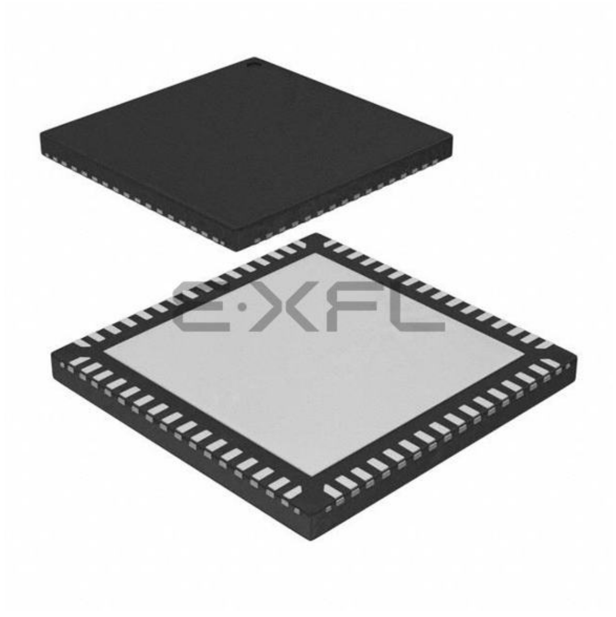
Microchip Technology - ATMEGA169P-15MT Datasheet



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	16MHz
Connectivity	SPI, UART/USART, USI
Peripherals	Brown-out Detect/Reset, LCD, POR, PWM, WDT
Number of I/O	54
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-QFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega169p-15mt

The AVR® core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the arithmetic logic unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The Atmel® ATmega169P provides the following features: 16Kbytes of in-system programmable flash with read-while-write capabilities, 512 bytes EEPROM, 1Kbyte SRAM, 53 general purpose I/O lines, 32 general purpose working registers, a JTAG interface for boundary-scan, on-chip debugging support and programming, a complete on-chip LCD controller with internal step-up voltage, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, universal serial interface with start condition detector, an 8-channel, 10-bit ADC, a programmable watchdog timer with internal oscillator, an SPI serial port, and five software selectable power saving modes. The idle mode stops the CPU while allowing the SRAM, timer/counters, SPI port, and interrupt system to continue functioning. The power-down mode saves the register contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset. In power-save mode, the asynchronous timer and the LCD controller continues to run, allowing the user to maintain a timer base and operate the LCD display while the rest of the device is sleeping. The ADC noise reduction mode stops the CPU and all I/O modules except asynchronous timer, LCD controller 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.

The device is manufactured using Atmel high density non-volatile memory technology. The on-chip ISP flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional non-volatile 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 ATmega169P is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega169P AVR 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 Automotive Quality Grade

The Atmel ATmega169P have been developed and manufactured according to the most stringent requirements of the international standard ISO-TS-16949. This data sheet contains limit values extracted from the results of extensive characterization (temperature and voltage). The quality and reliability of the ATmega169P have been verified during regular product qualification as per AEC-Q100 grade 3.

As indicated in the ordering information paragraph, the products are available in industrial temperature grades, but with equivalent automotive quality and reliability objectives. Different temperature identifiers have been defined as listed in Table 2-1.

Table 2-1. Temperature Grade Identification for Automotive Products

Temperature	Temperature Identifier	Comments
–40 to +85°C	T	Similar to industrial temperature grade but with automotive quality

11.2.3 EIFR – External Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x1C (0x3C)	**PCIF1**	**PCIF0**	–	–	–	–	–	**INTF0**	**EIFR**
Read/Write	R/W	R/W	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – PCIF1: Pin Change Interrupt Flag 1**

When a logic change on any PCINT15..8 pin triggers an interrupt request, PCIF1 becomes set (one). If the I-bit in SREG and the PCIE1 bit in EIMSK are set (one), the MCU will jump to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it.

- **Bit 6 – PCIF0: Pin Change Interrupt Flag 0**

When a logic change on any PCINT7:0 pin triggers an interrupt request, PCIF0 becomes set (one). If the I-bit in SREG and the PCIE0 bit in EIMSK are set (one), the MCU will jump to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it.

- **Bit 0 – INTF0: External Interrupt Flag 0**

When an edge or logic change on the INT0 pin triggers an interrupt request, INTF0 becomes set (one). If the I-bit in SREG and the INT0 bit in EIMSK are set (one), the MCU will jump to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it. This flag is always cleared when INT0 is configured as a level interrupt.

11.2.4 PCMSK1 – Pin Change Mask Register 1

Bit	7	6	5	4	3	2	1	0	
(0x6C)	**PCINT15**	**PCINT14**	**PCINT13**	**PCINT12**	**PCINT11**	**PCINT10**	**PCINT9**	**PCINT8**	**PCMSK1**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:0 – PCINT15:8: Pin Change Enable Mask 15..8**

Each PCINT15:8-bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT15:8 is set and the PCIE1 bit in EIMSK is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT15..8 is cleared, pin change interrupt on the corresponding I/O pin is disabled.

11.2.5 PCMSK0 – Pin Change Mask Register 0

Bit	7	6	5	4	3	2	1	0	
(0x6B)	**PCINT7**	**PCINT6**	**PCINT5**	**PCINT4**	**PCINT3**	**PCINT2**	**PCINT1**	**PCINT0**	**PCMSK0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:0 – PCINT7:0: Pin Change Enable Mask 7:0**

Each PCINT7:0 bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT7:0 is set and the PCIE0 bit in EIMSK is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT7:0 is cleared, pin change interrupt on the corresponding I/O pin is disabled.

Table 12-16 and Table 12-17 on page 69 relates the alternate functions of port E to the overriding signals shown in Figure 12-5 on page 60.

Table 12-16. Overriding Signals for Alternate Functions PE7:PE4

Signal Name	PE7/PCINT7	PE6/DO/ PCINT6	PE5/DI/SDA/ PCINT5	PE4/USCK/SCL/ PCINT4
PUOE	0	0	USI_TWO-WIRE	USI_TWO-WIRE
PUOV	0	0	0	0
DDOE	CKOUT[1]	0	USI_TWO-WIRE	USI_TWO-WIRE
DDOV	1	0	($\overline{SDA}$ + $\overline{PORTE5}$) × DDE5	(USI_SCL_HOLD × $\overline{PORTE4}$) + DDE4
PVOE	CKOUT[1]	USI_THREE-WIRE	USI_TWO-WIRE × DDE5	USI_TWO-WIRE× DDE4
PVOV	$clk_{I/O}$	DO	0	0
PTOE	–	–	0	USITC
DIEOE	PCINT7 ×PCIE0	PCINT6 × PCIE0	(PCINT5 × PCIE0) + USISIE	(PCINT4 × PCIE0) + USISIE
DIEOV	1	1	1	1
DI	PCINT7 INPUT	PCINT6 INPUT	DI/SDA INPUT PCINT5 INPUT	USCKL/SCL INPUT PCINT4 INPUT
AIO	–	–	–	–

