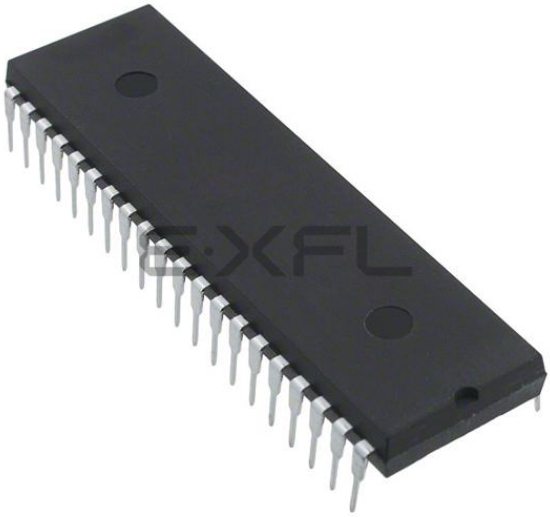


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

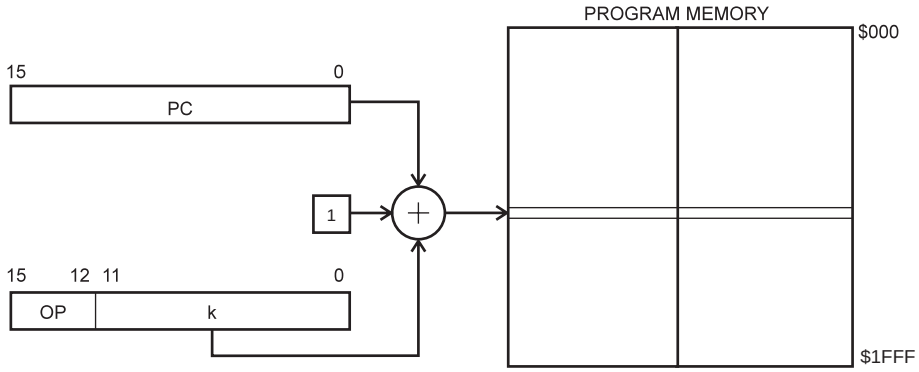
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	Obsolete
Core Processor	AVR
Core Size	8-Bit
Speed	8MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	32
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)	4V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega163-8pi

Relative Program Addressing, RJMP and RCALL

Figure 20. Relative Program Memory Addressing



Program execution continues at address PC + k + 1.
The relative address k is from -2,048 to 2,047.

The EEPROM Data Memory

The ATmega163 contains 512 bytes of data EEPROM memory. It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU is described on page 62 specifying the EEPROM Address Registers, the EEPROM Data Register, and the EEPROM Control Register.

For the SPI data downloading, see page 154 for a detailed description.

Memory Access Times and Instruction Execution Timing

This section describes the general access timing concepts for instruction execution and internal memory access.

The AVR CPU is driven by the System Clock Ø, directly generated from the main Oscillator for the chip. No internal clock division is used.

Figure 21 shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast-access Register File concept. This is the basic pipelining concept to obtain up to 1 MIPS per MHz with the corresponding unique results for functions per cost, functions per clocks, and functions per power-unit.

Figure 21. The Parallel Instruction Fetches and Instruction Executions

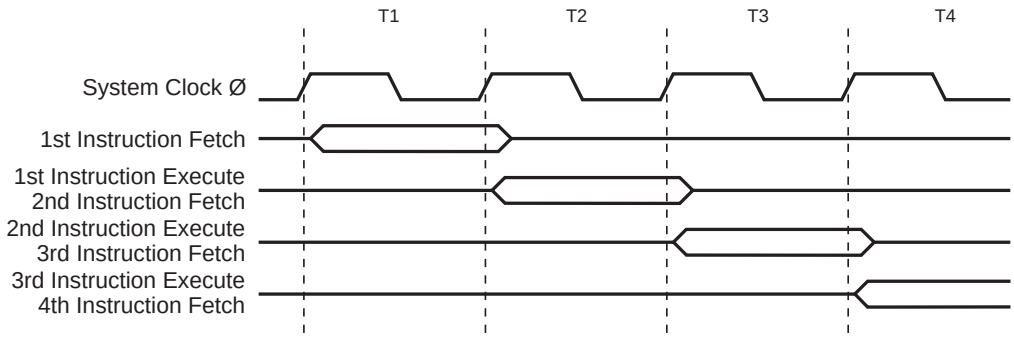


Figure 22 shows the internal timing concept for the Register File. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

