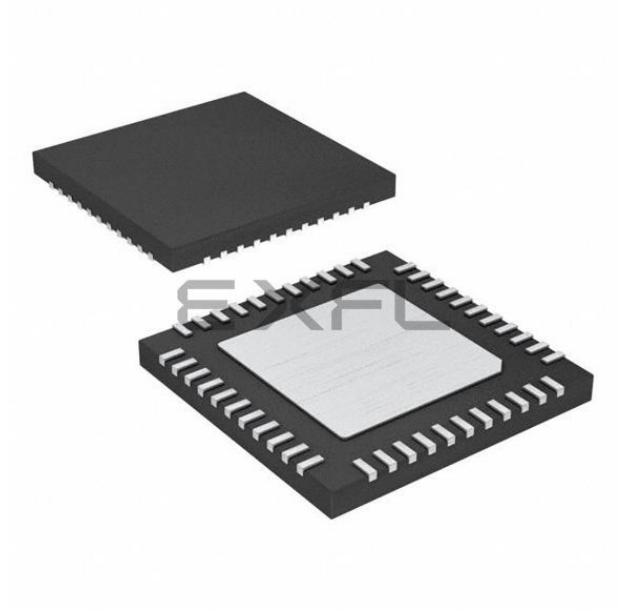


Welcome to E-XFL.COM

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-Bit
Speed	20MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	32
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)	1.8V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VFQFN Exposed Pad
Supplier Device Package	44-VQFN (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega644a-mur

timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the Crystal/Resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

Atmel offers the QTouch® library for embedding capacitive touch buttons, sliders and wheels functionality into AVR microcontrollers. The patented charge-transfer signal acquisition offers robust sensing and includes fully debounced reporting of touch keys and includes Adjacent Key Suppression® (AKS™) technology for unambiguous detection of key events. The easy-to-use QTouch Suite toolchain allows you to explore, develop and debug your own touch applications.

The device is manufactured using Atmel's high-density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega164A/164PA/324A/324PA/644A/644PA/1284/1284P is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega164A/164PA/324A/324PA/644A/644PA/1284/1284P is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

2.2 Comparison between ATmega164A, ATmega164PA, ATmega324A, ATmega324PA, ATmega644A, ATmega644PA, ATmega1284 and ATmega1284P

Table 2-1. Differences between ATmega164A, ATmega164PA, ATmega324A, ATmega324PA, ATmega644A, ATmega644PA, ATmega1284 and ATmega1284P.

Device	Flash	EEPROM	RAM	Units
ATmega164A	16K	512	1K	bytes
ATmega164PA	16K	512	1K	
ATmega324A	32K	1K	2K	
ATmega324PA	32K	1K	2K	
ATmega644A	64K	2K	4K	
ATmega644PA	64K	2K	4K	
ATmega1284	128K	4K	16K	
ATmega1284P	128K	4K	16K	

2.3 Pin Descriptions11

2.3.1 VC

Digital supply voltage.

2.3.2 GND

Ground.

Assembly Code Example [1][2]

```
USART_Transmit:
        ; Wait for empty transmit buffer
        sbis                UCSRnA,UDREn
        rjmp                USART_Transmit
        ; Copy 9th bit from r17 to TXB8
        cbi                 UCSRnB,TXB8
        sbrc                r17,0
        sbi                 UCSRnB,TXB8
        ; Put LSB data (r16) into buffer, sends the data
        out                 UDRn,r16
        ret
```

C Code Example [1][2]

```
void USART_Transmit( unsigned int data )
{
        /* Wait for empty transmit buffer */
        while ( !( UCSRnA & (1<<UDREn)) )
                        ;
        /* Copy 9th bit to TXB8 */
        UCSRnB &= ~(1<<TXB8);
        if ( data & 0x0100 )
                UCSRnB |= (1<<TXB8);
        /* Put data into buffer, sends the data */
        UDRn = data;
}
```

Notes: 1. These transmit functions are written to be general functions. They can be optimized if the contents of the UCSRnB is static. For example, only the TXB8 bit of the UCSRnB Register is used after initialization.

2. See “About code examples” on page 9.

The ninth bit can be used for indicating an address frame when using multi processor communication mode or for other protocol handling as for example synchronization.

19.7.3 Transmitter Flags and Interrupts

The USART Transmitter has two flags that indicate its state: USART Data Register Empty (UDREn) and Transmit Complete (TXCn). Both flags can be used for generating interrupts.

The Data Register Empty (UDREn) Flag indicates whether the transmit buffer is ready to receive new data. This bit is set when the transmit buffer is empty, and cleared when the transmit buffer contains data to be transmitted that has not yet been moved into the Shift Register. For compatibility with future devices, always write this bit to zero when writing the UCSRnA Register.

When the Data Register Empty Interrupt Enable (UDRIEn) bit in UCSRnB is written to one, the USART Data Register Empty Interrupt will be executed as long as UDREn is set (provided that global interrupts are enabled). UDREn is cleared by writing UDRn. When interrupt-driven data transmission is used, the Data Register Empty interrupt routine must either write new data to UDRn in order to clear UDREn or disable the Data Register Empty interrupt, otherwise a new interrupt will occur once the interrupt routine terminates.

The Transmit Complete (TXCn) Flag bit is set one when the entire frame in the Transmit Shift Register has been shifted out and there are no new data currently present in the transmit buffer. The TXCn Flag bit is automatically cleared when a transmit complete interrupt is executed, or it can be cleared by writing a one to its bit location. The TXCn Flag is useful in half-duplex communication interfaces (like the RS-485 standard), where a

Table 19-5. UPMn bits settings.

UPMn1	UPMn0	Parity Mode
0	0	Disabled
0	1	Reserved
1	0	Enabled, Even Parity
1	1	Enabled, Odd Parity

- **Bit 3 – USBSn: Stop Bit Select**

This bit selects the number of stop bits to be inserted by the Transmitter. The Receiver ignores this setting.

Table 19-6. USBS bit settings.

USBSn	Stop bit(s)
0	1-bit
1	2-bit

- **Bit 2:1 – UCSZn1:0: Character Size**

The UCSZn1:0 bits combined with the UCSZn2 bit in UCSRnB sets the number of data bits (Character SiZe) in a frame the Receiver and Transmitter use.

Table 19-7. UCSZn bits settings.

UCSZn2	UCSZn1	UCSZn0	Character Size
0	0	0	5-bit
0	0	1	6-bit
0	1	0	7-bit
0	1	1	8-bit
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	9-bit

Assembly Code Example [1]

```
USART_Init:
        clr r18
        out UBRRnH,r18
        out UBRRnL,r18
        ; Setting the XCKn port pin as output, enables master
mode.
        sbi XCKn_DDR, XCKn
        ; Set MSPI mode of operation and SPI data mode 0.
        ldi r18,
(1<<UMSELn1)|(1<<UMSELn0)|(0<<UCPHAn)|(0<<UCPOLn)
        out UCSRnC,r18
        ; Enable receiver and transmitter.
        ldi r18, (1<<RXENn)|(1<<TXENn)
        out UCSRnB,r18
        ; Set baud rate.
        ; IMPORTANT: The Baud Rate must be set after the
transmitter is enabled!
        out UBRRnH, r17
        out UBRRnL, r18
        ret
```

C Code Example[1]

```
void USART_Init( unsigned int baud )
{
        UBRRn = 0;
        /* Setting the XCKn port pin as output, enables master
mode. */
        XCKn_DDR |= (1<<XCKn);
        /* Set MSPI mode of operation and SPI data mode 0. */
        UCSRnC =
(1<<UMSELn1)|(1<<UMSELn0)|(0<<UCPHAn)|(0<<UCPOLn);
        /* Enable receiver and transmitter. */
        UCSRnB = (1<<RXENn)|(1<<TXENn);
        /* Set baud rate. */
        /* IMPORTANT: The Baud Rate must be set after the
transmitter is enabled */
        UBRRn = baud;
}
```

Note: 1. See “About code examples” on page 9.