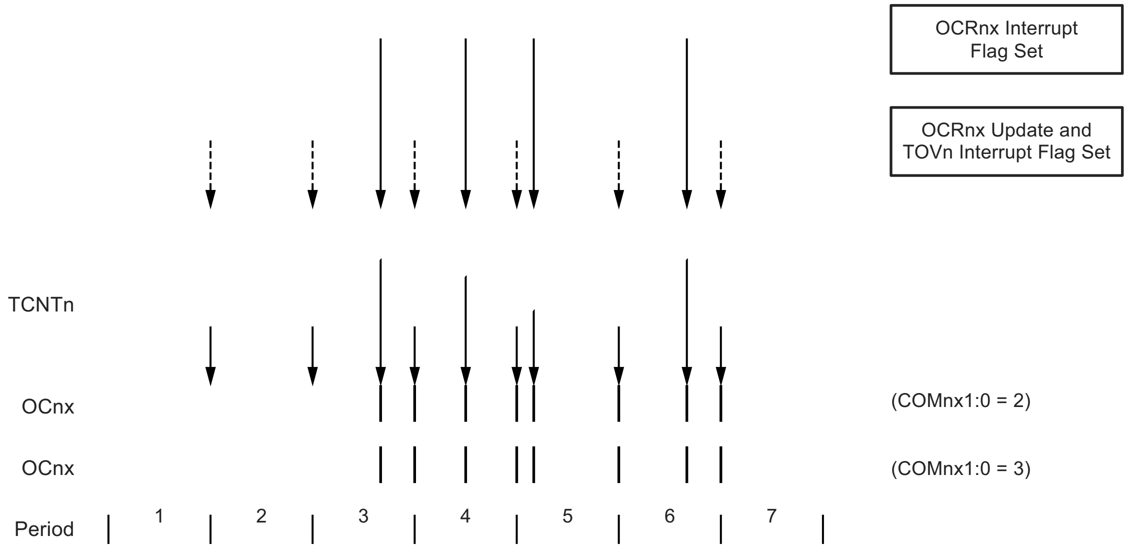
Note: 1. CKOUT is one if the CKOUT fuse is programmed

Table 12-17. Overriding Signals for Alternate Functions in PE3:PE0

Signal Name	PE3/AIN1/ PCINT3	PE2/XCK/AIN0/ PCINT2	PE1/TXD/ PCINT1	PE0/RXD/PCINT0
PUOE	0	0	TXENn	RXENn
PUOV	0	0	0	PORTE0 × $\overline{PUD}$
DDOE	0	0	TXENn	RXENn
DDOV	0	0	1	0
PVOE	0	XCK OUTPUT ENABLE	TXENn	0
PVOV	0	XCK	TXD	0
PTOE	–	–	–	–
DIEOE	(PCINT3 × PCIE0) + AIN1D[1]	(PCINT2 × PCIE0) + AIN0D[1]	PCINT1 × PCIE0	PCINT0 × PCIE0
DIEOV	PCINT3 × PCIE0	PCINT2 × PCIE0	1	1
DI	PCINT3 INPUT	XCK/PCINT2 INPUT	PCINT1 INPUT	RXD/PCINT0 INPUT
AIO	AIN1 INPUT	AIN0 INPUT	–	–

Note: 1. AIN0D and AIN1D is described in Section 20.2.3 “DIDR1 – Digital Input Disable Register 1” on page 180.

Figure 13-6. Fast PWM Mode, Timing Diagram



The timer/counter overflow flag (TOV0) is set each time the counter reaches MAX. If the interrupt is enabled, the interrupt handler routine can be used for updating the compare value.

In fast PWM mode, the compare unit allows generation of PWM waveforms on the OC0A pin. Setting the COM0A1:0 bits to two will produce a non-inverted PWM and an inverted PWM output can be generated by setting the COM0A1:0 to three (See Table 13-4 on page 88). The actual OC0A value will only be visible on the port pin if the data direction for the port pin is set as output. The PWM waveform is generated by setting (or clearing) the OC0A register at the compare match between OCR0A and TCNT0, and clearing (or setting) the OC0A register at the timer clock cycle the counter is cleared (changes from MAX to BOTTOM).

The PWM frequency for the output can be calculated by the following equation

$$:f_{OCnxPWM} = \frac{f_{\text{clk\_I/O}}}{N \cdot 256}$$

The *N* variable represents the prescale factor (1, 8, 64, 256, or 1024).

The extreme values for the OCR0A register represents special cases when generating a PWM waveform output in the fast PWM mode. If the OCR0A is set equal to BOTTOM, the output will be a narrow spike for each MAX+1 timer clock cycle. Setting the OCR0A equal to MAX will result in a constantly high or low output (depending on the polarity of the output set by the COM0A1:0 bits.)

A frequency (with 50% duty cycle) waveform output in fast PWM mode can be achieved by setting OC0A to toggle its logical level on each compare match (COM0A1:0 = 1). The waveform generated will have a maximum frequency of $f_{OC0} = f_{clk\_I/O}/2$ when OCR0A is set to zero. This feature is similar to the OC0A toggle in CTC mode, except the double buffer feature of the output compare unit is enabled in the fast PWM mode.

13.9.3 OCR0A – Output Compare Register A

Bit	7	6	5	4	3	2	1	0	
0x27 (0x47)	**OCR0A[7:0]**								**OCR0A**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The output compare register A contains an 8-bit value that is continuously compared with the counter value (TCNT0). A match can be used to generate an output compare interrupt, or to generate a waveform output on the OC0A pin.

13.9.4 TIMSK0 – Timer/Counter 0 Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
(0x6E)	–	–	–	–	–	–	**OCIE0A**	**TOIE0**	**TIMSK0**
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 1 – OCIE0A: Timer/Counter0 Output Compare Match A Interrupt Enable**

When the OCIE0A bit is written to one, and the I-bit in the status register is set (one), the timer/counter0 compare match A interrupt is enabled. The corresponding interrupt is executed if a compare match in timer/counter0 occurs, i.e., when the OCF0A bit is set in the timer/counter 0 interrupt flag register – TIFR0.

- **Bit 0 – TOIE0: Timer/Counter0 Overflow Interrupt Enable**

When the TOIE0 bit is written to one, and the I-bit in the status register is set (one), the timer/counter0 overflow interrupt is enabled. The corresponding interrupt is executed if an overflow in timer/counter0 occurs, i.e., when the TOV0 bit is set in the timer/counter 0 interrupt flag register – TIFR0.

13.9.5 TIFR0 – Timer/Counter 0 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x15 (0x35)	–	–	–	–	–	–	**OCF0A**	**TOV0**	**TIFR0**
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

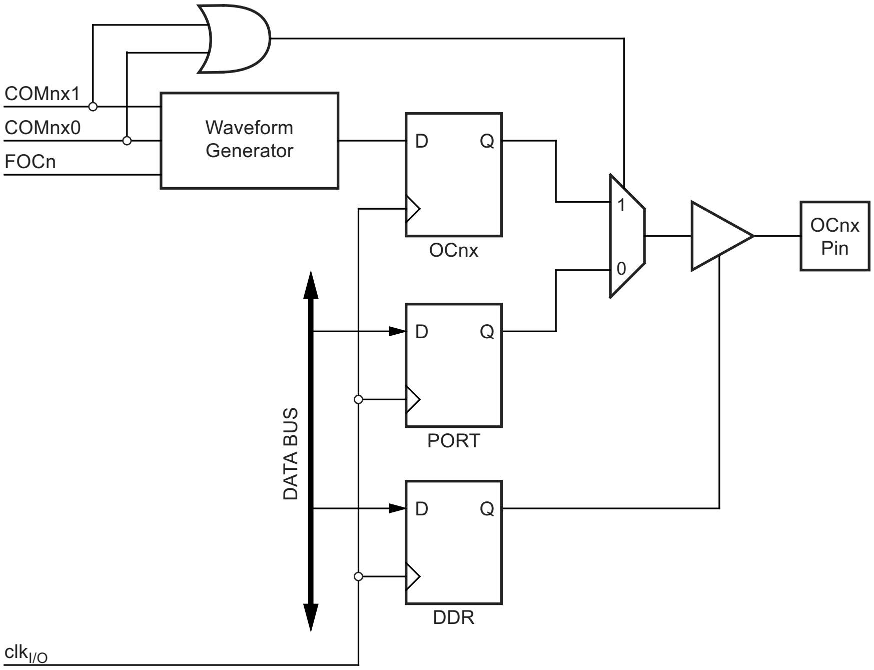
- **Bit 1 – OCF0A: Output Compare Flag 0 A**

