z, Rd	Load and Clear RAM location	Temp ← Rd, Rd ← (Z), (Z) ← ($FFh – Rd) • (Z)	None	2
LAT	Z, Rd	Load and Toggle RAM location	Temp ← Rd, Rd ← (Z), (Z) ← Temp ⊕ (Z)	None	2
Bit and bit-test instructions					
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0, C ← Rd(7)	Z,C,N,V,H	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0, C ← Rd(0)	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0) ← C, Rd(n+1) ← Rd(n), C ← Rd(7)	Z,C,N,V,H	1
ROR	Rd	Rotate Right Through Carry	Rd(7) ← C, Rd(n) ← Rd(n+1), C ← Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0) ↔ Rd(7..4)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
SBI	A, b	Set Bit in I/O Register	I/O(A, b) ← 1	None	1
CBI	A, b	Clear Bit in I/O Register	I/O(A, b) ← 0	None	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1

35. Packaging Information

35.1 100A



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	15.75	16.00	16.25	
D1	13.90	14.00	14.10	Note 2
E	15.75	16.00	16.25	
E1	13.90	14.00	14.10	Note 2
B	0.17	–	0.27	
C	0.09	–	0.20	
L	0.45	–	0.75	
e	0.50 TYP			

Notes:
1. This package conforms to JEDEC reference MS-026, Variation AED.
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 0.08mm maximum.

2014-02-05

	TITLE	DRAWING NO.	REV.
Atmel Package Drawing Contact: packagedrawings@atmel.com	**100A**, 100-lead, 14 x 14mm Body Size, 1.0mm Body Thickness, 0.5mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP)	100A	E

36.8 Analog Comparator Characteristics

Table 36-13. Analog Comparator Characteristics

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

36.9 Bandgap and Internal 1.0V Reference Characteristics

Table 36-14. Bandgap and Internal 1.0V Reference Characteristics

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
	Startup time	As reference for ADC	1 Clk_{PER} + 2.5µs			µs
		As input voltage to ADC and AC		1.5		
	Bandgap voltage			1.1		V
INT1V	Internal 1.00V reference for ADC	T= 85°C, after calibration	0.99	1	1.01	
	Variation over voltage and temperature	Calibrated at T= 85°C		2.25		%

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
$t_{SU;DAT}$	Data setup time	$f_{SCL} \leq 100kHz$	250			ns
		$f_{SCL} > 100kHz$	100			
$t_{SU;STO}$	Setup time for STOP condition	$f_{SCL} \leq 100kHz$	4.0			µs
		$f_{SCL} > 100kHz$	0.6			
t_{BUF}	Bus free time between a STOP and START condition	$f_{SCL} \leq 100kHz$	4.7			
		$f_{SCL} > 100kHz$	1.3			