20.6 Data Transfer

Using the USART in MSPI mode requires the Transmitter to be enabled, that is, the TXENn bit in the UCSRnB register is set to one. When the Transmitter is enabled, the normal port operation of the TxDn pin is overridden and given the function as the Transmitter's serial output. Enabling the receiver is optional and is done by setting the RXENn bit in the UCSRnB register to one. When the receiver is enabled, the normal pin operation of the RxDn pin is overridden and given the function as the Receiver's serial input. The XCKn will in both cases be used as the transfer clock.

After initialization the USART is ready for doing data transfers. A data transfer is initiated by writing to the UDRn I/O location. This is the case for both sending and receiving data since the transmitter controls the transfer

The upper seven bits are the address to which the two-wire Serial Interface will respond when addressed by a Master. If the LSB is set, the TWI will respond to the general call address (0x00), otherwise it will ignore the general call address.

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
value	0	1	0	0	0	1	0	X

TWEN must be written to one to enable the TWI. The TWEA bit must be written to one to enable the acknowledgement of the device's own slave address or the general call address. TWSTA and TWSTO must be written to zero.

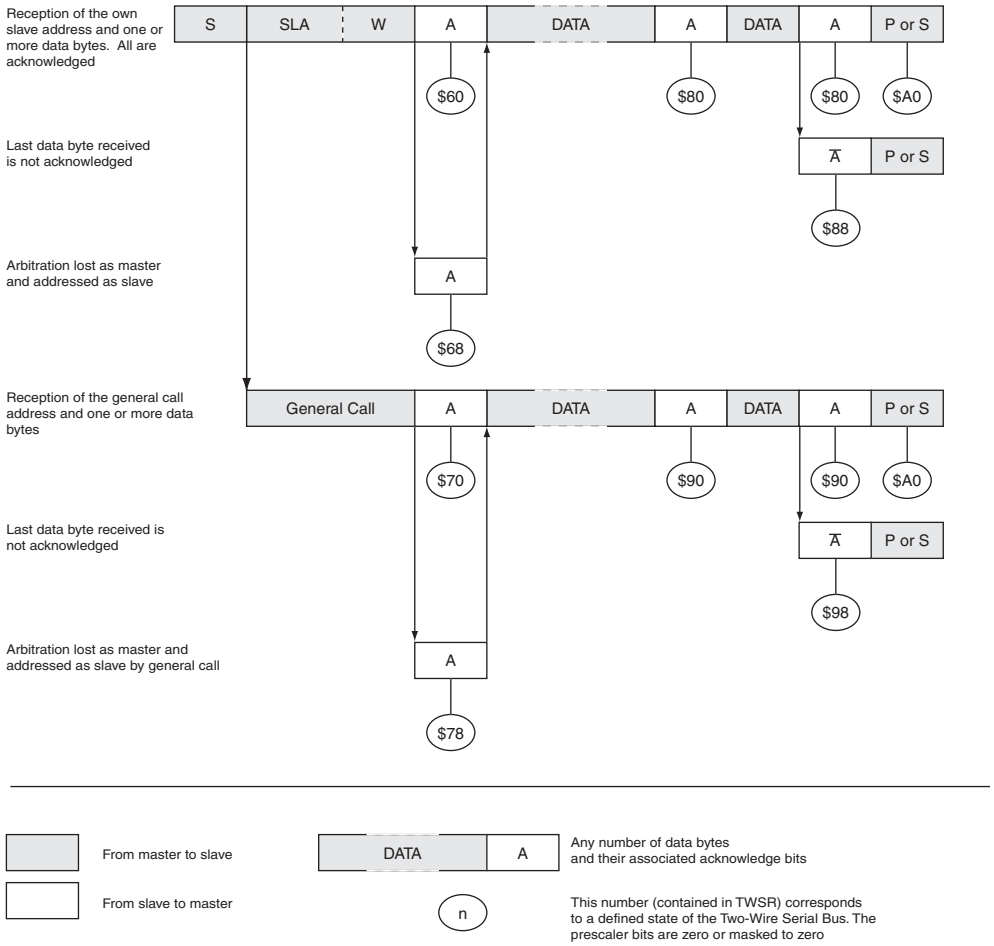
When TWAR and TWCR have been initialized, the TWI waits until it is addressed by its own slave address (or the general call address if enabled) followed by the data direction bit. If the direction bit is "0" (write), the TWI will operate in SR mode, otherwise ST mode is entered. After its own slave address and the write bit have been received, the TWINT Flag is set and a valid status code can be read from TWSR. The status code is used to determine the appropriate software action. The appropriate action to be taken for each status code is detailed in Table 21-5 on page 222. The Slave Receiver mode may also be entered if arbitration is lost while the TWI is in the Master mode (see states 0x68 and 0x78).

If the TWEA bit is reset during a transfer, the TWI will return a "Not Acknowledge" ("1") to SDA after the next received data byte. This can be used to indicate that the Slave is not able to receive any more bytes. While TWEA is zero, the TWI does not acknowledge its own slave address. However, the two-wire Serial Bus is still monitored and address recognition may resume at any time by setting TWEA. This implies that the TWEA bit may be used to temporarily isolate the TWI from the two-wire Serial Bus.

In all sleep modes other than Idle mode, the clock system to the TWI is turned off. If the TWEA bit is set, the interface can still acknowledge its own slave address or the general call address by using the two-wire Serial Bus clock as a clock source. The part will then wake up from sleep and the TWI will hold the SCL clock low during the wake up and until the TWINT Flag is cleared (by writing it to one). Further data reception will be carried out as normal, with the AVR clocks running as normal. Observe that if the AVR is set up with a long start-up time, the SCL line may be held low for a long time, blocking other data transmissions.

Note that the two-wire Serial Interface Data Register – TWDR does not reflect the last byte present on the bus when waking up from these Sleep modes.

Figure 21-16. Formats and states in the Slave Receiver mode.



21.7.4 Slave Transmitter mode

In the Slave Transmitter mode, a number of data bytes are transmitted to a Master Receiver (see Figure 21-17). All the status codes mentioned in this section assume that the prescaler bits are zero or are masked to zero.

Figure 21-17. Data transfer in Slave Transmitter mode.