The OCF0A bit is set (one) when a compare match occurs between the timer/counter0 and the data in OCR0A – output compare register0. OCF0A is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF0A is cleared by writing a logic one to the flag. When the I-bit in SREG, OCIE0A (timer/counter0 compare match interrupt enable), and OCF0A are set (one), the timer/counter0 compare match interrupt is executed.

- **Bit 0 – TOV0: Timer/Counter0 Overflow Flag**

The bit TOV0 is set (one) when an overflow occurs in timer/counter0. TOV0 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV0 is cleared by writing a logic one to the flag. When the SREG I-bit, TOIE0 (timer/counter0 overflow interrupt enable), and TOV0 are set (one), the timer/counter0 overflow interrupt is executed. In phase correct PWM mode, this bit is set when timer/counter0 changes counting direction at 0x00.

Figure 14-5. Compare Match Output Unit, Schematic



The general I/O port function is overridden by the output compare (OC1x) from the waveform generator if either of the COM1x1:0 bits are set. However, the OC1x pin direction (input or output) is still controlled by the *data direction register* (DDR) for the port pin. The data direction register bit for the OC1x pin (DDR_OC1x) must be set as output before the OC1x value is visible on the pin. The port override function is generally independent of the waveform generation mode, but there are some exceptions. Refer to Table 14-2 on page 111, Table 14-3 on page 111 and Table 14-4 on page 112 for details.

The design of the output compare pin logic allows initialization of the OC1x state before the output is enabled. Note that some COM1x1:0 bit settings are reserved for certain modes of operation.
See Section 14.11 “16-bit Timer/Counter Register Description” on page 111

The COM1x1:0 bits have no effect on the input capture unit.

14.8.1 Compare Output Mode and Waveform Generation

The waveform generator uses the COM1x1:0 bits differently in normal, CTC, and PWM modes. For all modes, setting the COM1x1:0 = 0 tells the waveform generator that no action on the OC1x register is to be performed on the next compare match. For compare output actions in the non-PWM modes refer to Table 14-2 on page 111. For fast PWM mode refer to Table 14-3 on page 111, and for phase correct and phase and frequency correct PWM refer to Table 14-4 on page 112.

A change of the COM1x1:0 bits state will have effect at the first compare match after the bits are written. For non-PWM modes, the action can be forced to have immediate effect by using the FOC1x strobe bits.

14.9 Modes of Operation

The mode of operation, i.e., the behavior of the timer/counter and the output compare pins, is defined by the combination of the waveform generation mode (WGM13:0) and compare output mode (COM1x1:0) bits. The compare output mode bits do not affect the counting sequence, while the waveform generation mode bits do. The COM1x1:0 bits control whether the PWM output generated should be inverted or not (inverted or non-inverted PWM). For non-PWM modes the COM1x1:0 bits control whether the output should be set, cleared or toggle at a compare match
(see Section 14.8 “Compare Match Output Unit” on page 101).

For detailed timing information refer to Section 14.10 “Timer/Counter Timing Diagrams” on page 109.

The extreme values for the OCR1x register represents special cases when generating a PWM waveform output in the phase and frequency correct PWM mode. If the OCR1x is set equal to BOTTOM the output will be continuously low and if set equal to TOP the output will be set to high for non-inverted PWM mode. For inverted PWM the output will have the opposite logic values. If OCR1A is used to define the TOP value (WGM13:0 = 9) and COM1A1:0 = 1, the OC1A output will toggle with a 50% duty cycle.

14.10 Timer/Counter Timing Diagrams

The timer/counter is a synchronous design and the timer clock (clk_{T1}) is therefore shown as a clock enable signal in the following figures. The figures include information on when interrupt flags are set, and when the OCR1x register is updated with the OCR1x buffer value (only for modes utilizing double buffering). Figure 14-10 shows a timing diagram for the setting of OCF1x.

Figure 14-10.Timer/Counter Timing Diagram, Setting of OCF1x, no Prescaling

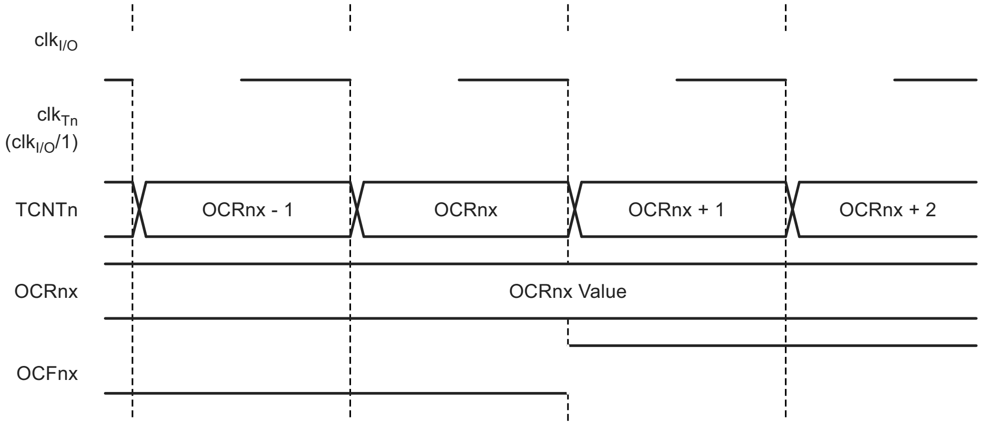
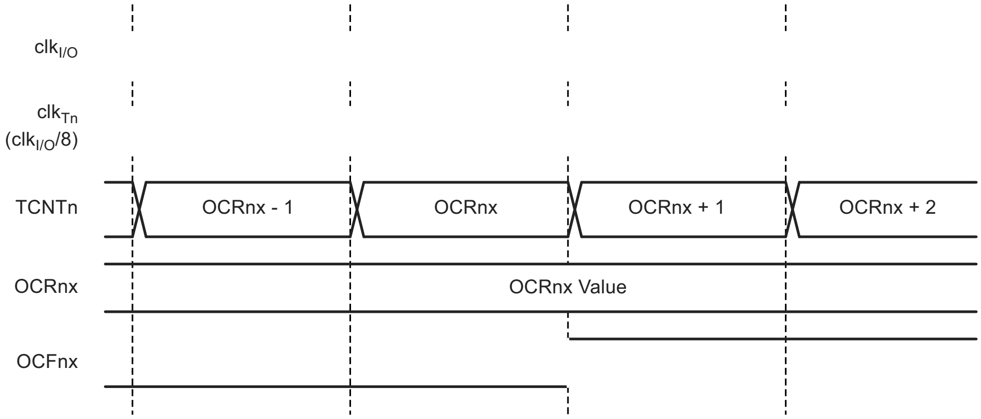


Figure 14-11 shows the same timing data, but with the prescaler enabled.

