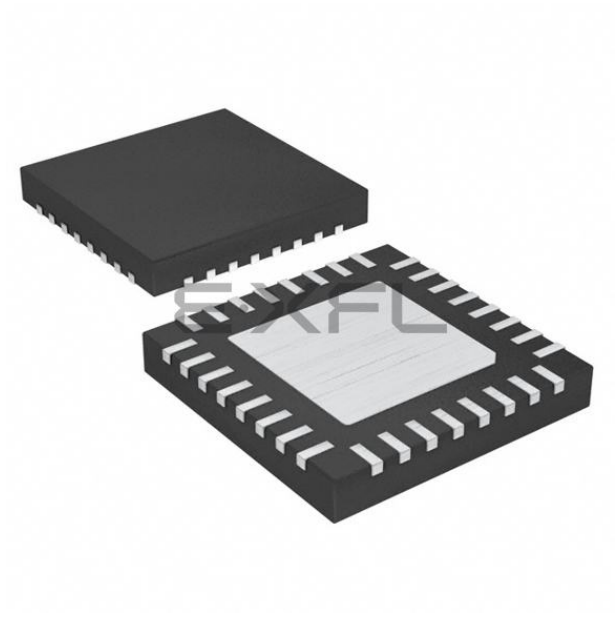


Welcome to E-XFL.COM

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	27
Program Memory Size	4KB (2K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 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	32-VFQFN Exposed Pad
Supplier Device Package	32-VFQFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega48pb-mu

12.3.1. Status Register

When addressing I/O Registers as data space using LD and ST instructions, the provided offset must be used. When using the I/O specific commands IN and OUT, the offset is reduced by 0x20, resulting in an I/O address offset within 0x00 - 0x3F.

Name: SREG
Offset: 0x5F
Reset: 0x00
Property: When addressing as I/O Register: address offset is 0x3F

Bit	7	6	5	4	3	2	1	0
	I	T	H	S	V	N	Z	C
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – I: Global Interrupt Enable
The Global Interrupt Enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the Global Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

Bit 6 – T: Copy Storage
The Bit Copy instructions BLD (Bit LoaD) and BST (Bit STore) use the T-bit as source or destination for the operated bit. A bit from a register in the Register File can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the Register File by the BLD instruction.

Bit 5 – H: Half Carry Flag
The Half Carry Flag H indicates a Half Carry in some arithmetic operations. Half Carry Flag is useful in BCD arithmetic. See the *Instruction Set Description* for detailed information.

Bit 4 – S: Sign Flag, S = N ⊕ V

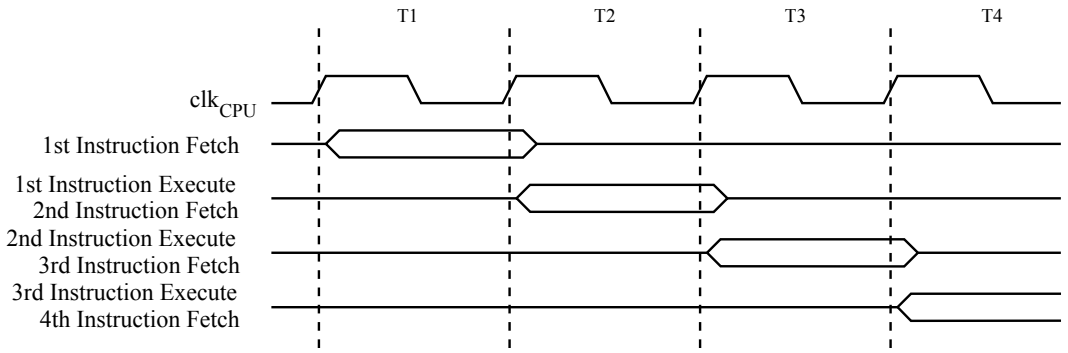
The S-bit is always an exclusive or between the Negative Flag N and the Two's Complement Overflow Flag V. See the *Instruction Set Description* for detailed information.

Bit 3 – V: Two's Complement Overflow Flag
The Two's Complement Overflow Flag V supports two's complement arithmetic. See the *Instruction Set Description* for detailed information.

Bit 2 – N: Negative Flag
The Negative Flag N indicates a negative result in an arithmetic or logic operation. See the *Instruction Set Description* for detailed information.

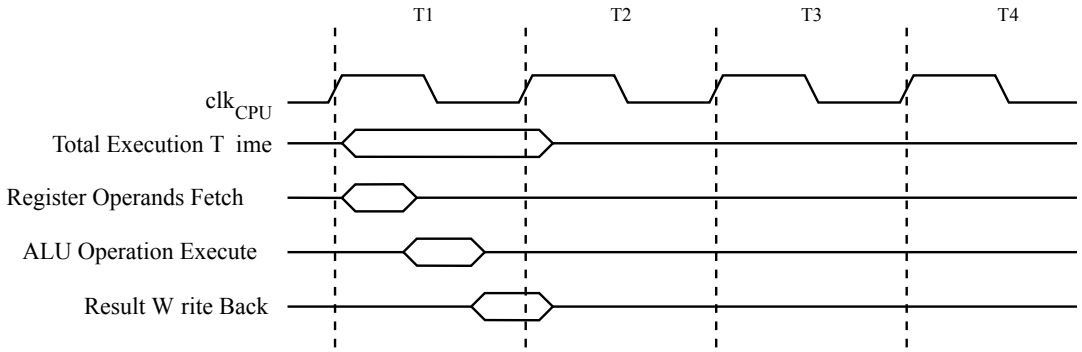
Bit 1 – Z: Zero Flag
The Zero Flag Z indicates a zero result in an arithmetic or logic operation. See the *Instruction Set Description* for detailed information.

Figure 12-4. The Parallel Instruction Fetches and Instruction Executions



The following Figure shows the internal timing concept for the Register File. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

Figure 12-5. Single Cycle ALU Operation



12.7. Reset and Interrupt Handling

The AVR provides several different interrupt sources. These interrupts and the separate Reset Vector each have a separate program vector in the program memory space. All interrupts are assigned individual enable bits which must be written logic one together with the Global Interrupt Enable bit in the Status Register in order to enable the interrupt. Depending on the Program Counter value, interrupts may be automatically disabled when Boot Lock bits BLB02 or BLB12 are programmed. This feature improves software security. See the section on MEMPROG- Memory Programming for details.