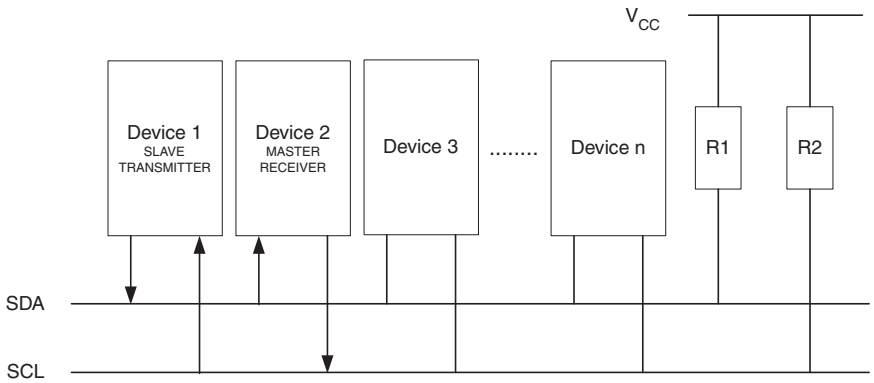
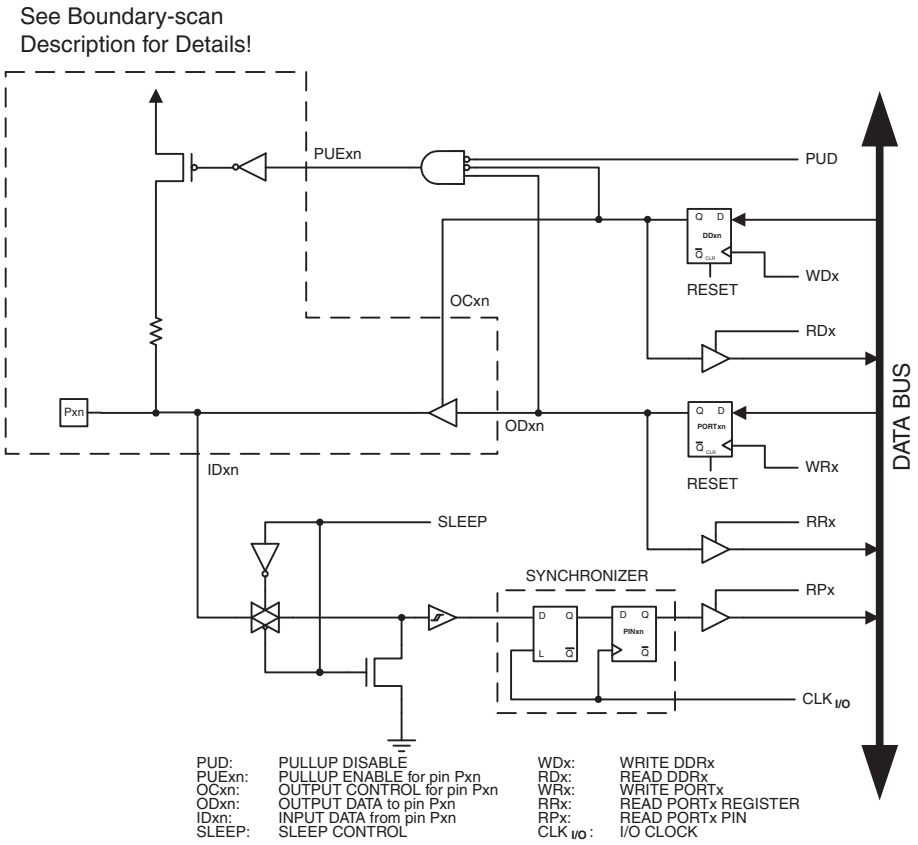


Figure 25-4. General Port Pin schematic diagram.



25.5.2 Scanning the RESET Pin

The RESET pin accepts 5V active low logic for standard reset operation, and 12V active high logic for High Voltage Parallel programming. An observe-only cell as shown in Figure 25-5 is inserted for the 5V reset signal.

Figure 25-5. Observe-only cell.

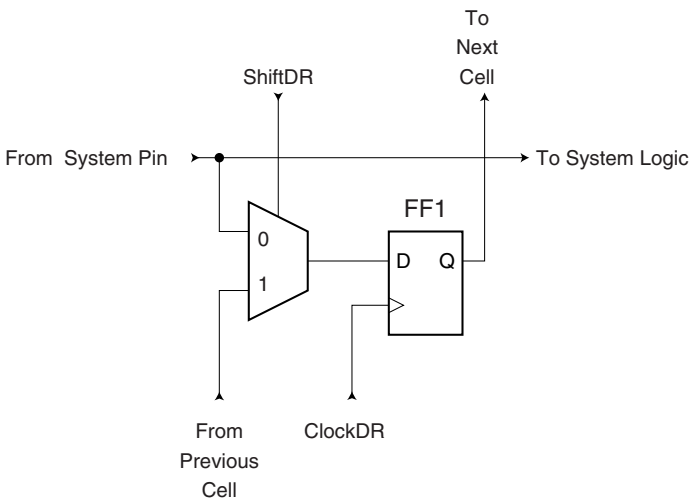


Table 27-9. Pin name mapping.

Signal name in programming mode	Pin name	I/O	Function
RDY/$\overline{BSY}$	PD1	O	0: Device is busy programming, 1: Device is ready for new command.
$\overline{OE}$	PD2	I	Output Enable (Active low).
$\overline{WR}$	PD3	I	Write Pulse (Active low).
BS1	PD4	I	Byte Select 1.
XA0	PD5	I	XTAL Action Bit 0
XA1	PD6	I	XTAL Action Bit 1
PAGEL	PD7	I	Program Memory and EEPROM data Page Load.
BS2	PA0	I	Byte Select 2.
DATA	PB7-0	I/O	Bi-directional Data bus (Output when $\overline{OE}$ is low).

Table 27-10. BS2 and BS1 encoding.

BS2	BS1	Flash / EEPROM address	Flash data loading / reading	Fuse programming	Reading fuse and lock bits
0	0	Low Byte	Low Byte	Low Byte	Fuse Low Byte
0	1	High Byte	High Byte	High Byte	Lockbits
1	0	Extended High Byte	Reserved	Extended Byte	Extended Fuse Byte
1	1	Reserved	Reserved	Reserved	Fuse High Byte

,

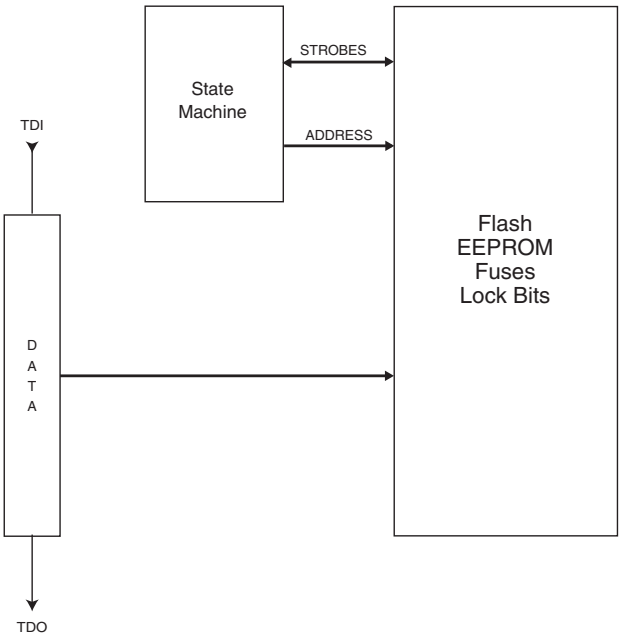
Table 27-11. Pin values used to enter Programming mode.

Pin	Symbol	Value
PAGEL	Prog_enable[3]	0
XA1	Prog_enable[2]	0
XA0	Prog_enable[1]	0
BS1	Prog_enable[0]	0

Table 27-12. XA1 and XA0 enoding.