Figure 14-11.Timer/Counter Timing Diagram, Setting of OCF1x, with Prescaler ($f_{clk\_I/O}$/8)



- **Bit 6, 3 – WGM21:0: Waveform Generation Mode**

These bits control the counting sequence of the counter, the source for the maximum (TOP) counter value, and what type of waveform generation to be used. Modes of operation supported by the timer/counter unit are: normal mode, clear timer on compare match (CTC) mode, and two types of pulse width modulation (PWM) modes.
See Table 16-2 and Section 16.6 “Modes of Operation” on page 124.

Table 16-2. Waveform Generation Mode Bit Description[1]

Mode	WGM21 (CTC2)	WGM20 (PWM2)	Timer/Counter Mode of Operation	TOP	Update of OCR2A at	TOV2 Flag Set on
0	0	0	Normal	0xFF	Immediate	MAX
1	0	1	PWM, phase correct	0xFF	TOP	BOTTOM
2	1	0	CTC	OCR2A	Immediate	MAX
3	1	1	Fast PWM	0xFF	BOTTOM	MAX

Note: 1. The CTC2 and PWM2 bit definition names are now obsolete. Use the WGM21:0 definitions. However, the functionality and location of these bits are compatible with previous versions of the timer.

- **Bit 5:4 – COM2A1:0: Compare Match Output Mode A**

These bits control the output compare pin (OC2A) behavior. If one or both of the COM2A1:0 bits are set, the OC2A output overrides the normal port functionality of the I/O pin it is connected to. However, note that the data direction register (DDR) bit corresponding to OC2A pin must be set in order to enable the output driver.

When OC2A is connected to the pin, the function of the COM2A1:0 bits depends on the WGM21:0 bit setting. Table 16-3 shows the COM2A1:0 bit functionality when the WGM21:0 bits are set to a normal or CTC mode (non-PWM).

Table 16-3. Compare Output Mode, non-PWM Mode

COM2A1	COM2A0	Description
0	0	Normal port operation, OC2A disconnected.
0	1	Toggle OC2A on compare match.
1	0	Clear OC2A on compare match.
1	1	Set OC2A on compare match.

Table 16-4 shows the COM2A1:0 bit functionality when the WGM21:0 bits are set to fast PWM mode.

Table 16-4. Compare Output Mode, Fast PWM Mode[1]

COM2A1	COM2A0	Description
0	0	Normal port operation, OC2A disconnected.
0	1	Reserved
1	0	Clear OC2A on compare match, set OC2A at BOTTOM (non-inverting mode).
1	1	Set OC2A on compare match, clear OC2A at BOTTOM (inverting mode).

Note: 1. A special case occurs when OCR2A equals TOP and COM2A1 is set. In this case, the compare match is ignored, but the set or clear is done at TOP. See Section 16.6.3 “Fast PWM Mode” on page 125 for more details.

17. SPI – Serial Peripheral Interface

17.1 Features

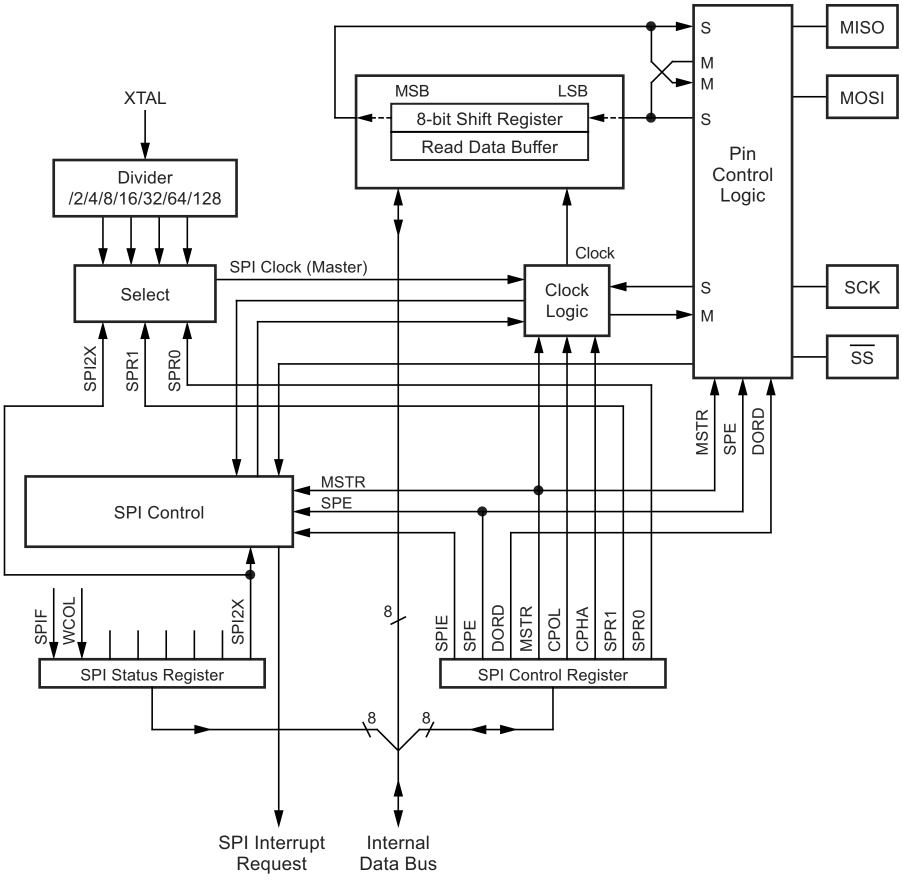
- Full-duplex, three-wire synchronous data transfer
- Master or slave operation
- LSB first or MSB first data transfer
- Seven programmable bit rates
- End of transmission interrupt flag
- Write collision flag protection
- Wake-up from idle mode
- Double speed (CK/2) master SPI mode

17.2 Overview

The serial peripheral interface (SPI) allows high-speed synchronous data transfer between the Atmel® ATmega169P and peripheral devices or between several AVR® devices.

The PRSPI bit in Section 8.9.2 “PRR – Power Reduction Register” on page 38 must be written to zero to enable SPI module.

Figure 17-1. SPI Block Diagram(1)



Note: 1. Refer to Figure 1-1 on page 3, and Table 12-6 on page 62 for SPI pin placement.

Table 21-3. Input Channel Selections

MUX4..0	Single Ended Input
00000	ADC0
00001	ADC1
00010	ADC2
00011	ADC3
00100	ADC4
00101	ADC5
00110	ADC6
00111	ADC7
01000	
01001	
01010	
01011	
01100	
01101	
01110	
01111	
10000	
10001	
10010	
10011	
10100	
10101	
10110	
10111	
11000	
11001	
11010	
11011	
11100	
11101	
11110	1.1V (V_{BG})
11111	0V (GND)

22.4.2 Updating the LCD

Display memory (LCDDR0, LCDDR1,.), LCD blanking (LCDBL), Low power waveform (LCDAB) and contrast control (LCDCCR) are latched prior to every new frame. There are no restrictions on writing these LCD register locations, but an LCD data update may be split between two frames if data are latched while an update is in progress. To avoid this, an interrupt routine can be used to update display memory, LCD blanking, Low power waveform, and contrast control, just after data are latched.