The lowest addresses in the program memory space are by default defined as the Reset and Interrupt Vectors. The complete list of vectors is shown in Interrupts. The lower the address the higher is the priority level. RESET has the highest priority, and next is INT0 – the External Interrupt Request 0. The Interrupt Vectors can be moved to the start of the Boot Flash section by setting the IVSEL bit in the MCU Control Register (MCUCR). Refer to Interrupts for more information. The Reset Vector can also be moved to the start of the Boot Flash section by programming the BOOTRST Fuse, see BTLDR - Boot Loader Support – Read-While-Write Self-Programming.

When an interrupt occurs, the Global Interrupt Enable I-bit is cleared and all interrupts are disabled. The user software can write logic one to the I-bit to enable nested interrupts. All enabled interrupts can then interrupt the current interrupt routine. The I-bit is automatically set when a Return from Interrupt instruction – RETI – is executed.

There are basically two types of interrupts. The first type is triggered by an event that sets the Interrupt Flag. For these interrupts, the Program Counter is vectored to the actual Interrupt Vector in order to execute the interrupt handling routine, and hardware clears the corresponding Interrupt Flag. Interrupt

Address	Labels		Code Comments
0x0033	RESET: ldi	r16, high(RAMEND)	; Main program start
0x0034	out	SPH,r16	; Set Stack Pointer to top of RAM
0x0035	ldi	r16, low(RAMEND)	
0x0036	out	SPL,r16	
0x0037	sei		; Enable interrupts
0x0038	<instr>	xxx	
...	...		...

When the BOOTRST Fuse is unprogrammed, the Boot section size set to 2Kbytes and the IVSEL bit in the MCUCR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega168PB is:

Address	Labels		Code Comments
0x0000	RESET: ldi	r16,high(RAMEND)	; Main program start
0x0001	out	SPH,r16	; Set Stack Pointer to top of RAM
0x0002	ldi	r16,low(RAMEND)	
0x0003	out	SPL,r16	
0x0004	sei		; Enable interrupts
0x0005	<instr>	xxx	
;			
.org	0x1C02		
0x1C02	jmp	EXT_INT0	; IRQ0 Handler
0x1C04	jmp	EXT_INT1	; IRQ1 Handler
...	...	...	
;			
0x1C32	jmp	SPM_RDY	; Store Program Memory Ready Handler

When the BOOTRST Fuse is programmed and the Boot section size set to 2Kbytes, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega168PB is:

Address	Labels		Code Comments
.org	0x0002		
0x0002	jmp	EXT_INT0	; IRQ0 Handler
0x0004	jmp	EXT_INT1	; IRQ1 Handler
...	...	...	

```
i = PINB;
...
```

19.2.5. Digital Input Enable and Sleep Modes

As shown in the figure of General Digital I/O, the digital input signal can be clamped to ground at the input of the Schmitt Trigger. The signal denoted SLEEP in the figure, is set by the MCU Sleep Controller in Power-down mode, Power-save mode, and Standby mode to avoid high power consumption if some input signals are left floating, or have an analog signal level close to $V_{CC}/2$.

SLEEP is overridden for port pins enabled as external interrupt pins. If the external interrupt request is not enabled, SLEEP is active also for these pins. SLEEP is also overridden by various other alternate functions as described in Alternate Port Functions.

If a logic high level is present on an asynchronous external interrupt pin configured as “Interrupt on Rising Edge, Falling Edge, or Any Logic Change on Pin” while the external interrupt is not enabled, the corresponding External Interrupt Flag will be set when resuming from the above mentioned Sleep mode, as the clamping in these sleep mode produces the requested logic change.

19.2.6. Unconnected Pins

If some pins are unused, it is recommended to ensure that these pins have a defined level. Even though most of the digital inputs are disabled in the deep sleep modes as described above, floating inputs should be avoided to reduce current consumption in all other modes where the digital inputs are enabled (Reset, Active mode and Idle mode).

The simplest method to ensure a defined level of an unused pin, is to enable the internal pull-up. In this case, the pull-up will be disabled during reset. If low power consumption during reset is important, it is recommended to use an external pull-up or pull-down. Connecting unused pins directly to V_{CC} or GND is not recommended, since this may cause excessive currents if the pin is accidentally configured as an output.

19.3. Alternate Port Functions

Most port pins have alternate functions in addition to being general digital I/Os. The following figure shows how the port pin control signals from the simplified Figure 19-2 General Digital I/O(1) can be overridden by alternate functions. The overriding signals may not be present in all port pins, but the figure serves as a generic description applicable to all port pins in the AVR microcontroller family.

Signal Name	PD3/OC2B/INT1/ PCINT19	PD2/INT0/ PCINT18	PD1/TXD0/ PCINT17	PD0/RXD0/ PCINT16
DI	PCINT19 INPUT / INT1 INPUT	PCINT18 INPUT / INT0 INPUT	PCINT17 INPUT	PCINT16 INPUT / RXD0
AIO	–	–	–	–

19.3.4. Alternate Functions of Port E

The Port E pins with alternate functions are shown in this table:

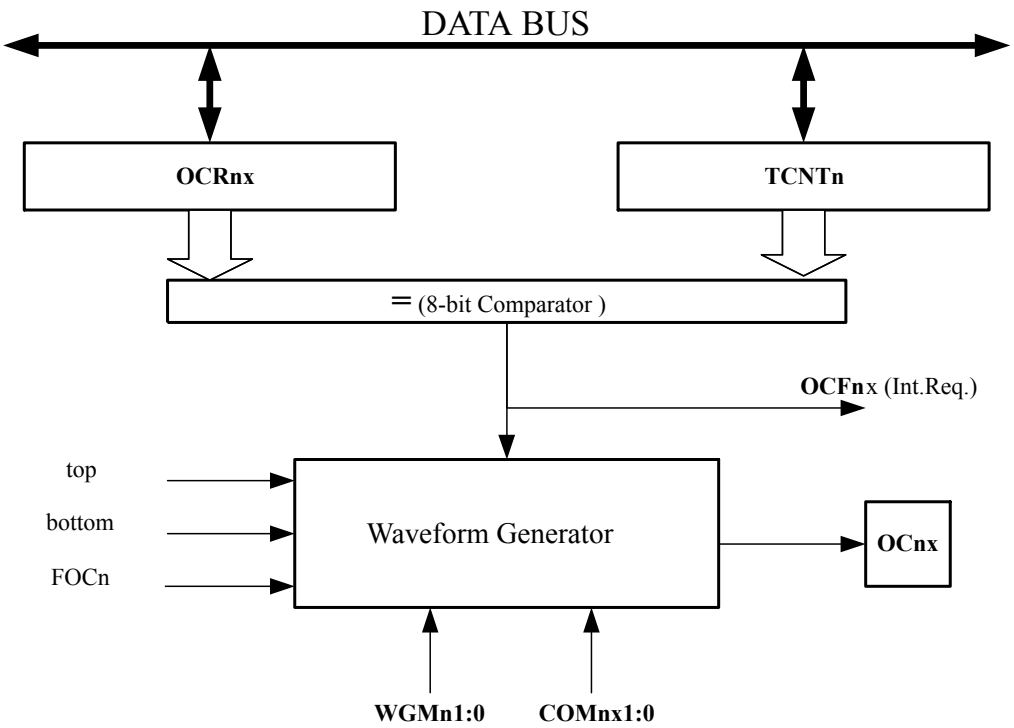
Table 19-12. Port E Pins Alternate Functions