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

37.3 ADC Characteristics

Figure 37-29. INL Error vs. External V_{REF}

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

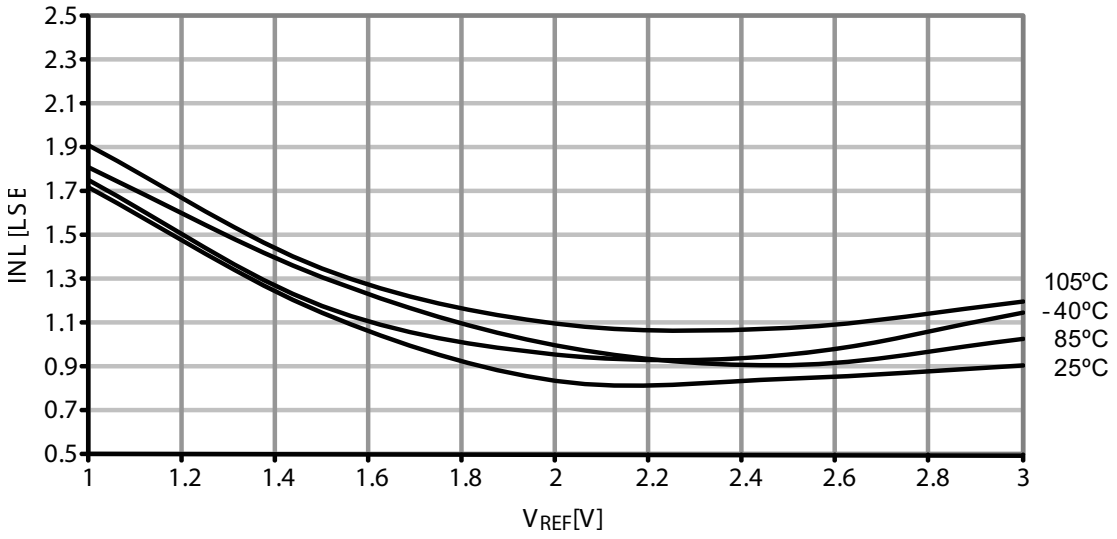
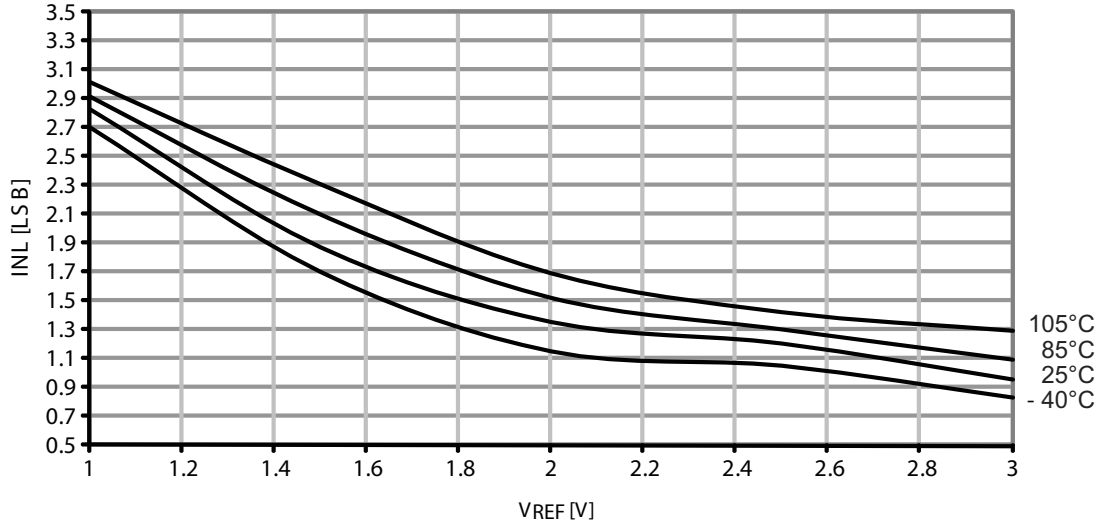


Figure 37-30. ADC INL vs. V_{REF}

SE Unsigned mode, V_{CC} = 3.6V external reference.



37.4 Analog Comparator Characteristics

Figure 37-43. Analog Comparator Hysteresis vs. V_{CC}
***High-speed mode, small hysteresis*.**

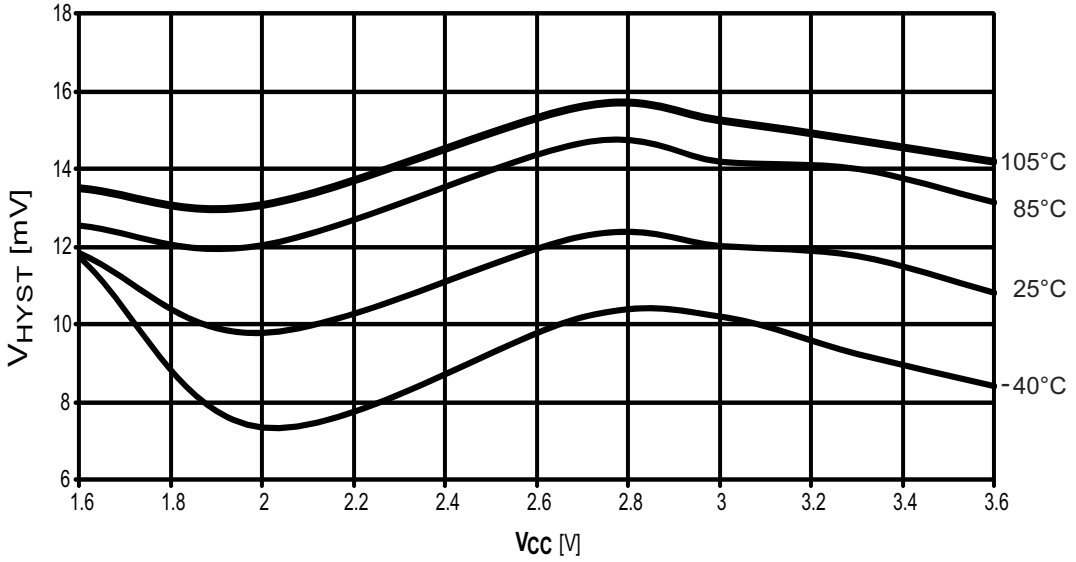


Figure 37-44. Analog Comparator Hysteresis vs. V_{CC}
***High-speed mode, large hysteresis*.**