XA1	XA0	Action when XTAL1 is pulsed
0	0	Load Flash or EEPROM Address (High or low address byte determined by BS2 and BS1)
0	1	Load Data (High or Low data byte for Flash determined by BS1)
1	0	Load Command
1	1	No Action, Idle

Figure 27-17. Flash data byte register.



The state machine controlling the Flash Data Byte Register is clocked by TCK. During normal operation in which eight bits are shifted for each Flash byte, the clock cycles needed to navigate through the TAP controller automatically feeds the state machine for the Flash Data Byte Register with sufficient number of clock pulses to complete its operation transparently for the user. However, if too few bits are shifted between each Update-DR state during page load, the TAP controller should stay in the Run-Test/Idle state for some TCK cycles to ensure that there are at least 11 TCK cycles between each Update-DR state.

27.10.12Programming Algorithm

All references below of type “1a”, “1b”, and so on, refer to Table 27-18 on page 310.

27.10.13Entering Programming Mode

1. Enter JTAG instruction AVR_RESET and shift 1 in the Reset Register.
2. Enter instruction PROG_ENABLE and shift 0b1010_0011_0111_0000 in the Programming Enable Register.

27.10.14Leaving Programming Mode

1. Enter JTAG instruction PROG_COMMANDS.
2. Disable all programming instructions by using no operation instruction 11a.
3. Enter instruction PROG_ENABLE and shift 0b0000_0000_0000_0000 in the programming Enable Register.
4. Enter JTAG instruction AVR_RESET and shift 0 in the Reset Register.

27.10.15Performing Chip Erase

1. Enter JTAG instruction PROG_COMMANDS.
2. Start Chip Erase using programming instruction 1a.
3. Poll for Chip Erase complete using programming instruction 1b, or wait for t_{WLRH_CE} (refer to Table 27-14 on page 300).

28.4 System and reset characteristics

Table 28-12. Reset, Brown-out and Internal Voltage Reference characteristics.

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
V_{POT}	Power-on Reset Threshold Voltage (rising)		1.1	1.4	1.6	V
	Power-on Reset Threshold Voltage (falling) (1)		0.6	1.3	1.6	
V_{RST}	$\overline{\text{RESET}}$ Pin Threshold Voltage		$0.2V_{CC}$		$0.9V_{CC}$	
t_{RST}	Minimum pulse width on $\overline{\text{RESET}}$ Pin				2.5	µs
V_{HYST}	Brown-out Detector Hysteresis			50		mV
t_{BOD}	Min Pulse Width on Brown-out Reset			2		µs
V_{BG}	Bandgap reference voltage	V_{CC}= 2.7V, T_A = 25°C	1.0	1.1	1.2	V
t_{BG}	Bandgap reference start-up time	V_{CC}= 2.7V, T_A = 25°C		40	70	µs
I_{BG}	Bandgap reference current consumption	V_{CC}= 2.7V, T_A = 25°C		10		µA

Notes: 1. The Power-on Reset will not work unless the supply voltage has been below V_{POT} (falling).

Table 28-13. BODLEVEL fuse coding (1).

BODLEVEL 2:0 Fuses	Min. V_{BOT}	Typ. V_{BOT}	Max. V_{BOT}	Units
111	BOD disabled			
110	1.7	1.8	2.0	V
101	2.5	2.7	2.9	
100	4.1	4.3	4.5	
011	Reserved			
010				
001				
000				

Note: 1. V_{BOT} may be below nominal minimum operating voltage for some devices. For devices where this is the case, the device is tested down to V_{CC} = V_{BOT} during the production test. This guarantees that a Brown-Out Reset will occur before V_{CC} drops to a voltage where correct operation of the microcontroller is no longer guaranteed. The test is performed using BODLEVEL = 101 and BODLEVEL = 110.

28.5 External interrupts characteristics

Table 28-14. Asynchronous external interrupt characteristics.

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
t_{INT}	Minimum pulse width for asynchronous external interrupt			50		ns

Table 28-18. ADC characteristics, differential channels.

Symbol	Parameter	Condition	Min. (1)	Typ. (1)	Max. (1)	Units
	Resolution	Gain = 1×		10		Bits
		Gain = 10×		10		
		Gain = 200×		7		
	Absolute Accuracy (Including INL, DNL Quantization Error and Offset Error)	Gain = 1× V_{CC} = 5V, V_{REF} = 4V ADC clock = 200kHz		19		LSB
		Gain = 10× V_{CC} = 5V, V_{REF} = 4V ADC clock = 200kHz		19		
		Gain = 200× V_{CC} = 5V, V_{REF} = 4V ADC clock = 200kHz		12		
	Integral Non-linearity (INL)	Gain = 1× V_{CC} = 5V, V_{REF} = 4V ADC clock = 200kHz		2		
		Gain = 10× V_{CC} = 5V, V_{REF} = 4V ADC clock = 200kHz		4		
		Gain = 200× V_{CC} = 5V, V_{REF} = 4V ADC clock = 200kHz		11		
	Differential Non-linearity (DNL)	Gain = 1× V_{CC} = 5V, V_{REF} = 4V ADC clock = 200kHz		1		
		Gain = 10× V_{CC} = 5V, V_{REF} = 4V ADC clock = 200kHz		1.5		
		Gain = 200× V_{CC} = 5V, V_{REF} = 4V ADC clock = 200kHz		11		

Figure 30-39. Atmel ATmega164A: Calibrated 8MHz RC oscillator vs. temperature.

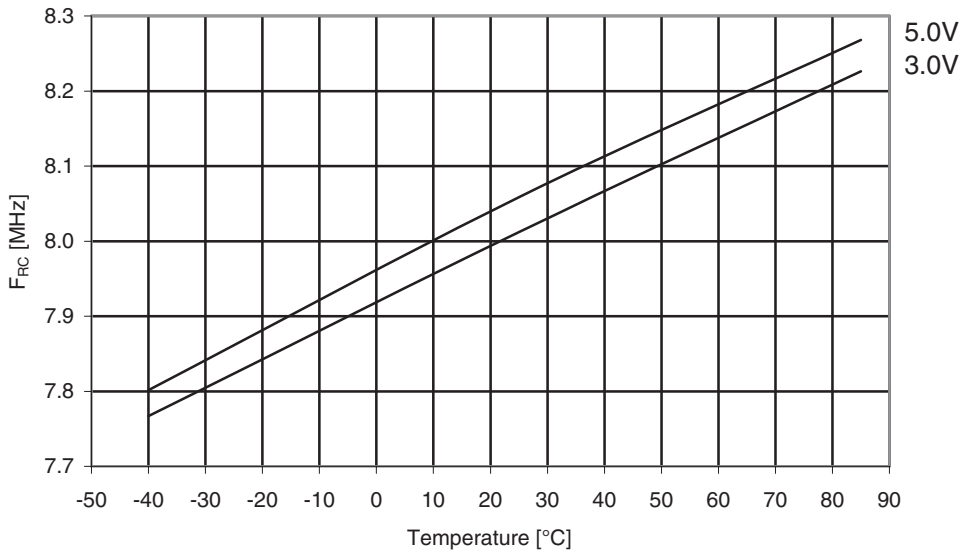


Figure 30-40. Atmel ATmega164A: Calibrated 8MHz RC oscillator vs. OSCCAL value.