Port Pin	Alternate Function
PE3	ADC7 (ADC Input Channel 7) T3 (Timer/Counter 3 External Counter Input) PCINT27
PE2	ADC6 (ADC Input Channel 6) ICP3 (Timer/Counter3 Input Capture Input) SS1 (SPI1 Bus Master Slave select) PCINT26
PE1	T4 (Timer/Counter 4 External Counter Input) SCL1 (2-wire Serial1 Bus Clock Line) PCINT25
PE0	ACO (AC Output Channel 0) ICP4 (Timer/Counter4 Input Capture Input) SDA1 (2-wire Serial1 Bus Data Input/Output Line) PCINT24

The alternate pin configuration is as follows:

- ADC7/T3/MOSI1/PCINT27– Port E, Bit 3
  - PE3 can also be used as ADC input Channel 7.
  - T3: Timer/Counter3 counter source.
  - MOSI1: SPI1 Master Data output, Slave Data input for SPI1 channel. When the SPI1 is enabled as a Slave, this pin is configured as an input regardless of the setting of DDE3. When the SPI1 is enabled as a Master, the data direction of this pin is controlled by DDE3. When the pin is forced by the SPI1 to be an input, the pull-up can still be controlled by the PORTE3 bit.
  - PCINT27: Pin Change Interrupt source 27. The PE3 pin can serve as an external interrupt source.
- ADC6/ICP3/SS1/PCINT26 – Port E, Bit 2
  - PE2 can also be used as ADC input Channel 6.
  - ICP3: Input Capture Pin. The PE2 pin can act as an Input Capture Pin for Timer/Counter3.

Figure 20-3. Output Compare Unit, Block Diagram



Note: The “n” in the register and bit names indicates the device number (n = 0 for Timer/Counter 0), and the “x” indicates Output Compare unit (A/B).

The OCR0x Registers are double buffered when using any of the Pulse Width Modulation (PWM) modes. When double buffering is enabled, the CPU has access to the OCR0x Buffer Register. The double buffering synchronizes the update of the OCR0x Compare Registers to either top or bottom of the counting sequence. The synchronization prevents the occurrence of odd-length, non-symmetrical PWM pulses, thereby making the output glitch-free.

The double buffering is disabled for the normal and Clear Timer on Compare (CTC) modes of operation, and the CPU will access the OCR0x directly.

Related Links

Modes of Operation on page 145

20.5.1. Force Output Compare

In non-PWM waveform generation modes, the match output of the comparator can be forced by writing a '1' to the Force Output Compare (FOCnx) bit. Forcing compare match will not set the OCFnx Flag or reload/clear the timer, but the OCnx pin will be updated as if a real compare match had occurred (the COMnx[1:0] bits define whether the OCnx pin is set, cleared or toggled).

20.5.2. Compare Match Blocking by TCNTn Write

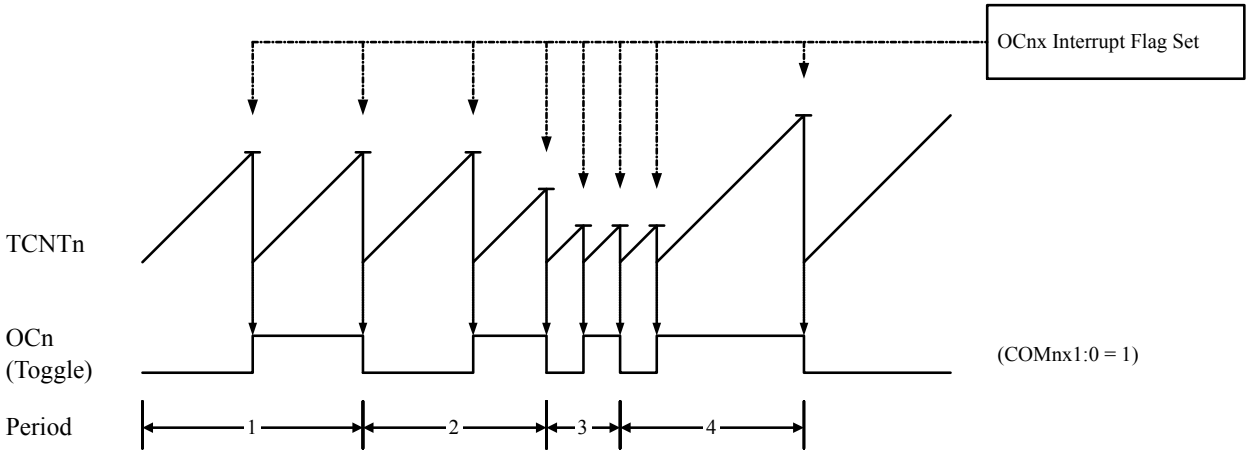
All CPU write operations to the TCNTn Register will block any compare match that occur in the next timer clock cycle, even when the timer is stopped. This feature allows OCRnx to be initialized to the same value as TCNTn without triggering an interrupt when the Timer/Counter clock is enabled.

20.5.3. Using the Output Compare Unit

Since writing TCNTn in any mode of operation will block all compare matches for one timer clock cycle, there are risks involved when changing TCNTn when using the Output Compare Unit, independently of whether the Timer/Counter is running or not. If the value written to TCNTn equals the OCRnx value, the

The timing diagram for the CTC mode is shown below. The counter value (TCNTn) increases until a compare match occurs between TCNTn and OCRnA, and then counter (TCNTn) is cleared.

Figure 20-5. CTC Mode, Timing Diagram



An interrupt can be generated each time the counter value reaches the TOP value by setting the OCFnA Flag. If the interrupt is enabled, the interrupt handler routine can be used for updating the TOP value.