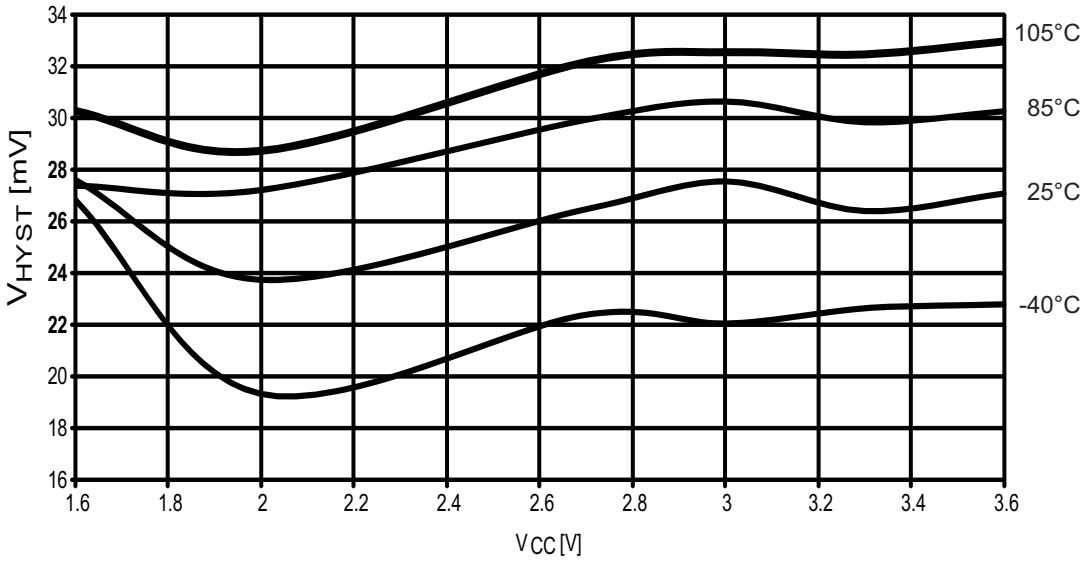
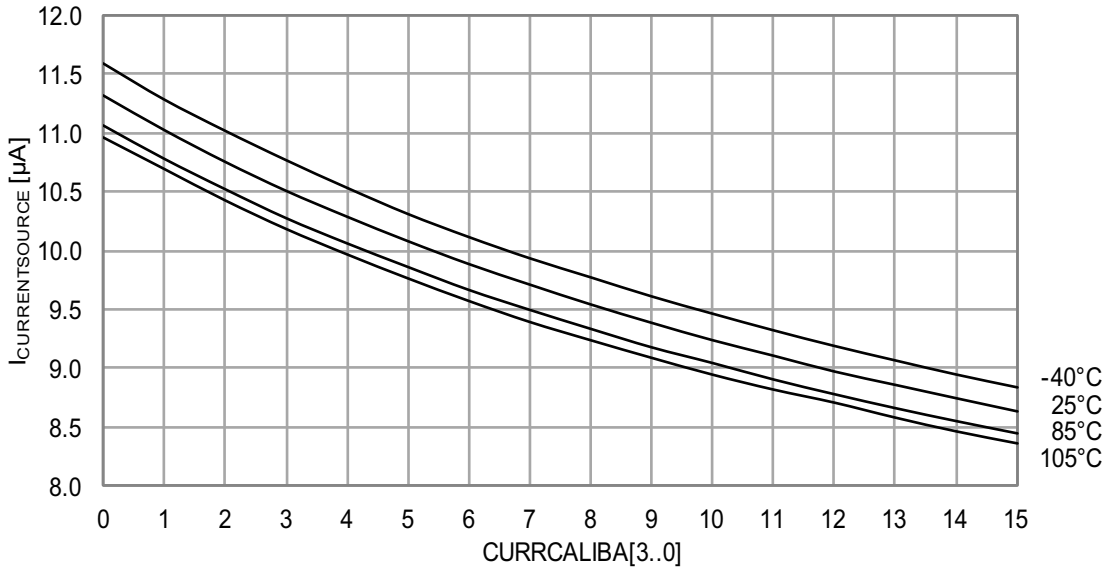