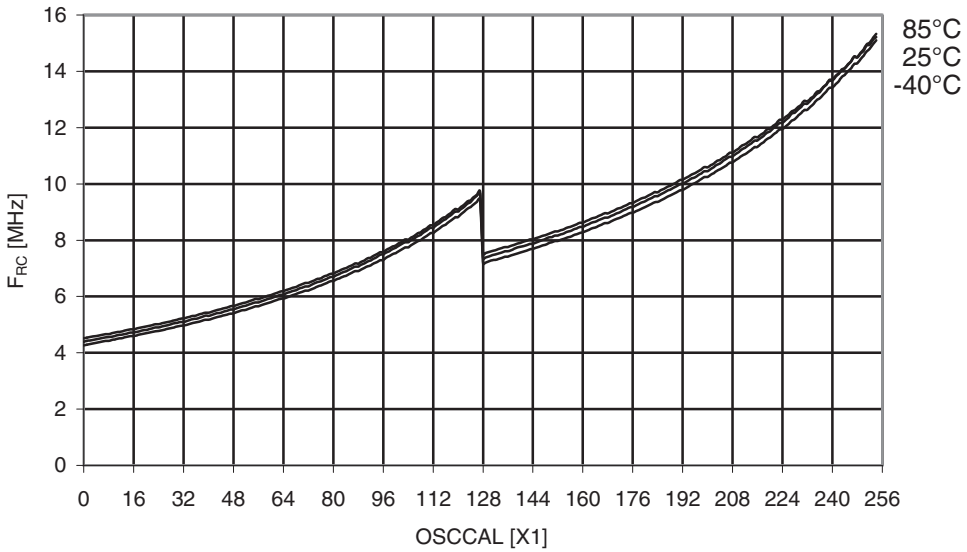


Figure 30-86. Atmel ATmega164PA: Watchdog oscillator frequency vs. V_{CC}.

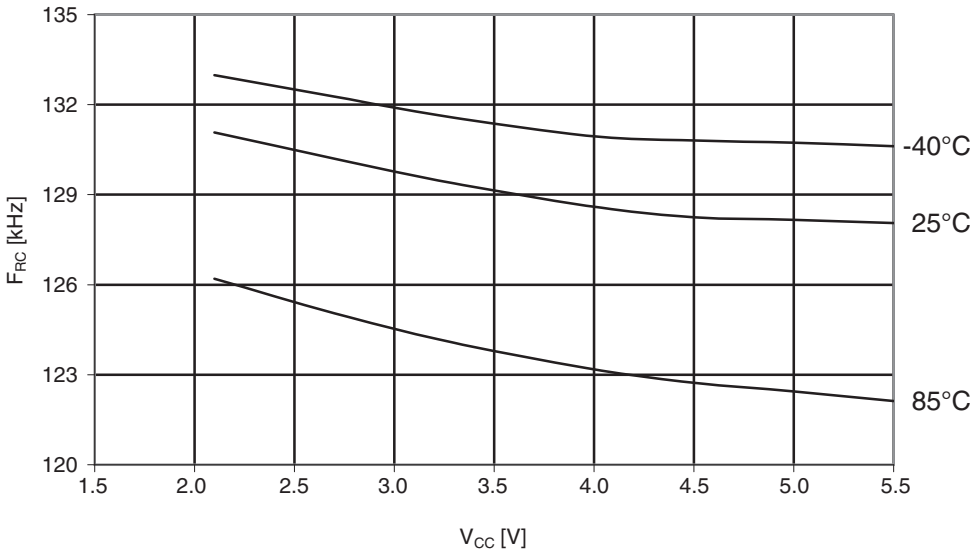


Figure 30-87. Atmel ATmega164PA: Calibrated 8MHz RC oscillator vs. V_{CC}.

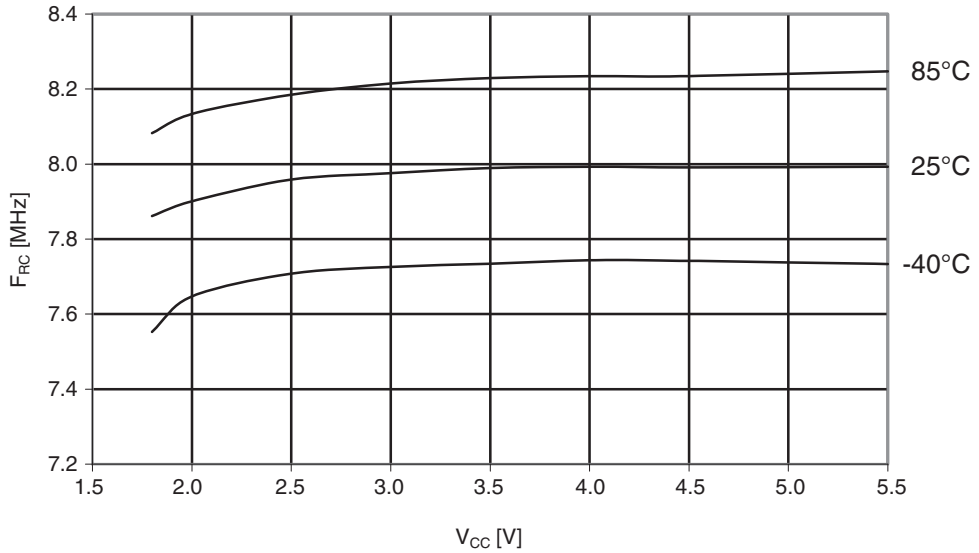


Figure 30-137. Atmel ATmega324A: Calibrated 8MHz RC oscillator vs. temperature.

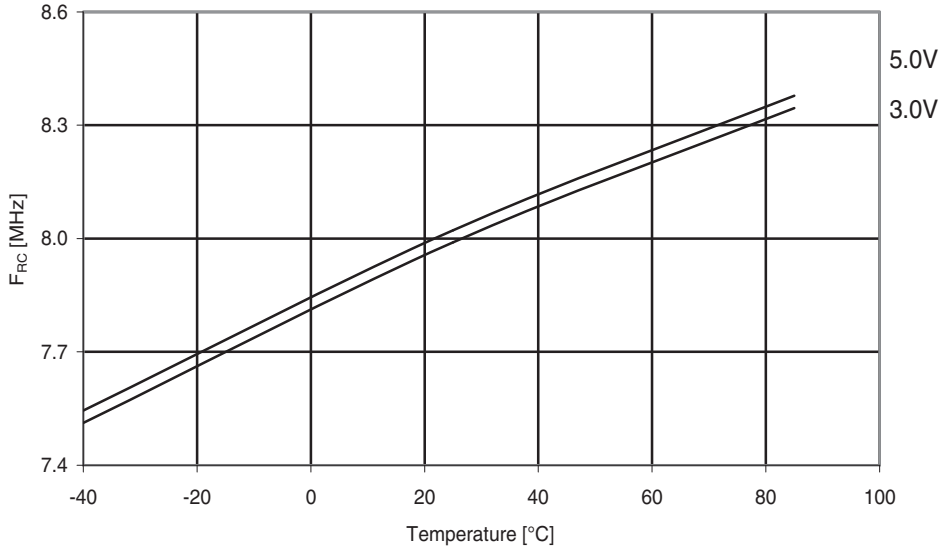


Figure 30-138. Atmel ATmega324A: Calibrated 8MHz RC oscillator vs. OSCCAL value.