Note: Changing TOP to a value close to BOTTOM while the counter is running must be done with care, since the CTC mode does not provide double buffering. If the new value written to OCRnA is lower than the current value of TCNTn, the counter will miss the compare match. The counter will then count to its maximum value (0xFF for a 8-bit counter, 0xFFFF for a 16-bit counter) and wrap around starting at 0x00 before the compare match will occur.

For generating a waveform output in CTC mode, the OCnA output can be set to toggle its logical level on each compare match by writing the two least significant Compare Output mode bits in the Timer/Counter Control Register A Control to toggle mode (TCCRnA.COMnA[1:0]=0x1). The OCnA value will only be visible on the port pin unless the data direction for the pin is set to output. The waveform generated will have a maximum frequency of $f_{OCn} = f_{clk\_I/O}/2$ when OCRnA is written to 0x00. The waveform frequency is defined by the following equation:

$$f_{OCnx} = \frac{f_{clk\_I/O}}{2 \cdot N \cdot (1 + OCRnx)}$$

N represents the prescaler factor (1, 8, 64, 256, or 1024).

As for the Normal mode of operation, the Timer/Counter Overflow Flag TOVn is set in the same clock cycle that the counter wraps from MAX to 0x00.

20.7.3. Fast PWM Mode

The Fast Pulse Width Modulation or Fast PWM modes (WGM0[2:0]=0x3 or WGM0[2:0]=0x7) provide a high frequency PWM waveform generation option. The Fast PWM modes differ from the other PWM options by their single-slope operation. The counter counts from BOTTOM to TOP, then restarts from BOTTOM. TOP is defined as 0xFF when WGM0[2:0]=0x3. TOP is defined as OCR0A when WGM0[2:0]=0x7.

In non-inverting Compare Output mode, the Output Compare register (OC0x) is cleared on the compare match between TCNT0 and OCR0x, and set at BOTTOM. In inverting Compare Output mode, the output is set on compare match and cleared at BOTTOM. Due to the single-slope operation, the operating frequency of the Fast PWM mode can be twice as high as the phase correct PWM modes, which use dual-slope operation. This high frequency makes the Fast PWM mode well suited for power regulation,

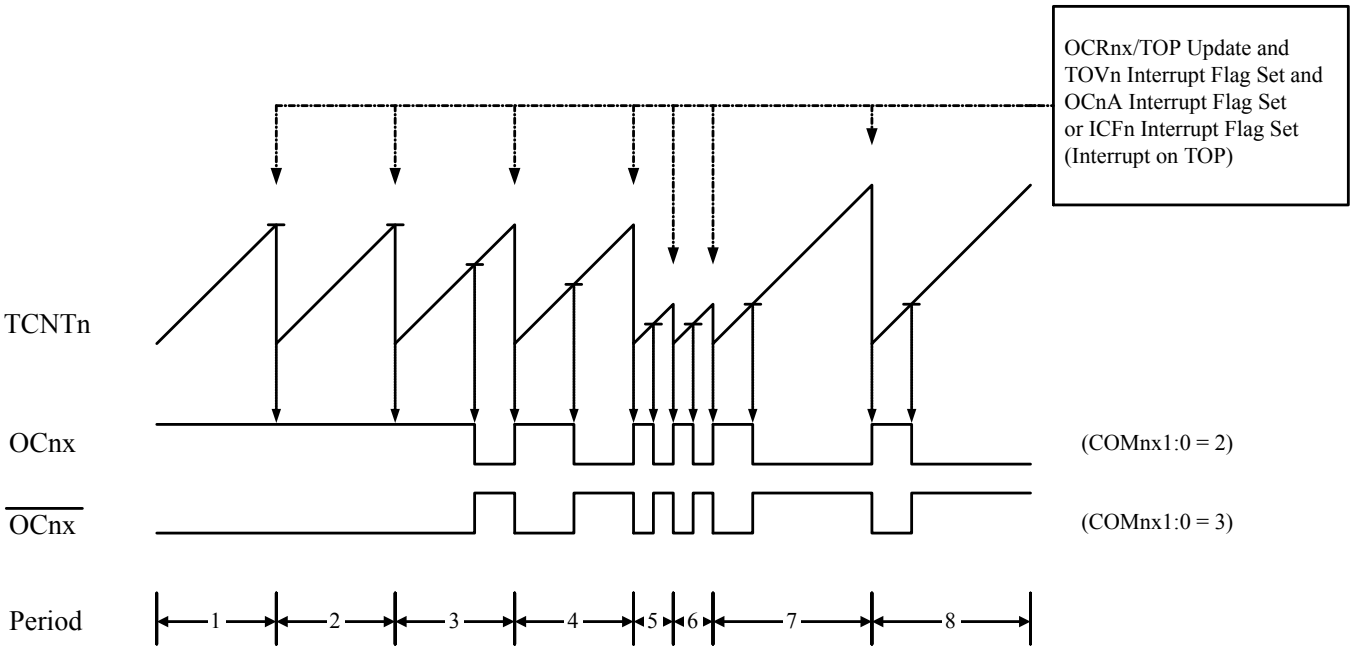
In non-inverting Compare Output mode, the Output Compare (OC1x) is cleared on the compare match between TCNT1 and OCR1x, and set at BOTTOM. In inverting Compare Output mode output is set on compare match and cleared at BOTTOM. Due to the single-slope operation, the operating frequency of the Fast PWM mode can be twice as high as the phase correct and phase and frequency correct PWM modes that use dual-slope operation. This high frequency makes the Fast PWM mode well suited for power regulation, rectification, and DAC applications. High frequency allows physically small sized external components (coils, capacitors), hence reduces total system cost.

The PWM resolution for Fast PWM can be fixed to 8-, 9-, or 10-bit, or defined by either ICR1 or OCR1A. The minimum resolution allowed is 2-bit (ICR1 or OCR1A register set to 0x0003), and the maximum resolution is 16-bit (ICR1 or OCR1A registers set to MAX). The PWM resolution in bits can be calculated by using the following equation:

$$R_{\mathrm{FPWM}} = \frac{\log(\mathrm{TOP}+1)}{\log(2)}$$

In Fast PWM mode the counter is incremented until the counter value matches either one of the fixed values 0x00FF, 0x01FF, or 0x03FF (WGM1[3:0] = 0x5, 0x6, or 0x7), the value in ICR1 (WGM1[3:0]=0xE), or the value in OCR1A (WGM1[3:0]=0xF). The counter is then cleared at the following timer clock cycle. The timing diagram for the Fast PWM mode using OCR1A or ICR1 to define TOP is shown below. The TCNT1 value is in the timing diagram shown as a histogram for illustrating the single-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal lines on the TCNT1 slopes mark compare matches between OCR1x and TCNT1. The OC1x Interrupt Flag will be set when a compare match occurs.