When the BOOTRST Fuse is programmed and the Boot section size set to 512 bytes, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega163 is:

```
Address   Labels  Code              Comments
$002              jmp     EXT_INT0   ; IRQ0 Handler
...     ...       ...
$022              jmp     TWI     ; Two-wire Serial Interface Interrupt Handler
;
$024      MAIN:   ldi     r16,high(RAMEND) ; Main program start
$025              out     SPH,r16    ; Set stack pointer to top of RAM
$026              ldi     r16,low(RAMEND)
$027              out     SPL,r16
$028              <instr> xxx
;
.org $1f00
$1f00             jmp     RESET      ; Reset Handler
```

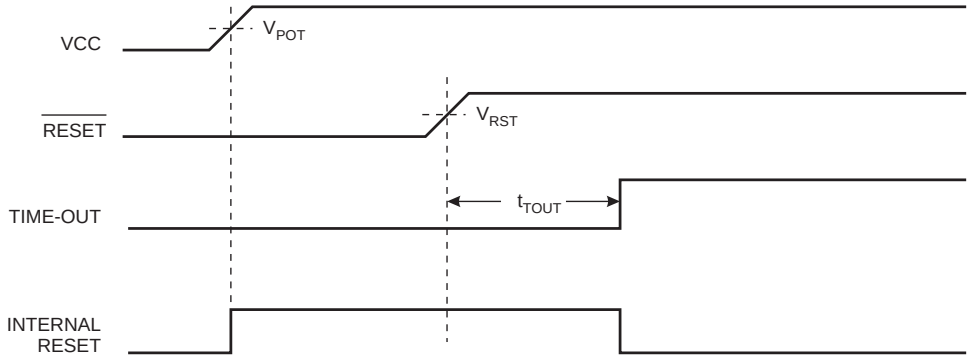
Reset Sources

The ATmega163 has four sources of reset:

- Power-on Reset. The MCU is reset when the supply voltage is below the Power-on Reset threshold (V_{POT}).
- External Reset. The MCU is reset when a low level is present on the $\overline{\text{RESET}}$ pin for more than 500 ns.
- Watchdog Reset. The MCU is reset when the Watchdog Timer period expires and the Watchdog is enabled.
- Brown-out Reset. The MCU is reset when the supply voltage V_{CC} is below the Brown-out Reset threshold (V_{BOT}).

During Reset, all I/O Registers are set to their initial values, and the program starts execution from address $000 (unless the BOOTRST Fuse is programmed, as explained above). The instruction placed in this address location must be a JMP – absolute jump – instruction to the reset handling routine. If the program never enables an interrupt source, the Interrupt Vectors are not used, and regular program code can be placed at these locations. The circuit diagram in Figure 24 shows the Reset Logic. Table 4 and Table 5 define the timing and electrical parameters of the reset circuitry.

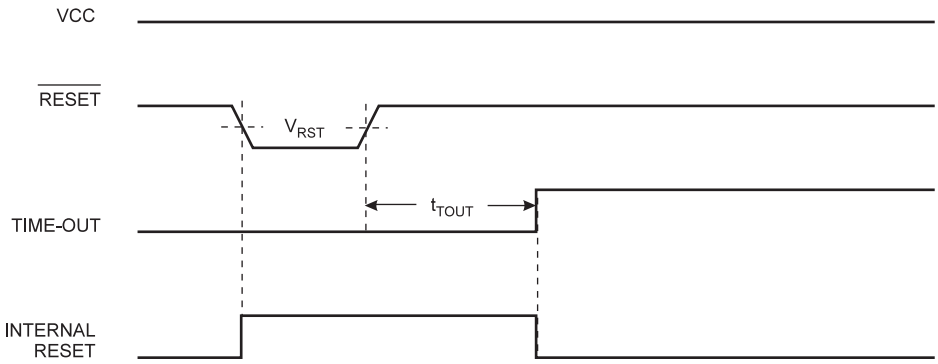
Figure 26. MCU Start-up, $\overline{\text{RESET}}$ Extended Externally



External Reset

An External Reset is generated by a low level on the $\overline{\text{RESET}}$ pin. Reset pulses longer than 500 ns will generate a Reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a Reset. When the applied signal reaches the Reset Threshold Voltage – V_{RST} on its positive edge, the delay timer starts the MCU after the Time-out Period t_{TOUT} has expired.

Figure 27. External Reset During Operation



Brown-out Detection

ATmega163 has an On-chip Brown-out Detection (BOD) circuit for monitoring the V_{CC} level during the operation. The BOD circuit can be enabled/disabled by the fuse BODEN. When the BOD is enabled (BODEN programmed), and V_{CC} decreases to a value below the trigger level, the Brown-out Reset is immediately activated. When V_{CC} increases above the trigger level, the Brown-out Reset is deactivated after a delay. The delay is defined by the user in the same way as the delay of POR signal, in Table 5. The trigger level for the BOD can be selected by the fuse BODLEVEL to be 2.7V (BODLEVEL unprogrammed), or 4.0V (BODLEVEL programmed). The trigger level has a hysteresis of 50 mV to ensure spike free Brown-out Detection.