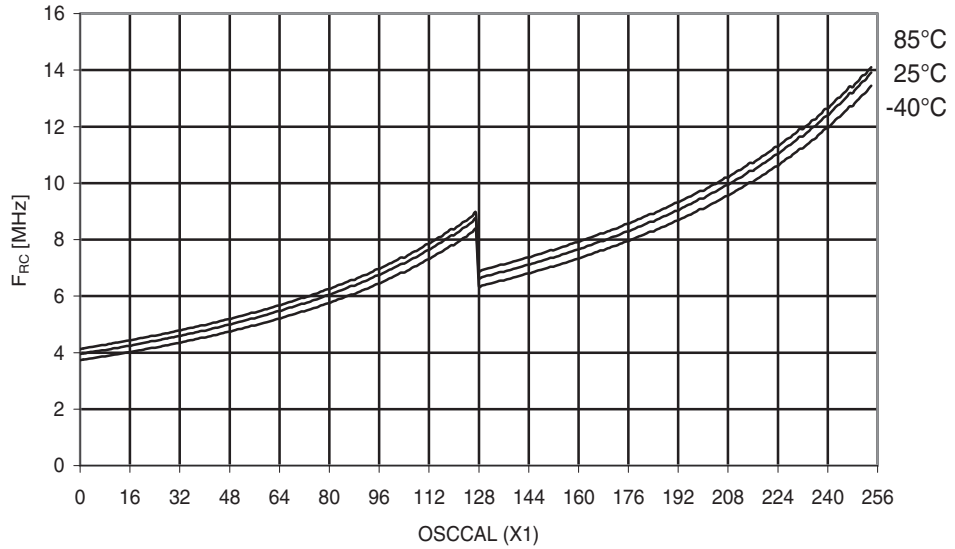


Figure 30-274. Atmel ATmega644PA: Reset pin input threshold vs. V_{CC} (V_{IL}, I/O pin read as '0').

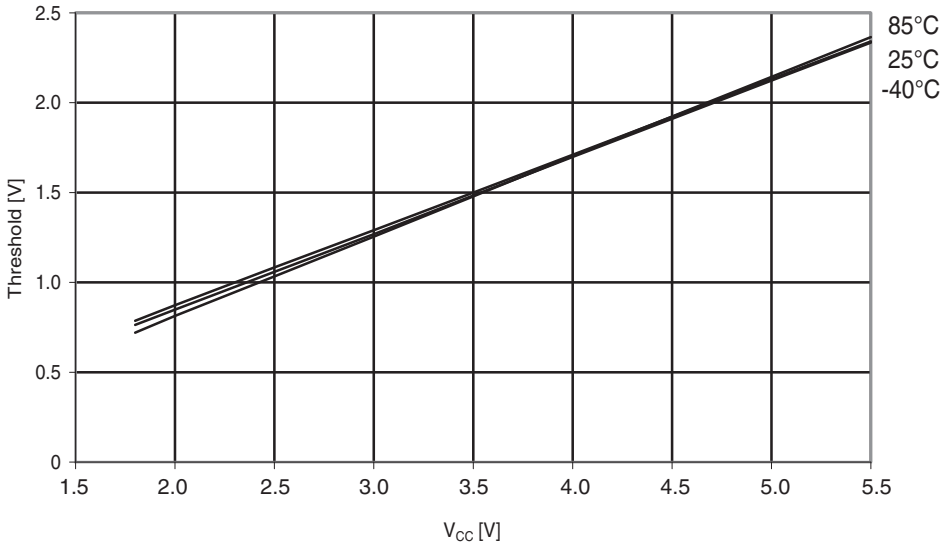
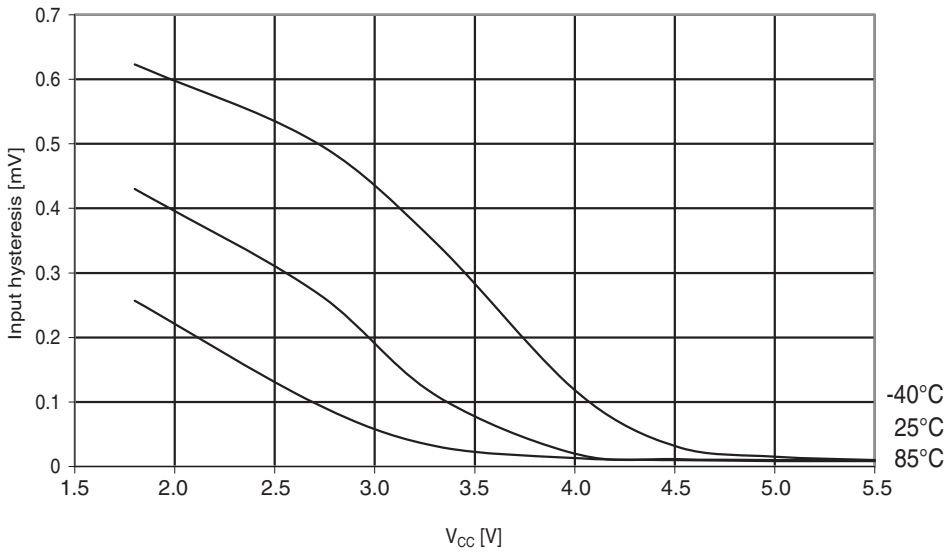


Figure 30-275. Atmel ATmega644PA: Reset pin input hysteresis vs. V_{CC}.



31. Typical Characteristics - TA = -40°C to 105°C

The following charts show typical behavior. These figures are not tested during manufacturing. All current consumption measurements are performed with all I/O pins configured as inputs and with internal pull-ups enabled. A sine wave generator with rail-to-rail output is used as clock source.

All Active- and Idle current consumption measurements are done with all bits in the PRR registers set and thus, the corresponding I/O modules are turned off. Also the Analog Comparator is disabled during these measurements. The power consumption in Power-down mode is independent of clock selection.

The current consumption is a function of several factors such as: operating voltage, operating frequency, loading of I/O pins, switching rate of I/O pins, code executed and ambient temperature. The dominating factors are operating voltage and frequency.

The current drawn from capacitive loaded pins may be estimated (for one pin) as C_L*V_{CC}*f where C_L = load capacitance, V_{CC} = operating voltage and f = average switching frequency of I/O pin.

The parts are characterized at frequencies higher than test limits. Parts are not guaranteed to function properly at frequencies higher than the ordering code indicates.

The difference between current consumption in Power-down mode with Watchdog Timer enabled and Power-down mode with Watchdog Timer disabled represents the differential current drawn by the Watchdog Timer.