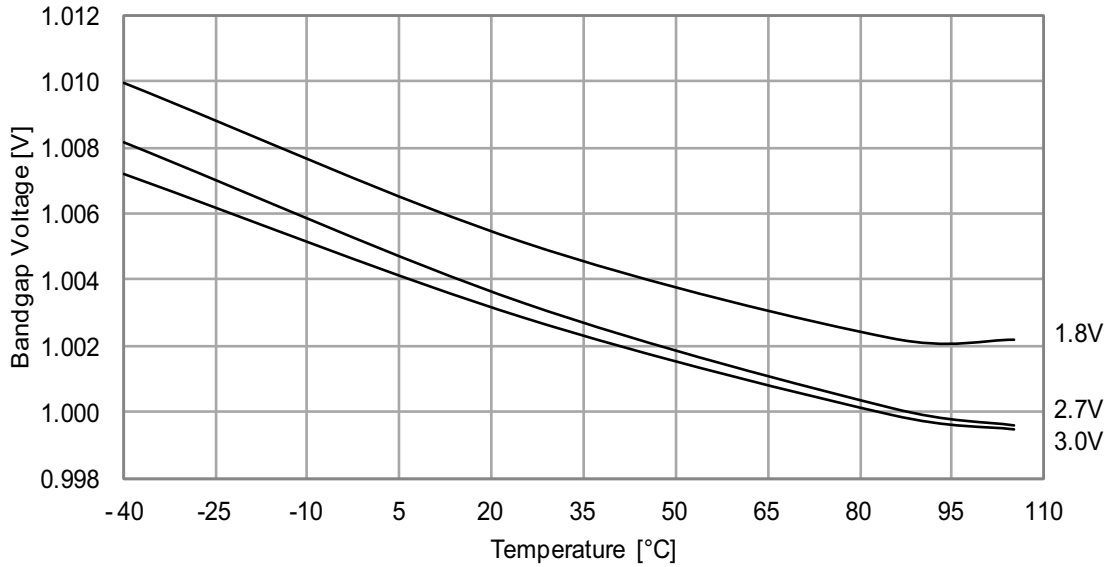
Figure 37-49. Analog Comparator Current Source vs. Calibration