Figure 21-7. Fast PWM Mode, Timing Diagram



Note: The “n” in the register and bit names indicates the device number (n = 1 for Timer/Counter 1), and the “x” indicates Output Compare unit (A/B).

The Timer/Counter Overflow Flag (TOV1) is set each time the counter reaches TOP. In addition, when either OCR1A or ICR1 is used for defining the TOP value, the OC1A or ICF1 Flag is set at the same timer clock cycle TOV1 is set. If one of the interrupts are enabled, the interrupt handler routine can be used for updating the TOP and compare values.

4. When entering Power-save or ADC Noise Reduction mode after having written to TCNT2, OCR2x, or TCCR2x, the user must wait until the written register has been updated if TC2 is used to wake up the device. Otherwise, the MCU will enter sleep mode before the changes are effective. This is particularly important if any of the Output Compare2 interrupts is used to wake up the device, since the Output Compare function is disabled during writing to OCR2x or TCNT2. If the write cycle is not finished, and the MCU enters sleep mode before the corresponding OCR2xUB bit returns to zero, the device will never receive a compare match interrupt, and the MCU will not wake up.
5. If TC2 is used to wake the device up from Power-save or ADC Noise Reduction mode, precautions must be taken if the user wants to re-enter one of these modes: If re-entering sleep mode within the TOSC1 cycle, the interrupt will immediately occur and the device wake up again. The result is multiple interrupts and wake-ups within one TOSC1 cycle from the first interrupt. If the user is in doubt whether the time before re-entering Power-save or ADC Noise Reduction mode is sufficient, the following algorithm can be used to ensure that one TOSC1 cycle has elapsed:
    5.1. Write a value to TCCR2x, TCNT2, or OCR2x.
    5.2. Wait until the corresponding Update Busy Flag in ASSR returns to zero.
    5.3. Enter Power-save or ADC Noise Reduction mode.
6. When the asynchronous operation is selected, the 32.768kHz oscillator for TC2 is always running, except in Power-down and Standby modes. After a Power-up Reset or wake-up from Power-down or Standby mode, the user should be aware of the fact that this oscillator might take as long as one second to stabilize. The user is advised to wait for at least one second before using TC2 after power-up or wake-up from Power-down or Standby mode. The contents of all TC2 Registers must be considered lost after a wake-up from Power-down or Standby mode due to unstable clock signal upon start-up, no matter whether the Oscillator is in use or a clock signal is applied to the TOSC1 pin.
7. Description of wake up from Power-save or ADC Noise Reduction mode when the timer is clocked asynchronously: When the interrupt condition is met, the wake up process is started on the following cycle of the timer clock, that is, the timer is always advanced by at least one before the processor can read the counter value. After wake-up, the MCU is halted for four cycles, it executes the interrupt routine, and resumes execution from the instruction following SLEEP.
8. Reading of the TCNT2 Register shortly after wake-up from Power-save may give an incorrect result. Since TCNT2 is clocked on the asynchronous TOSC clock, reading TCNT2 must be done through a register synchronized to the internal I/O clock domain. Synchronization takes place for every rising TOSC1 edge. When waking up from Power-save mode, and the I/O clock ($clk_{I/O}$) again becomes active, TCNT2 will read as the previous value (before entering sleep) until the next rising TOSC1 edge. The phase of the TOSC clock after waking up from Power-save mode is essentially unpredictable, as it depends on the wake-up time. The recommended procedure for reading TCNT2 is thus as follows:
    8.1. Wait for the corresponding Update Busy Flag to be cleared.
    8.2. Read TCNT2.

During asynchronous operation, the synchronization of the Interrupt Flags for the asynchronous timer takes 3 processor cycles plus one timer cycle. The timer is therefore advanced by at least one before the processor can read the timer value causing the setting of the Interrupt Flag. The Output Compare pin is changed on the timer clock and is not synchronized to the processor clock.

24. SPI – Serial Peripheral Interface

24.1. Features

- Full-duplex, Three-wire Synchronous Data Transfer
- Master or Slave Operation
- LSB First or MSB First Data Transfer
- Seven Programmable Bit Rates
- End of Transmission Interrupt Flag
- Write Collision Flag Protection
- Wake-up from Idle Mode
- Double Speed (CK/2) Master SPI Mode

24.2. Overview

The Serial Peripheral Interface (SPI) allows high-speed synchronous data transfer between the device and peripheral units, or between several AVR devices.

The USART can also be used in Master SPI mode, please refer to *USART in SPI Mode* chapter.

To enable the SPI module, Power Reduction Serial Peripheral Interface bit in the Power Reduction Register (PRSPI0.PRR) must be written to '0'.

Bits 1:0 – ACISn: Analog Comparator Interrupt Mode Select [n = 1:0]

These bits determine which comparator events that trigger the Analog Comparator interrupt.

Table 28-3. ACIS[1:0] Settings

ACIS1	ACIS0	Interrupt Mode
0	0	Comparator Interrupt on Output Toggle.
0	1	Reserved
1	0	Comparator Interrupt on Falling Output Edge.
1	1	Comparator Interrupt on Rising Output Edge.

When changing the ACIS1/ACIS0 bits, the Analog Comparator Interrupt must be disabled by clearing its Interrupt Enable bit in the ACSR Register. Otherwise an interrupt can occur when the bits are changed.

- To protect only the Boot Loader Flash section from a software update by the MCU
- To protect only the Application Flash section from a software update by the MCU
- Allow software update in the entire Flash

The Boot Lock bits can be set in software and in Serial or Parallel Programming mode, but they can be cleared by a Chip Erase command only. The general Write Lock (Lock Bit mode 2) does not control the programming of the Flash memory by SPM instruction. Similarly, the general Read/Write Lock (Lock Bit mode 1) does not control reading nor writing by LPM/SPM, if it is attempted.

Table 32-2. Boot Lock Bit0 Protection Modes (Application Section)

BLB0 Mode	BLB02	BLB01	Protection
1	1	1	No restrictions for SPM or LPM accessing the Application section.
2	1	0	SPM is not allowed to write to the Application section.
3	0	0	SPM is not allowed to write to the Application section, and LPM executing from the Boot Loader section is not allowed to read from the Application section. If Interrupt Vectors are placed in the Boot Loader section, interrupts are disabled while executing from the Application section.
4	0	1	LPM executing from the Boot Loader section is not allowed to read from the Application section. If Interrupt Vectors are placed in the Boot Loader section, interrupts are disabled while executing from the Application section.