Figure 31-85. Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 1 MHz)

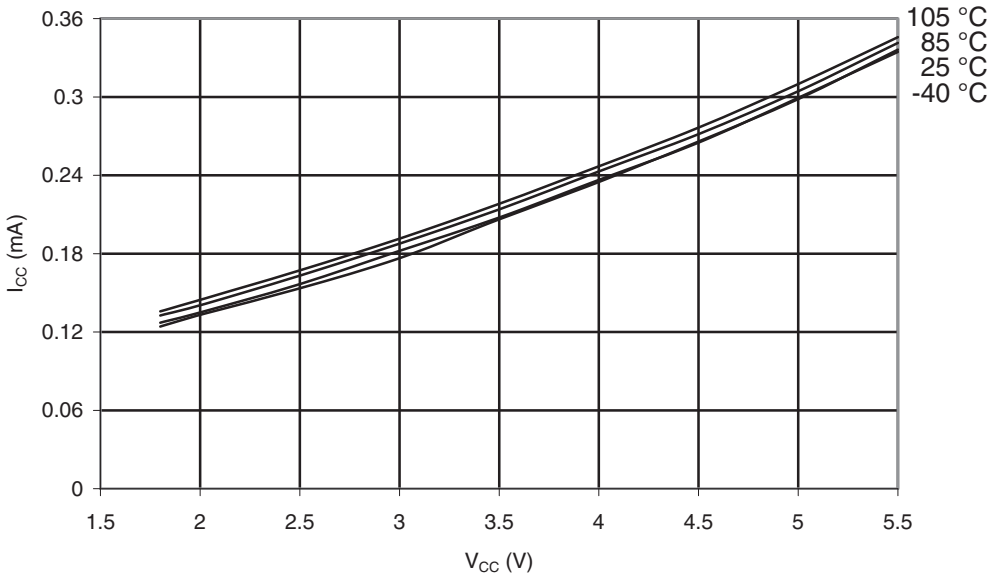
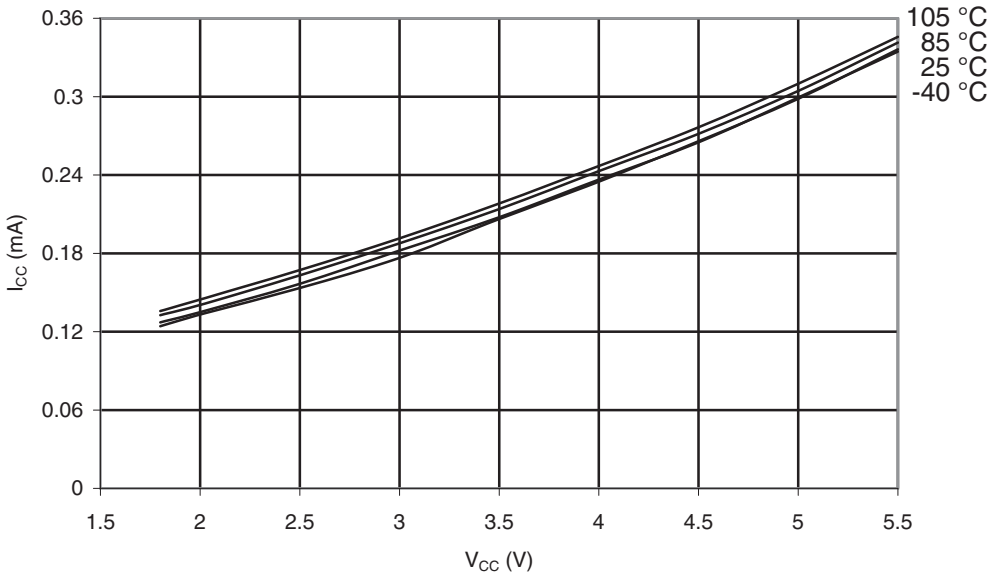


Figure 31-86. Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 128 kHz)



31.4.3 Power-down supply current

Figure 31-126. Atmel ATmega1284P: Power-down supply current vs. V_{CC} (watchdog timer disabled).

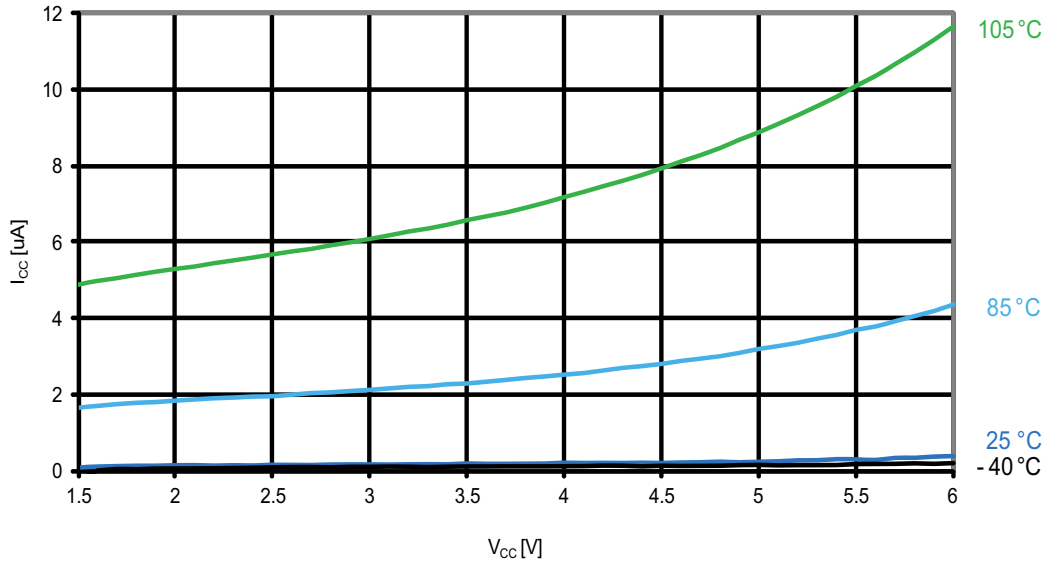


Figure 31-127. Atmel ATmega1284P: Power-down supply current vs. V_{CC} (watchdog timer enabled).