***V_{CC} = 3.0V, double mode*.**



37.5 Internal 1.0V Reference Characteristics

Figure 37-50. ADC/DAC Internal 1.0V Reference vs. Temperature



37.6 BOD Characteristics

Figure 37-51. BOD Thresholds vs. Temperature

***BOD level = 1.6V*.**

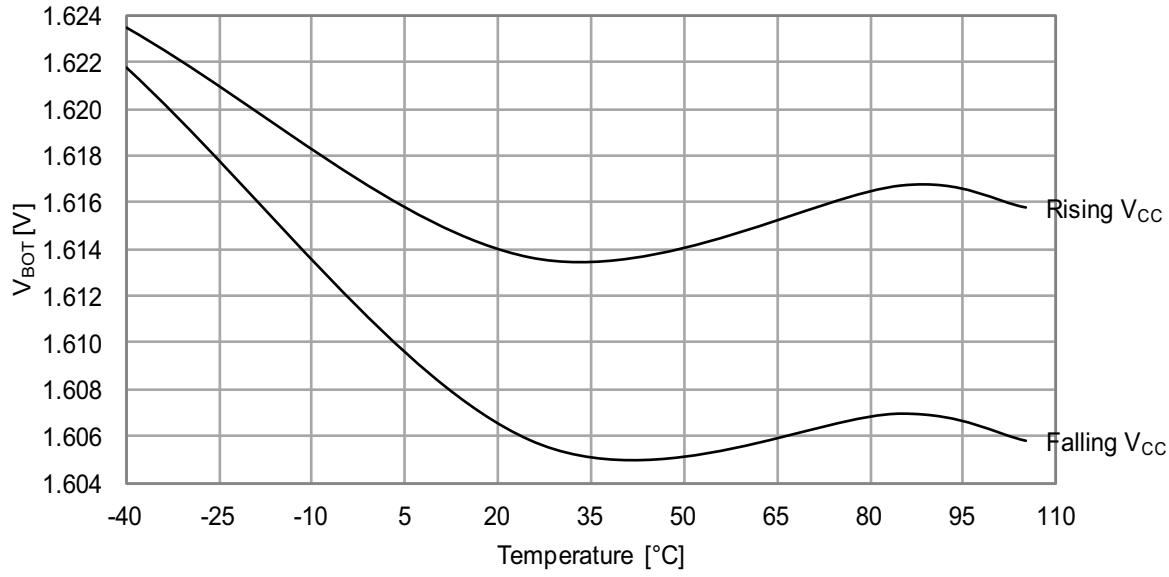


Figure 37-52. BOD Thresholds vs. Temperature

***BOD level = 2.2V*.**

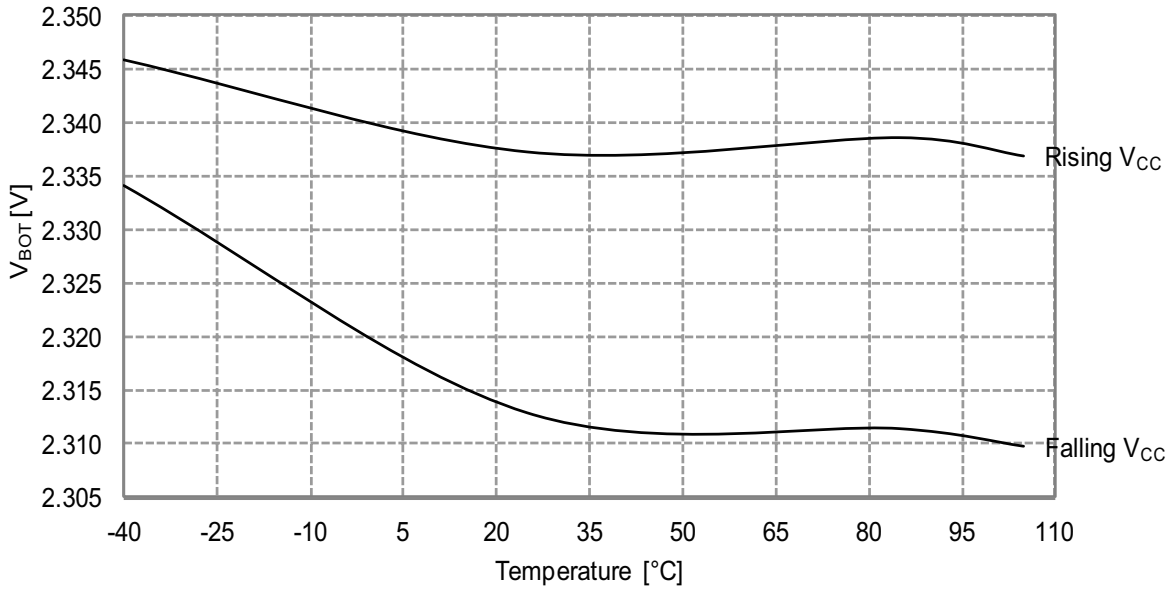
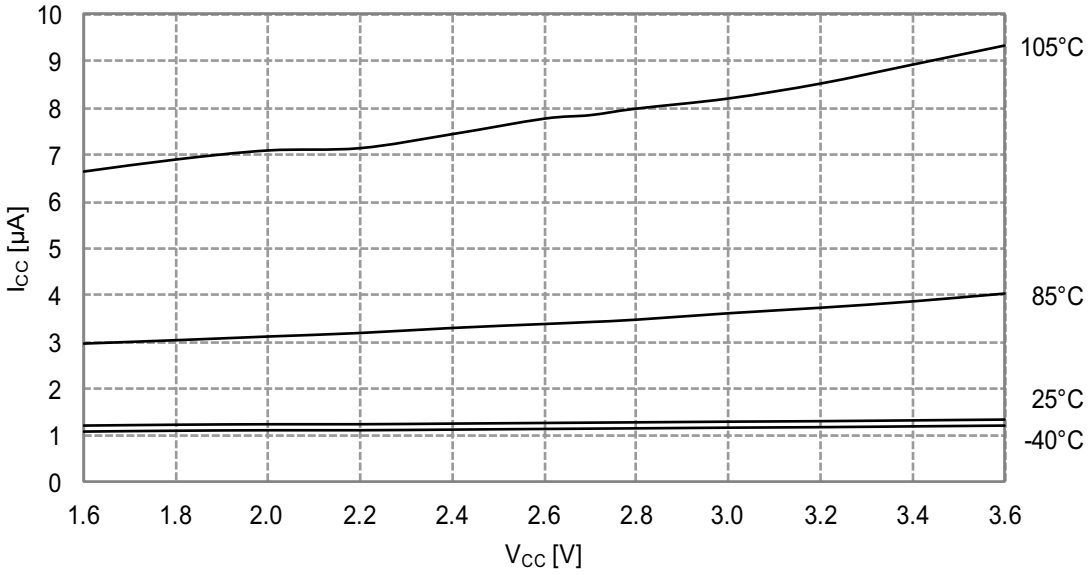