The BOD circuit will only detect a drop in V_{CC} if the voltage stays below the trigger level for longer than 9 µs for trigger level 4.0V, 21 µs for trigger level 2.7V (typical values).

If one or more interrupt conditions occur when the Global Interrupt Enable bit is cleared (zero), the corresponding interrupt flag(s) will be set and remembered until the Global Interrupt Enable bit is set (one), and will be executed by order of priority.

Note that external level interrupt does not have a flag, and will only be remembered for as long as the interrupt condition is present.

Note that the Status Register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt routine. This must be handled by software.

Interrupt Response Time

The interrupt execution response for all the enabled AVR interrupts is four clock cycles minimum. After four clock cycles the Program Vector address for the actual interrupt handling routine is executed. During this four clock cycle period, the Program Counter (13 bits) is pushed onto the Stack. The vector is normally a jump to the interrupt routine, and this jump takes three clock cycles. If an interrupt occurs during execution of a multi-cycle instruction, this instruction is completed before the interrupt is served. If an interrupt occurs when the MCU is in sleep mode, the interrupt execution response time is increased by four clock cycles.

A return from an interrupt handling routine takes four clock cycles. During these four clock cycles, the Program Counter (two bytes) is popped back from the Stack, the Stack Pointer is incremented by two, and the I Flag in SREG is set. When *AVR* exits from an interrupt, it will always return to the main program and execute one more instruction before any pending interrupt is served.

The General Interrupt Mask Register – GIMSK

Bit	7	6	5	4	3	2	1	0	
$3B ($5B)	**INT1**	**INT0**	–	–	–	–	–	–	**GIMSK**
Read/Write	R/W	R/W	R	R	R	R	R	R	
Initial Value	0	0	x	0	0	0	0	0	

- **Bit 7 – INT1: External Interrupt Request 1 Enable**

When the INT1 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is activated. The Interrupt Sense Control1 bits 1/0 (ISC11 and ISC10) in the MCU general Control Register (MCUCR) define whether the external interrupt is activated on rising and/or falling edge of the INT1 pin or level sensed. Activity on the pin will cause an interrupt request even if INT1 is configured as an output. The corresponding interrupt of External Interrupt Request 1 is executed from program memory address $004. See also “External Interrupts”.

- **Bit 6 – INT0: External Interrupt Request 0 Enable**

When the INT0 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is activated. The Interrupt Sense Control0 bits 1/0 (ISC01 and ISC00) in the MCU General Control Register (MCUCR) define whether the external interrupt is activated on rising or falling edge of the INT0 pin or level sensed. Activity on the pin will cause an interrupt request even if INT0 is configured as an output. The corresponding interrupt of External Interrupt Request 0 is executed from Program Memory address $002. See also “External Interrupts.”

- **Bits 5 – Res: Reserved Bits**

This bit is reserved in the ATmega163 and the read value is undefined.

- **Bits 4..0 – Res: Reserved Bits**

These bits are reserved bits in the ATmega163 and always read as zero.

The General Interrupt Flag Register – GIFR

Bit	7	6	5	4	3	2	1	0	
$3A ($5A)	INTF1	INTF0	–	–	–	–	–	–	GIFR
Read/Write	R/W	R/W	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – INTF1: External Interrupt Flag1**

When an edge on the INT1 pin triggers an interrupt request, the correspnding Interrupt Flag, INTF1, becomes set (one). If the I-bit in SREG and the corresponding Interrupt Enable bit, INT1 in GIMSK are set (one), the MCU will jump to the 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 INT1 is configured as a level interrupt.

- **Bit 6 – INTF0: External Interrupt Flag0**

When an event on the INT0 pin triggers an interrupt request, the corresponding Interrupt Flag, INTF0 becomes set (one). If the I-bit in SREG and the corresponding Interrupt Enable bit, INT0 in GIMSK are set (one), the MCU will jump to the 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.

- **Bits 5..0 – Res: Reserved Bits**

These bits are reserved bits in the ATmega163 and always read as zero.

The Timer/Counter Interrupt Mask Register – TIMSK

Bit	7	6	5	4	3	2	1	0	
$39 ($59)	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	–	TOIE0	TIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – OCIE2: Timer/Counter2 Output Compare Match Interrupt Enable**

When the OCIE2 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter2 Compare Match Interrupt is enabled. The corresponding interrupt (at vector $006) is executed if a Compare Match in Timer/Counter2 occurs, i.e., when the OCF2 bit is set in the Timer/Counter Interrupt Flag Register – TIFR.