In the example below we assume SEG10 and COM1 and SEG4 in COM0 are the only segments changed from frame to frame. Data are stored in r20 and r21 for simplicity.

Assembly Code Example[1]

```
LCD_update:
      ; LCD Blanking and Low power waveform are unchanged.
      ; Update Display memory.
      sts    LCDDR0, r20
      sts    LCDDR6, r21
      ret
```

C Code Example[1]

```
Void LCD_update(unsigned char data1, data2);
{
      /* LCD Blanking and Low power waveform are unchanged. */
      /* Update Display memory. */
      LCDDR0 = data1;
      LCDDR6 = data2;
}
```

Note: 1. See Section 4. “About Code Examples” on page 8.

23. JTAG Interface and On-chip Debug System

23.1 Features

- JTAG (IEEE std. 1149.1 compliant) interface
- Boundary-scan capabilities according to the IEEE std. 1149.1 (JTAG) standard
- Debugger access to:
  - All internal peripheral units
  - Internal and external RAM
  - The internal register file
  - Program counter
  - EEPROM and flash memories
- Extensive on-chip debug support for break conditions, including
  - AVR® break instruction
  - Break on change of program memory flow
  - Single step break
  - Program memory break points on single address or address range
  - Data memory break points on single address or address range
- Programming of flash, EEPROM, fuses, and lock bits through the JTAG interface
- On-chip debugging supported by AVR Studio®

23.2 Overview

The AVR® IEEE std. 1149.1 compliant JTAG interface can be used for

- Testing PCBs by using the JTAG boundary-scan capability
- Programming the non-volatile memories, fuses and lock bits
- On-chip debugging

A brief description is given in the following sections. Detailed descriptions for programming via the JTAG interface, and using the boundary-scan chain can be found in the sections Section 26.9 “Programming via the JTAG Interface” on page 268 and Section 24. “IEEE 1149.1 (JTAG) Boundary-scan” on page 218**,** respectively. The on-chip debug support is considered being private JTAG instructions, and distributed within ATMEL and to selected third party vendors only.

Figure 23-1 on page 213 shows a block diagram of the JTAG interface and the on-chip debug system. The TAP controller is a state machine controlled by the TCK and TMS signals. The TAP controller selects either the JTAG instruction register or one of several data registers as the scan chain (shift register) between the TDI – input and TDO – output. The Instruction register holds JTAG instructions controlling the behavior of a data register.

The ID-register, bypass register, and the boundary-scan chain are the data registers used for board-level testing. The JTAG programming interface (actually consisting of several physical and virtual data registers) is used for serial programming via the JTAG interface. The internal scan chain and break point scan chain are used for on-chip debugging only.

24.5 Boundary-scan Chain

The boundary-scan chain has the capability of driving and observing the logic levels on the digital I/O pins, as well as the boundary between digital and analog logic for analog circuitry having off-chip connection.

24.5.1 Scanning the Digital Port Pins

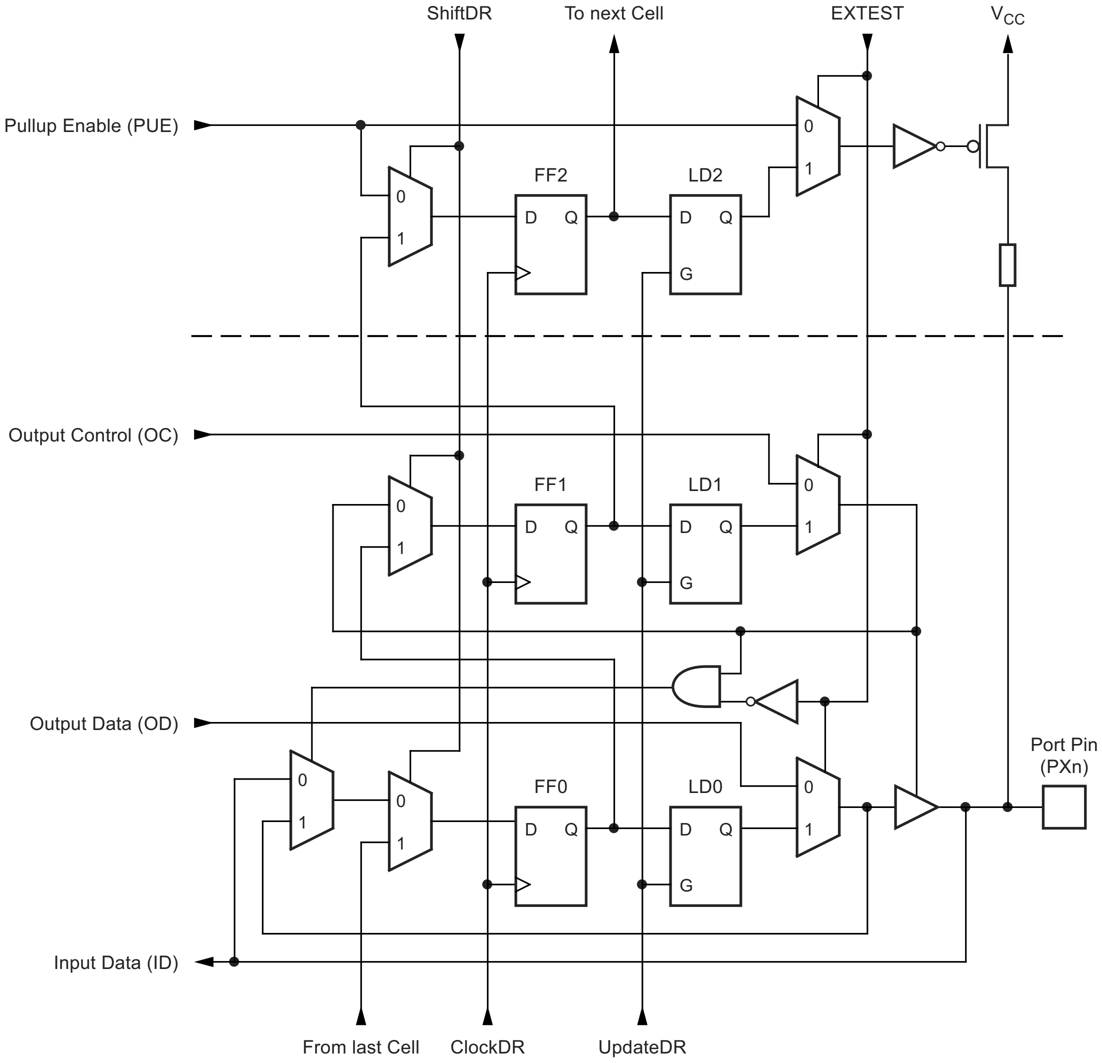
Figure 24-3 shows the boundary-scan cell for a bi-directional port pin with pull-up function. The cell consists of a standard boundary-scan cell for the pull-up enable – PUExn – function, and a bi-directional pin cell that combines the three signals output control – OCxn, output data – ODxn, and input data – IDxn, into only a two-stage shift register. The port and pin indexes are not used in the following description

The boundary-scan logic is not included in the figures in the datasheet. Figure 24-4 on page 222 shows a simple digital port pin as described in the section Section 12. "I/O-Ports" on page 55. The boundary-scan details from Figure 24-3 replaces the dashed box in Figure 24-4 on page 222.