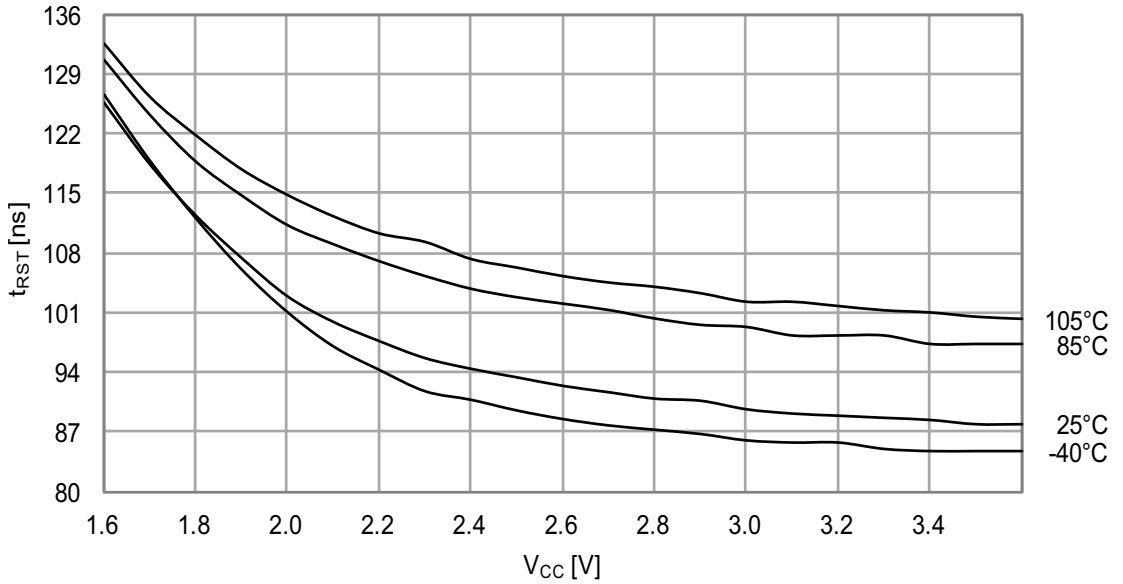
Figure 37-55. BOD Current Consumption vs. V_{CC}

***Sampled mode, BOD level = 1.6V*.**



37.7 External Reset Characteristics

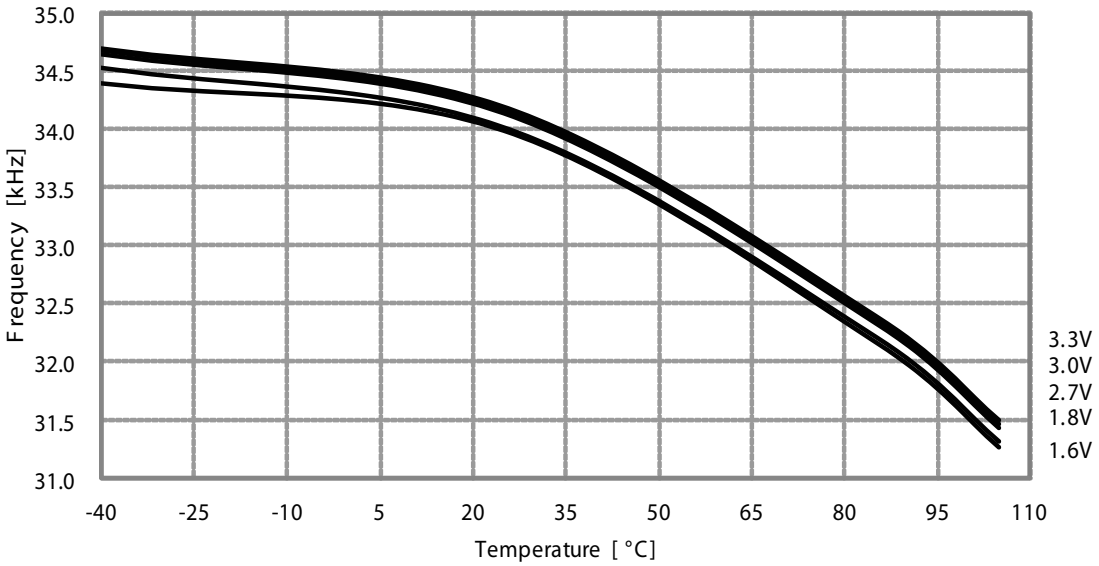
Figure 37-56. Minimum Reset Pin Pulse Width vs. V_{CC}