Note: “1” means unprogrammed, “0” means programmed.

Table 32-3. Boot Lock Bit1 Protection Modes (Boot Loader Section)

BLB1 Mode	BLB12	BLB11	Protection
1	1	1	No restrictions for SPM or LPM accessing the Boot Loader section.
2	1	0	SPM is not allowed to write to the Boot Loader section.
3	0	0	SPM is not allowed to write to the Boot Loader section, and LPM executing from the Application section is not allowed to read from the Boot Loader section. If Interrupt Vectors are placed in the Application section, interrupts are disabled while executing from the Boot Loader section.
4	0	1	LPM executing from the Application section is not allowed to read from the Boot Loader section. If Interrupt Vectors are placed in the Application section, interrupts are disabled while executing from the Boot Loader section.

Note: “1” means unprogrammed, “0” means programmed.

32.6. Entering the Boot Loader Program

Entering the Boot Loader takes place by a jump or call from the application program. This may be initiated by a trigger such as a command received via USART, or SPI interface. Alternatively, the Boot Reset Fuse can be programmed so that the Reset Vector is pointing to the Boot Flash start address after a reset. In this case, the Boot Loader is started after a reset. After the application code is loaded, the program can

Signature Byte	Z-pointer Address
Serial Number Byte 4	0x0013
Serial Number Byte 6	0x0015
Serial Number Byte 7	0x0016
Serial Number Byte 8	0x0017

Note: All other addresses are reserved for future use.

32.8.11. Preventing Flash Corruption

During periods of low V_{CC}, the Flash program can be corrupted because the supply voltage is too low for the CPU and the Flash to operate properly. These issues are the same as for board level systems using the Flash, and the same design solutions should be applied.

A Flash program corruption can be caused by two situations when the voltage is too low. First, a regular write sequence to the Flash requires a minimum voltage to operate correctly. Secondly, the CPU itself can execute instructions incorrectly, if the supply voltage for executing instructions is too low.

Flash corruption can easily be avoided by following these design recommendations (one is sufficient):

1. If there is no need for a Boot Loader update in the system, program the Boot Loader Lock bits to prevent any Boot Loader software updates.
2. Keep the AVR RESET active (low) during periods of insufficient power supply voltage. This can be done by enabling the internal Brown-out Detector (BOD) if the operating voltage matches the detection level. If not, an external low V_{CC} reset protection circuit can be used. If a reset occurs while a write operation is in progress, the write operation will be completed provided that the power supply voltage is sufficient.
3. Keep the AVR core in Power-down sleep mode during periods of low V_{CC}. This will prevent the CPU from attempting to decode and execute instructions, effectively protecting the SPMCSR Register and thus the Flash from unintentional writes.

32.8.12. Programming Time for Flash when Using SPM

The calibrated RC Oscillator is used to time Flash accesses. The following table shows the typical programming time for Flash accesses from the CPU.

Table 32-6. SPM Programming Time

Symbol	Min. Programming Time	Max. Programming Time
Flash write (Page Erase, Page Write, and write Lock bits by SPM)	3.2ms	3.4ms

Note: Minimum and maximum programming time is per individual operation.

32.8.13. Simple Assembly Code Example for a Boot Loader

```
;-the routine writes one page of data from RAM to Flash

; the first data location in RAM is pointed to by the Y pointer

; the first data location in Flash is pointed to by the Z-pointer

;-error handling is not included
 ;-the routine must be placed inside the Boot space
```

35.1.2. Idle Supply Current

Figure 35-6. ATmega48PB/88PB: Idle Supply Current vs. Low Frequency (0.1-1.0MHz)

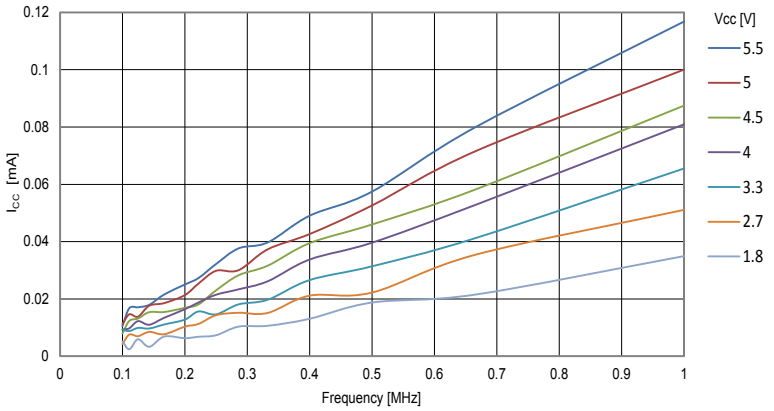


Figure 35-7. ATmega48PB/88PB: Idle Supply Current vs. Frequency (1-20MHz)

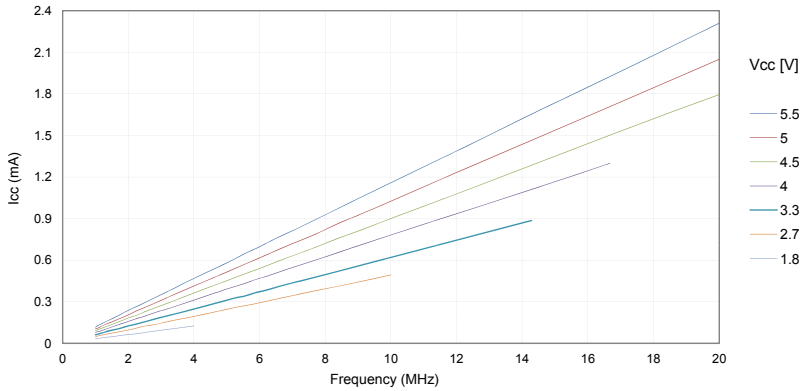


Figure 35-8. ATmega48PB/88PB: Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 128kHz)