When no alternate port function is present, the input data – ID – corresponds to the PINxn register value (but ID has no synchronizer), output data corresponds to the PORT register, output control corresponds to the data direction – DD register, and the pull-up enable – PUExn – corresponds to logic expression $\overline{\text{PUD}} \cdot \overline{\text{DDxn}} \cdot \text{PORTxn}$.

Digital alternate port functions are connected outside the dotted box in Figure 24-4 on page 222 to make the scan chain read the actual pin value. For analog function, there is a direct connection from the external pin to the analog circuit, and a scan chain is inserted on the interface between the digital logic and the analog circuitry.

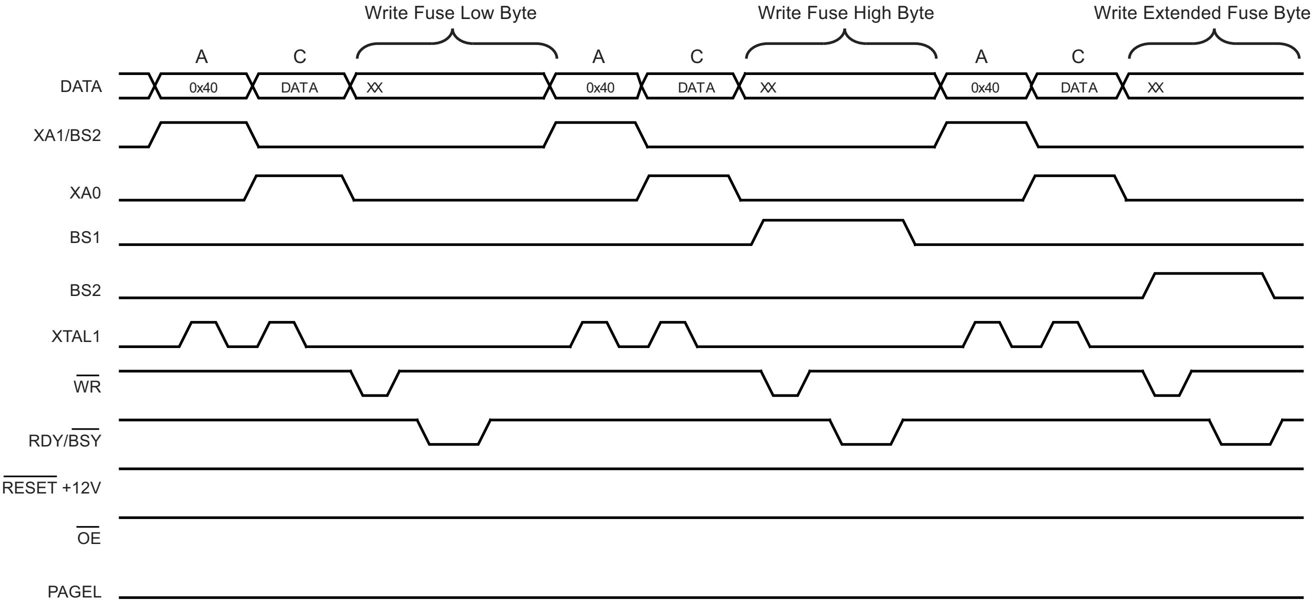
Figure 24-3. Boundary-scan Cell for Bi-directional Port Pin with Pull-up Function



25.8.12 Boot Loader: Simple Assembly Code Example

```
       ;-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
       ; (at least the Do_spm sub routine). Only code inside NRWW section can
       ; be read during Self-Programming (Page Erase and Page Write).
       ;-registers used: r0, r1, temp1 (r16), temp2 (r17), looplo (r24),
       ; loophi (r25), spmcrval (r20)
       ; storing and restoring of registers is not included in the routine
       ; register usage can be optimized at the expense of code size
       ;-It is assumed that either the interrupt table is moved to the Boot
       ; loader section or that the interrupts are disabled.
.equu              PAGESIZEB = PAGESIZE*2;PAGESIZEB is page size in BYTES, not words
.org SMALLBOOTSTART
Write_page:
       ;      Page Erase
       ldi    spmcrval, (1<<PGERS) | (1<<SPMEN)
       call   Do_spm

       ;      re-enable the RWW section
       ldi    spmcrval, (1<<RWWSRE) | (1<<SPMEN)
       call   Do_spm

       ;      transfer data from RAM to Flash page buffer
       ldi    looplo, low(PAGESIZEB)     ;init loop variable
       ldi    loophi, high(PAGESIZEB)    ;not required for PAGESIZEB<=256
Wrloop:
       ld     r0, Y+
       ld     r1, Y+
       ldi    spmcrval, (1<<SPMEN)
       call   Do_spm
       adiw   ZH:ZL, 2
       sbiw   loophi:looplo, 2           ;use subi for PAGESIZEB<=256
       brne   Wrloop

       ;      execute Page Write
       subi   ZL, low(PAGESIZEB)         ;restore pointer
       sbci   ZH, high(PAGESIZEB)        ;not required for PAGESIZEB<=256
       ldi    spmcrval, (1<<PGWRT) | (1<<SPMEN)
       call   Do_spm

       ;      re-enable the RWW section
       ldi    spmcrval, (1<<RWWSRE) | (1<<SPMEN)
       call   Do_spm

       ;      read back and check, optional
       ldi    looplo, low(PAGESIZEB)     ;init loop variable
       ldi    loophi, high(PAGESIZEB)    ;not required for PAGESIZEB<=256
       subi   YL, low(PAGESIZEB)         ;restore pointer
       sbci   YH, high(PAGESIZEB)
Rdloop:
       lpm    r0, Z+
       ld     r1, Y+
       cpse   r0, r1
       jmp    Error
       sbiw   loophi:looplo, 1           ;use subi for PAGESIZEB<=256
       brne   Rdloop
```

Figure 26-5. Programming the FUSES Waveforms



26.7.11 Programming the Lock Bits

The algorithm for programming the lock bits is as follows (refer to Section 26.7.4 “Programming the Flash” on page 256 for details on command and data loading):

1. A: Load command “0010 0000”.
2. C: Load data low byte. Bit n = “0” programs the lock bit. If LB mode 3 is programmed (LB1 and LB2 is programmed), it is not possible to program the boot lock bits by any external programming mode.
3. Give $\overline{\text{WR}}$ a negative pulse and wait for RDY/$\overline{\text{BSY}}$ to go high.

The lock bits can only be cleared by executing chip erase.

26.7.12 Reading the Fuse and Lock Bits

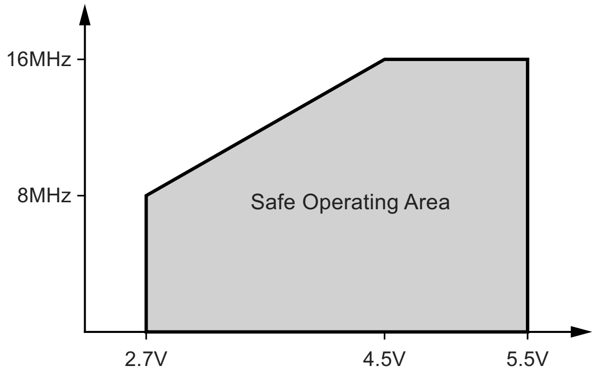
The algorithm for reading the fuse and lock bits is as follows (refer to Section 26.7.4 “Programming the Flash” on page 256 for details on command loading):

1. A: Load command “0000 0100”.
2. Set $\overline{\text{OE}}$ to “0”, BS2 to “0” and BS1 to “0”. The status of the fuse low bits can now be read at DATA (“0” means programmed).
3. Set $\overline{\text{OE}}$ to “0”, BS2 to “1” and BS1 to “1”. The status of the fuse high bits can now be read at DATA (“0” means programmed).
4. Set OE to “0”, BS2 to “1”, and BS1 to “0”. The status of the extended fuse bits can now be read at DATA (“0” means programmed).
5. Set $\overline{\text{OE}}$ to “0”, BS2 to “0” and BS1 to “1”. The status of the lock bits can now be read at DATA (“0” means programmed).
6. Set $\overline{\text{OE}}$ to “1”.