37.8 Oscillator Characteristics

37.8.1 Ultra Low-power Internal Oscillator

Figure 37-62. Ultra Low-power Internal Oscillator Frequency vs. Temperature.



37.8.2 32.768kHz Internal Oscillator

Figure 37-63. 32.768kHz Internal Oscillator Frequency vs. Temperature.

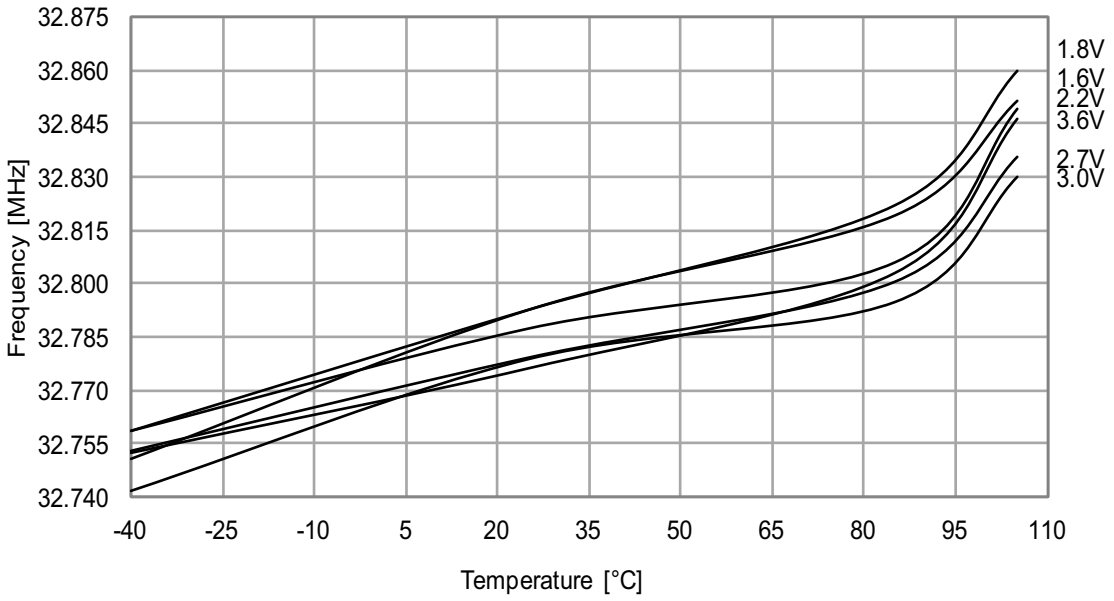


Figure 37-64. 32.768kHz Internal Oscillator Calibration Step Size

T = -40°C to 85°C, V_{CC} = 3V.