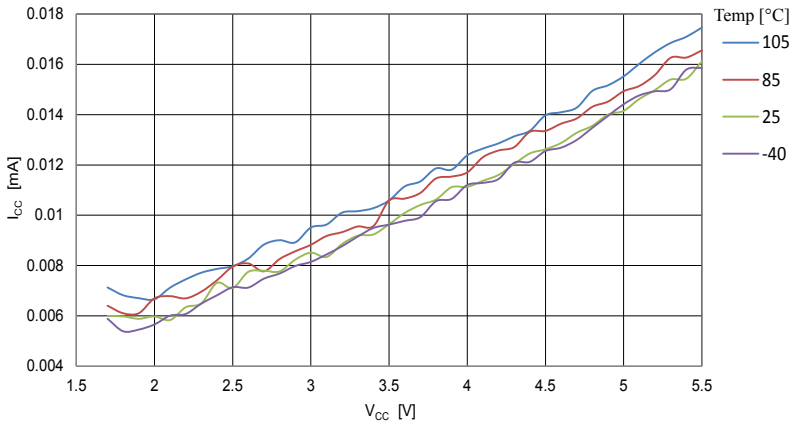
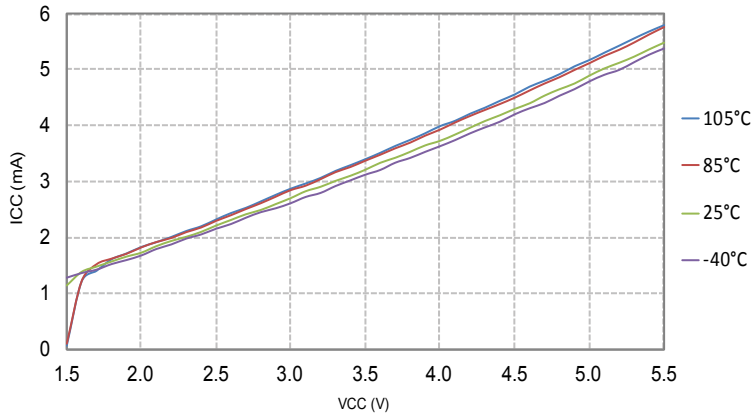


Figure 35-53. ATmega168PB: Active Supply Current vs. V_{CC} (Internal RC Oscillator, 8MHz)



35.2.2. Idle Supply Current

Figure 35-54. ATmega168PB: Idle Supply Current vs. Low Frequency (0.1-1.0MHz)

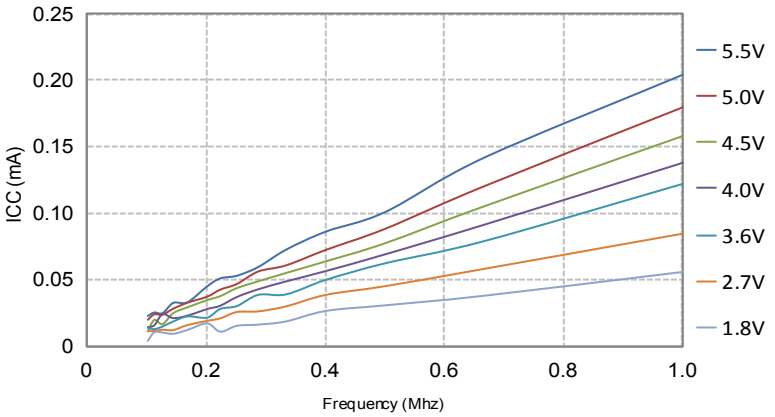
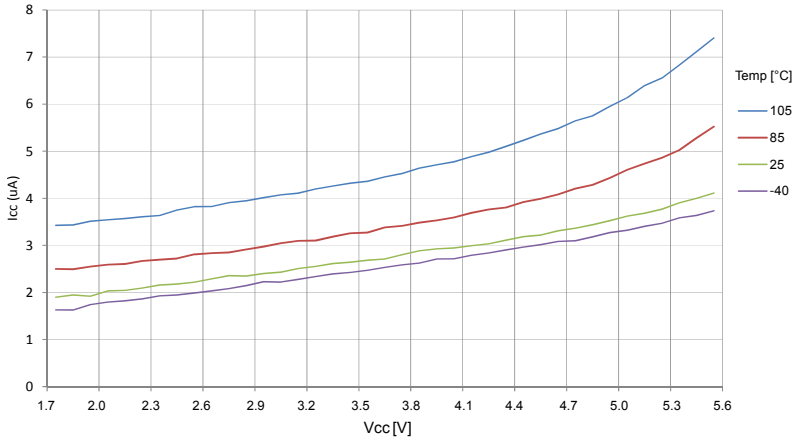
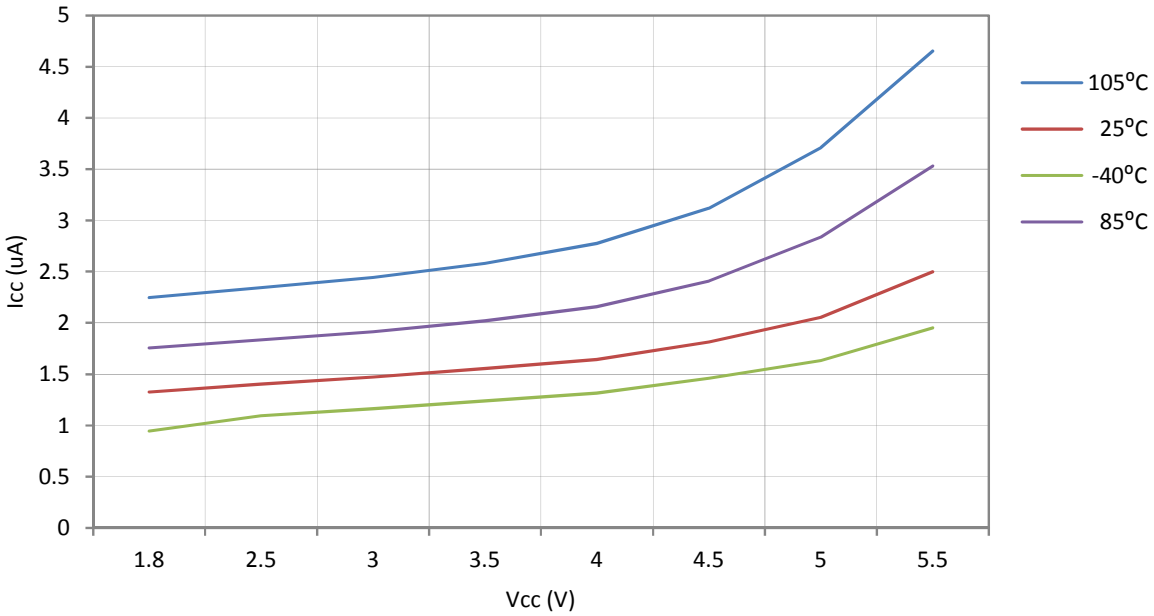