27.3 Speed Grades

Maximum frequency is depending on $V_{CC.}$ As shown in Figure 27-1, the Maximum Frequency versus V_{CC} curve is linear between 2.7V < V_{CC} < 4.5V.

Figure 27-1. Maximum Frequency versus V_{CC}, ATmega169P



27.4 Clock Characteristics

27.4.1 Calibrated Internal RC Oscillator Accuracy

Table 27-1. Calibration Accuracy of Internal RC Oscillator

	Frequency	V_{CC}	Temperature	Calibration Accuracy
Factory calibration	8.0MHz	3V	25°C	±2%
Factory calibration	8.0MHz	2.7V to 5.5V	–40°C - 85°C	±14%

27.4.2 External Clock Drive Waveforms

Figure 27-2. External Clock Drive Waveforms

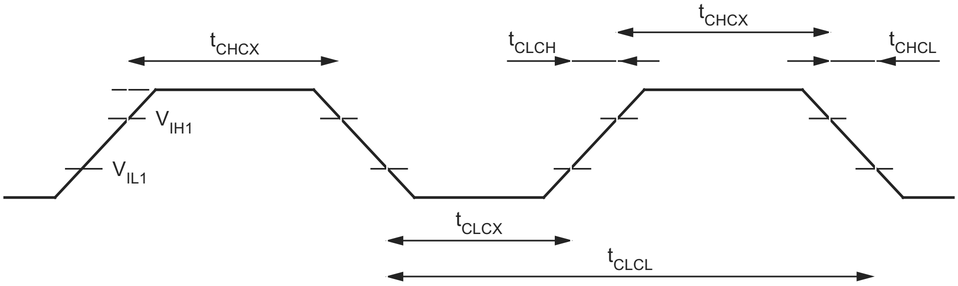


Figure 28-29. BOD Thresholds versus Temperature (BOD Level is 2.7V)

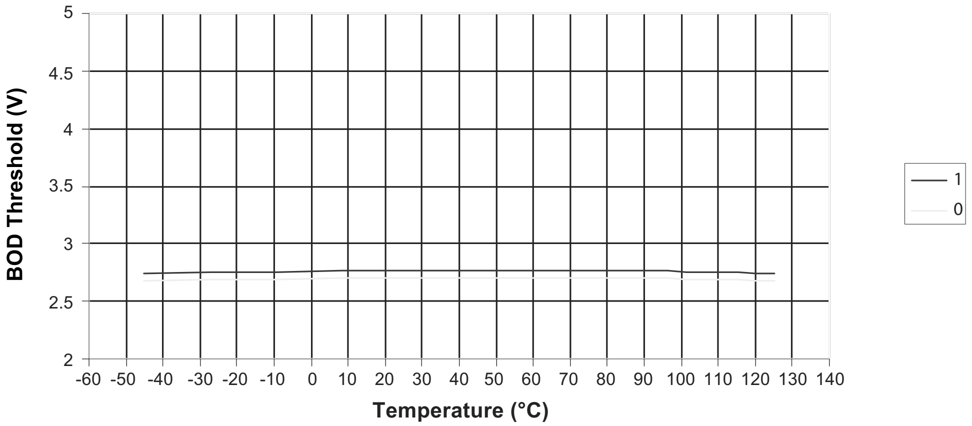


Figure 28-30.Bandgap Voltage versus Temperature

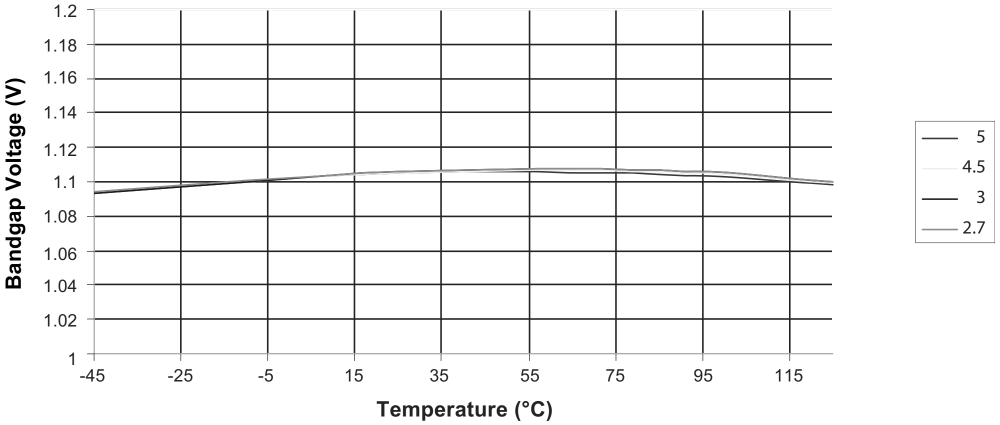
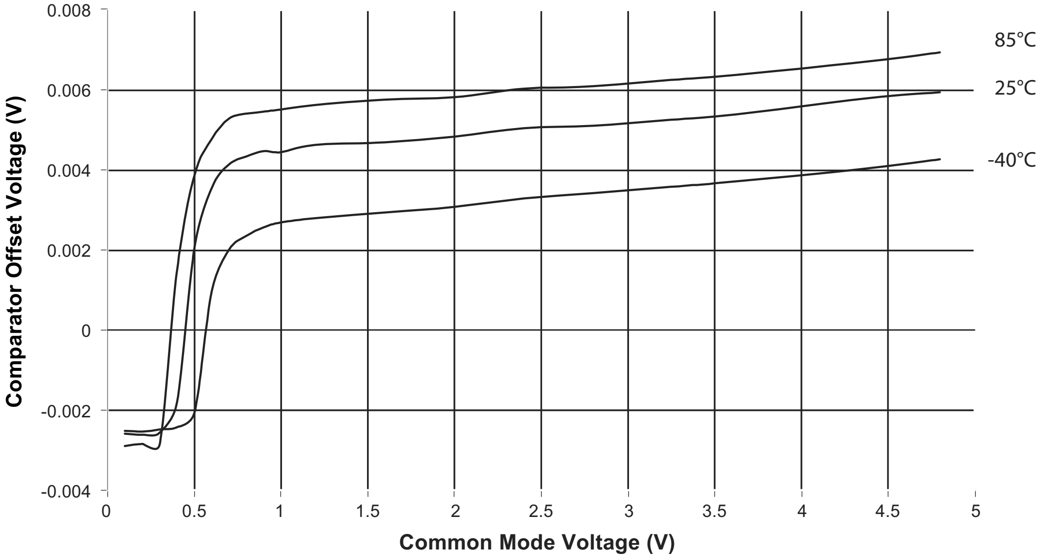


Figure 28-31. Analog Comparator Offset Voltage versus Common Mode Voltage (V_{CC} = 5V)