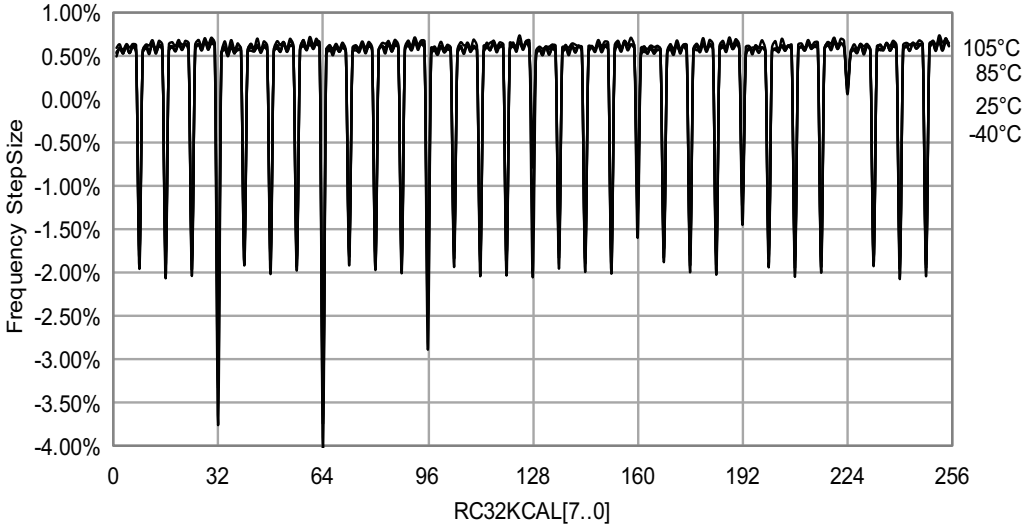
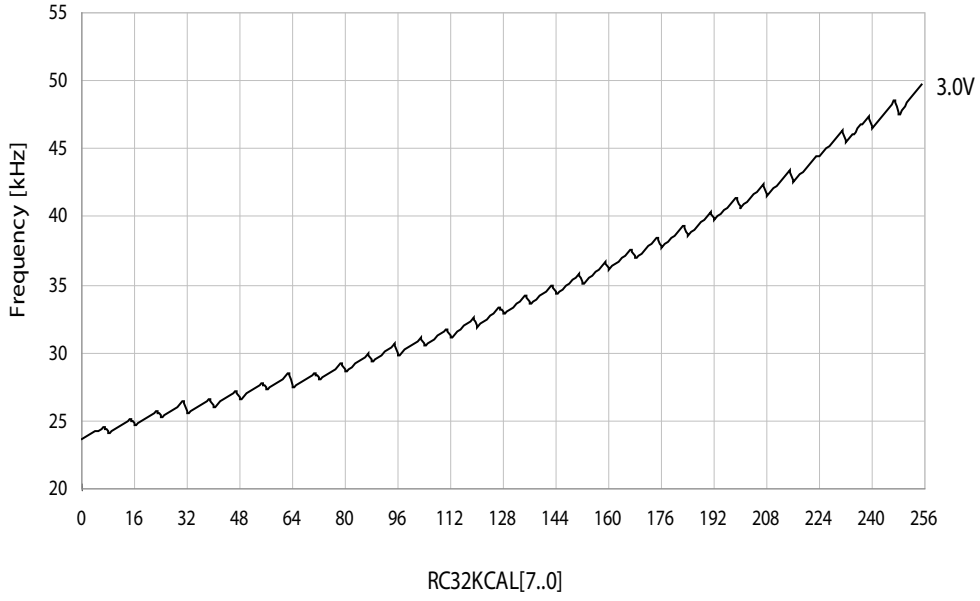


Figure 37-65. 32.768kHz Internal Oscillator Frequency vs. Calibration Value

V_{CC} = 3.0V, T = 25°C.



Problem fix/Workaround

The TxD pin direction can be set to input using the Port DIR register. Be advised that setting the Port DIR register to input will be immediate. Ongoing transmissions will be truncated.

39. Datasheet Revision History

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

39.1 8330H – 12/2014

1.	Corrected description for VFBGA pinout in Table 2-1 on page 5.

39.2 8330G – 11/2014

1.	Information for 105°C are added.
2.	Several small changes are made to set up the document according to the new template.
3.	AVDD in Table 32-2 on page 56 has been corrected to AVCC.
4.	Changed Vcc to AVcc in Section 29. “ADC – 12-bit Analog to Digital Converter” on page 49 and Section 30. “AC – Analog Comparator” on page 51.
5.	Added error information on USART transmitter to Section 38. “Errata” on page 136.
6.	Changed pin names in Table 2-1 on page 5 from VCC1, LVCC1 and LVCC01 to VCC from GND1, LGND1 and LGND01 to GND from D2W_D to PDI_DATA
7.	Changed unit value for parameter $t_{SU;DAT}$ to ns in Table 36-30 on page 89.
8.	Corrected description for VFBGA pinout. Pin F2 changed from VCC to GND in Table 2-1 on page 5.

39.3 8330F – 02/2014

1.	Added the correct VFBGA package drawing: ● Replaced the package 100C1 (CBGA) by the package 7A1 (VFBGA) ● Updated the package type in “Ordering Information” on page 2 ● Updated the title in Figure 2-2 on page 5 and in the Table 2-1 on page 5 ● Updated the package type in “Typical Characteristics” on page 91
2.	Updated the title name of the Table 36-24 on page 82.

39.4 8330E – 06/2013

1.	Added pinout for CBGA package option: Figure 2-2 and Table 2-1 on page 5.