Figure 35-60. ATmega168PB: Power-Down Supply Current vs. V_{CC} (Watchdog Timer Enabled)



35.2.5. Power-save Supply Current

Figure 35-61. ATmega168PB: Power-Save Supply Current vs. V_{CC}



35.2.8. Pin Driver Strength

Figure 35-69. ATmega168PB: I/O Pin Output Voltage vs. Sink Current (V_{CC} = 3V)

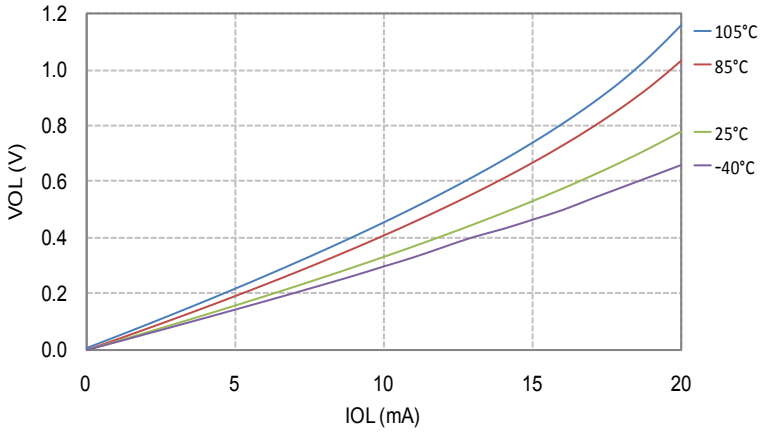


Figure 35-70. ATmega168PB: I/O Pin Output Voltage vs. Sink Current (V_{CC} = 5V)

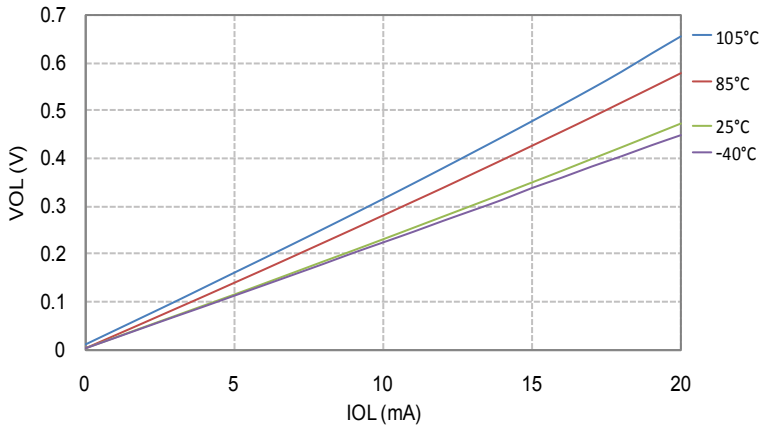


Figure 35-87. ATmega168PB: Calibrated 8MHz RC Oscillator Frequency vs. Temperature

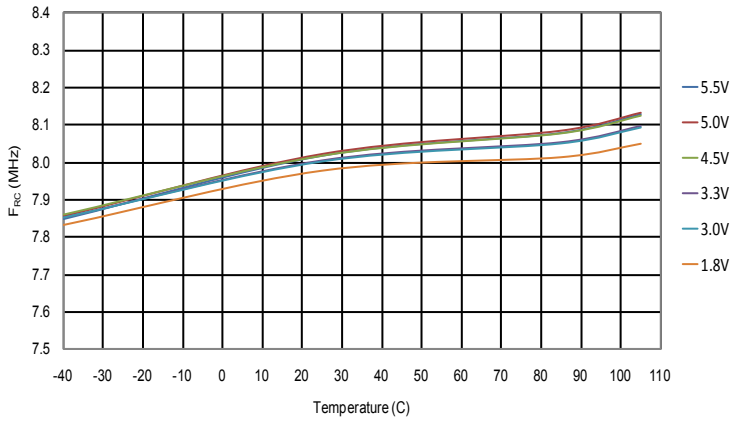
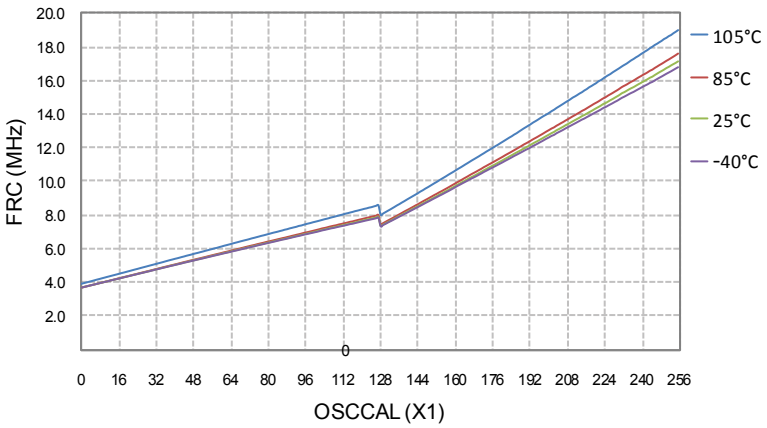


Figure 35-88. ATmega168PB: Calibrated 8MHz RC Oscillator Frequency vs. OSCCAL Value



39. Errata

39.1. Errata ATmega48PB

The revision letter in this section refers to the revision of the ATmega48PB device.

39.1.1. Rev. A

– Wrong device ID when using debugWire

– Power consumption in power save modes

– USART start-up functionality not working

– External capacitor on AREF pin

1.) Wrong device ID when using debugWire

The device ID returned using debugWire is incorrect.

Problem Fix/Workaround

None.

2.) Power consumption in power save modes

Power consumption in power save modes will be higher due to improper control of internal power management.

Problem Fix/Workaround

None.

3.) USART start-up functionality not working

While in power save modes, the USART start bit detection logic fails to wakeup the device.

Problem Fix/Workaround

None.

4.) External capacitor on AREF pin

If an external capacitor is used on the analog reference pin (AREF), it should be equal to or larger than 100nF. Smaller capacitor value can make the AREF buffer unstable with large ringing which will reduce the accuracy of the ADC.

Problem Fix/Workaround

None.

39.1.2. Rev. B

– External capacitor on AREF pin

1.) External capacitor on AREF pin

If an external capacitor is used on the analog reference pin (AREF), it should be equal to or larger than 100nF. Smaller capacitor value can make the AREF buffer unstable with large ringing which will reduce the accuracy of the ADC.

Problem Fix/Workaround

None.