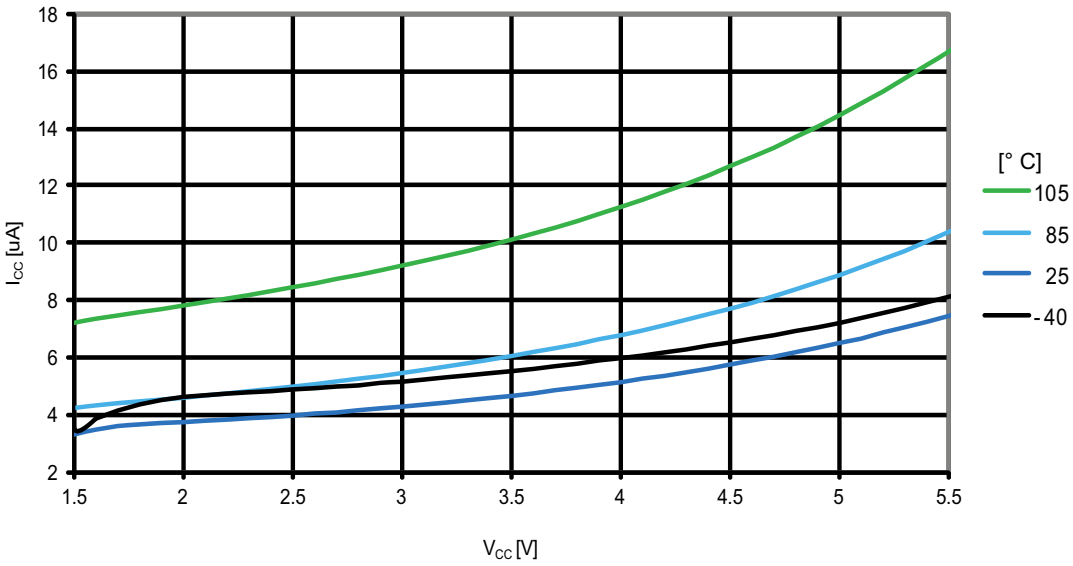
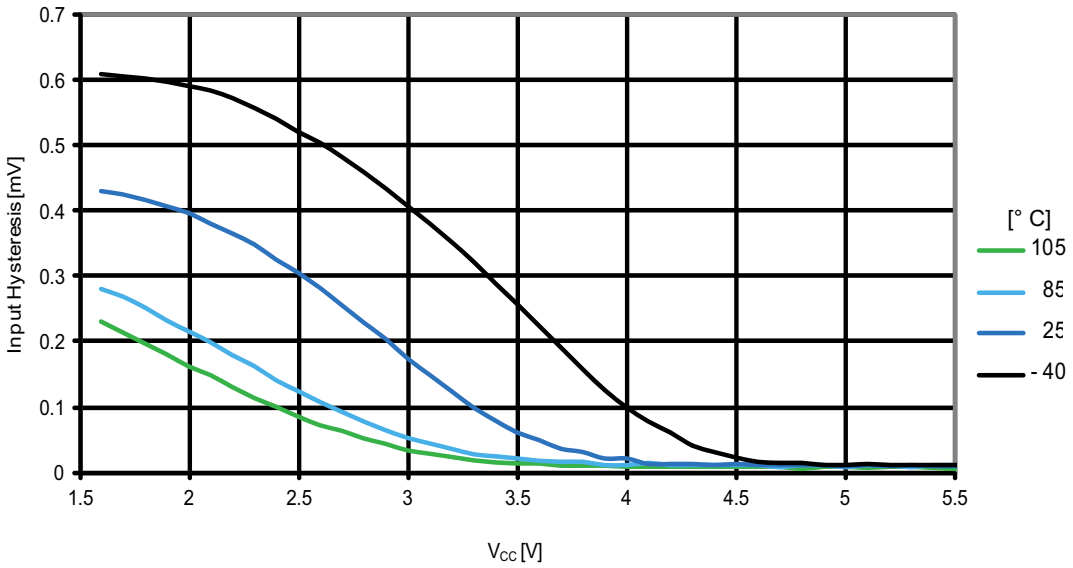
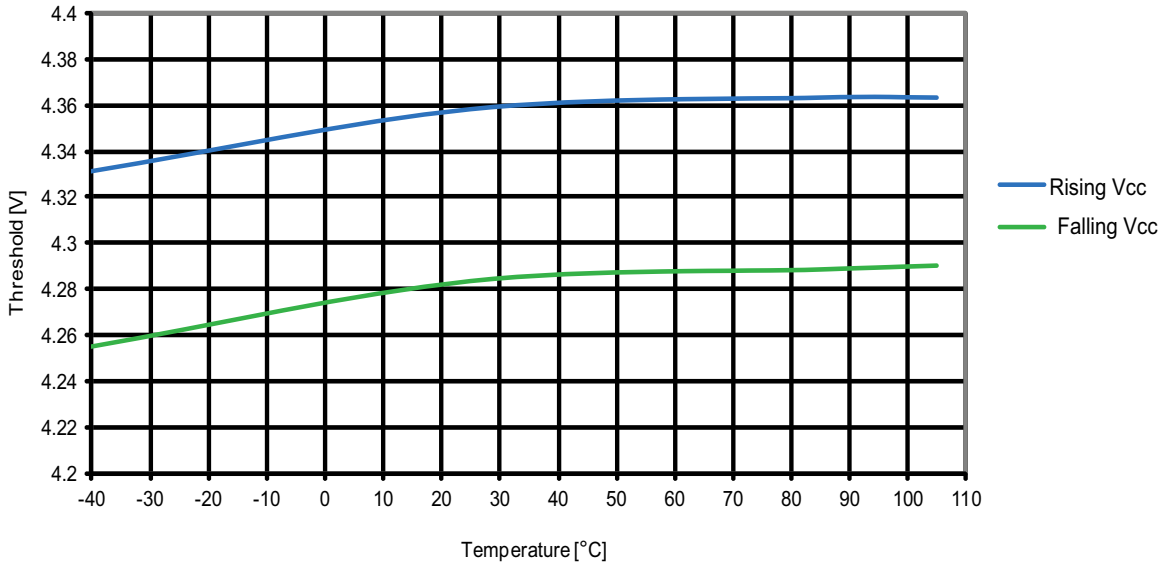


Figure 31-145. Atmel ATmega1284P: Reset pin input hysteresis vs. V_{CC}.

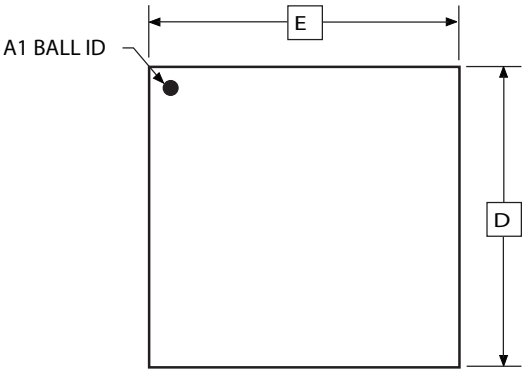


31.4.8 BOD threshold

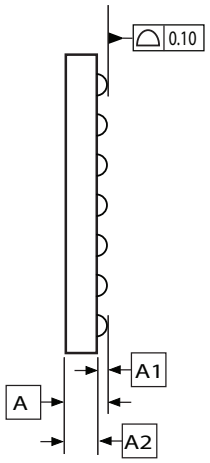
Figure 31-146. Atmel ATmega1284P: BOD threshold vs. temperature (V_{BOT} = 4.3V).



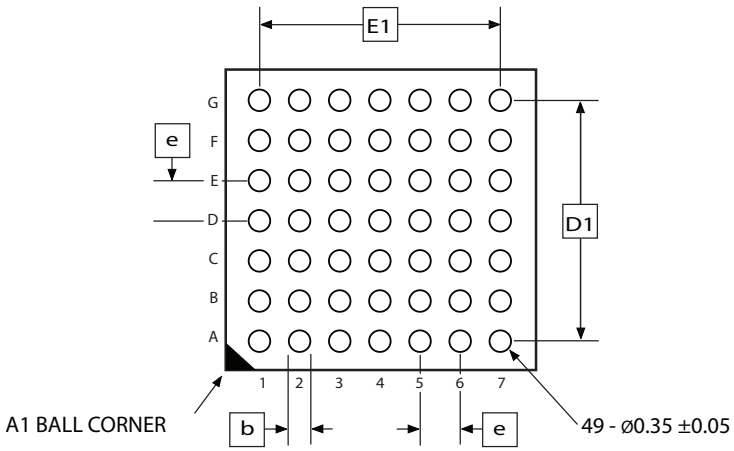
35.5 49C2



TOP VIEW



SIDE VIEW



BOTTOM VIEW

COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.00	
A1	0.20	–	–	
A2	0.65	–	–	
D	4.90	5.00	5.10	
D1		3.90 BSC		
E 4.90	5.00	5.10		
E1		3.90 BSC		
b	0.30	0.35	0.40	
e		0.65 BSC		

3/14/08

Atmel Package Drawing Contact: packagedrawings@atmel.com	TITLE 49C2, 49-ball (7 x 7 array), 0.65mm pitch, 5.0 x 5.0 x 1.0mm, very thin, fine-pitch ball grid array package (VFBGA)	GPC CBD	DRAWING NO. 49C2	REV. A