28.13 Current Consumption in Reset and Reset Pulsewidth

Figure 28-41. Reset Supply Current versus V_{CC} (0.1 - 1.0 MHz, Excluding Current Through The Reset Pull-up)

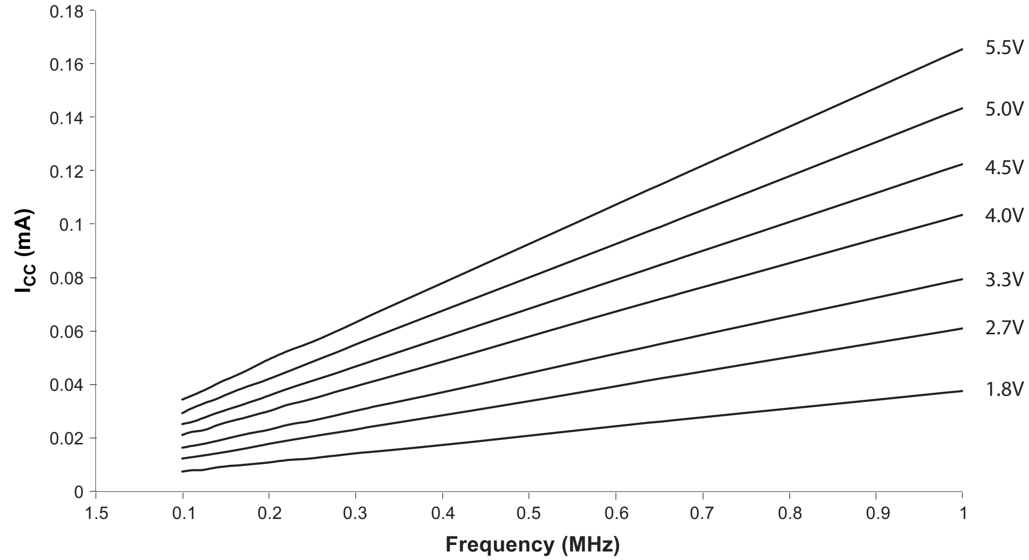
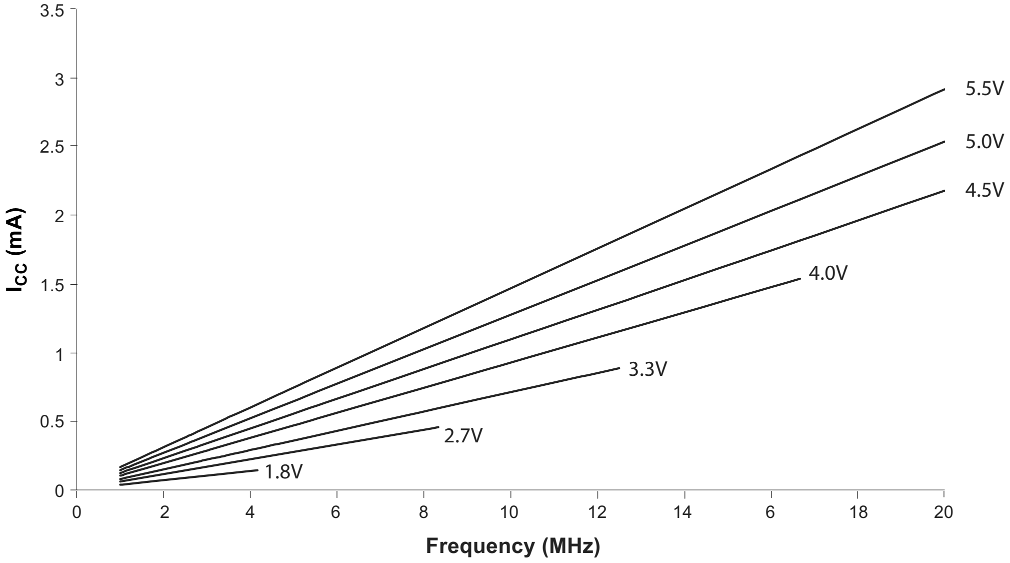


Figure 28-42. Reset Supply Current versus V_{CC} (1 - 16 MHz, Excluding Current Through The Reset Pull-up)

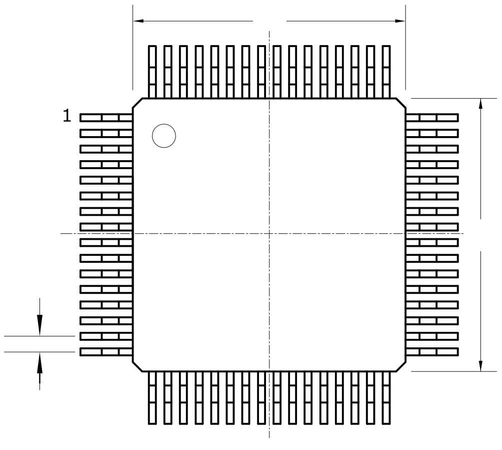


29. Register Summary (Continued)

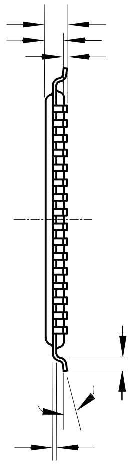
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x01 (0x21)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	74
0x00 (0x20)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	74

Notes:
1. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
2. I/O registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions.
3. Some of the status flags are cleared by writing a logical one to them. Note that, unlike most other AVRs, the CBI and SBI instructions will only operate on the specified bit, and can therefore be used on registers containing such status flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.
4. When using the I/O specific commands IN and OUT, the I/O addresses 0x00 - 0x3F must be used. When addressing I/O registers as data space using LD and ST instructions, 0x20 must be added to these addresses. The ATmega169P is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcode for the IN and OUT instructions. For the extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

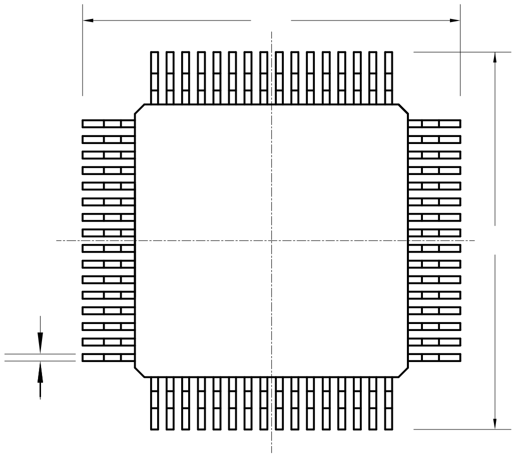
32. Packaging Information



Top View



Side View



Bottom View

COMMON DIMENSIONS

(Unit of Measure = mm)

Symbol	MIN	NOM	MAX	NOTE
A	––––	––––	1.20	
A1	0.05	––––	0.15	
A2	0.95	1.00	1.05	
D/E	15.75	16.00	16.25	
D1/E1	13.90	14.00	14.10	2
C	0.09	––––	0.20	
L	0.45	––––	0.75	
b	0.30	0.37	0.45	
e	0.80 TYP.			
n	44			

03/30/12

	TITLE	GPC	DRAWING NO.	REV.
Atmel **Package Drawing Contact:** packagedrawings@atmel.com	MD, 64 Lds - 0.80mm Pitch, 14x14x1.00mm Body size Thin Profile Plastic Quad Flat Package (TQFP)	ADY	MD	G