- **Bit 6 – TOIE2: Timer/Counter2 Overflow Interrupt Enable**

When the TOIE2 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter2 Overflow Interrupt is enabled. The corresponding interrupt (at vector $008) is executed if an overflow in Timer/Counter2 occurs, i.e., when the TOV2 bit is set in the Timer/Counter Interrupt Flag Register – TIFR.

- **Bit 5 – TICIE1: Timer/Counter1 Input Capture Interrupt Enable**

When the TICIE1 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Input Capture Event Interrupt is enabled. The corresponding interrupt

TCNT1 Timer/Counter1 Write

When the CPU writes to the high byte TCNT1H, the written data is placed in the TEMP Register. Next, when the CPU writes the low byte TCNT1L, this byte of data is combined with the byte data in the TEMP Register, and all 16 bits are written to the TCNT1 Timer/Counter1 Register simultaneously. Consequently, the high byte TCNT1H must be accessed first for a full 16-bit register write operation.

TCNT1 Timer/Counter1 Read

When the CPU reads the low byte TCNT1L, the data of the Low Byte TCNT1L is sent to the CPU and the data of the High Byte TCNT1H is placed in the TEMP Register. When the CPU reads the data in the High Byte TCNT1H, the CPU receives the data in the TEMP Register. Consequently, the Low Byte TCNT1L must be accessed first for a full 16-bit register read operation.

The Timer/Counter1 is realized as an up or up/down (in PWM mode) counter with read and write access. If Timer/Counter1 is written to and a clock source is selected, the Timer/Counter1 continues counting in the timer clock cycle after it is preset with the written value.

Timer/Counter1 Output Compare Register – OCR1AH and OCR1AL

Bit	15	14	13	12	11	10	9	8	
$2B ($4B)	**MSB**								**OCR1AH**
$2A ($4A)								**LSB**	**OCR1AL**
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	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	
	0	0	0	0	0	0	0	0	

Timer/Counter1 Output Compare Register – OCR1BH and OCR1BL

Bit	15	14	13	12	11	10	9	8	
$29 ($49)	**MSB**								**OCR1BH**
$28 ($48)								**LSB**	**OCR1BL**
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	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	
	0	0	0	0	0	0	0	0	

The Output Compare Registers are 16-bit read/write registers.

The Timer/Counter1 Output Compare Registers contain the data to be continuously compared with Timer/Counter1. Actions on compare matches are specified in the Timer/Counter1 Control and Status Register. A software write to the Timer/Counter Register blocks compare matches in the next Timer/Counter clock cycle. This prevents immediate interrupts when initializing the Timer/Counter.

A Compare Match will set the compare interrupt flag in the CPU clock cycle following the compare event.

Since the Output Compare Registers – OCR1A and OCR1B – are 16-bit registers, a temporary register TEMP is used when OCR1A/B are written to ensure that both bytes are updated simultaneously. When the CPU writes the high byte, OCR1AH or OCR1BH, the data is temporarily stored in the TEMP Register. When the CPU writes the Low Byte, OCR1AL or OCR1BL, the TEMP Register is simultaneously written to OCR1AH or OCR1BH. Consequently, the high byte OCR1AH or OCR1BH must be written first for a full 16-bit register write operation.

Timer/Counter2 Output Compare Register – OCR2

Bit	7	6	5	4	3	2	1	0	
$23 ($43)	MSB							LSB	OCR2
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 is an 8-bit read/write register.

The Timer/Counter Output Compare Register contains the data to be continuously compared with Timer/Counter2. Actions on compare matches are specified in TCCR2. A software write to the Timer/Counter2 Register blocks compare matches in the next Timer/Counter2 clock cycle. This prevents immediate interrupts when initializing the Timer/Counter2.

A Compare Match will set the Compare Interrupt Flag in the CPU clock cycle following the compare event.

Timer/Counter2 in PWM Mode

When PWM mode is selected, the Timer/Counter2 either wraps (overflows) when it reaches $FF or it acts as an up/down counter.

If the up/down mode is selected, the Timer/Counter2 and the Output Compare Register – OCR2 form an 8-bit, free-running, glitch-free, and phase correct PWM with outputs on the PD7(OC2) pin.

If the overflow mode is selected, the Timer/Counter2 and the Output Compare Register – OCR2 form an 8-bit, free-running, and glitch-free PWM, operating with twice the speed of the up/down counting mode.

PWM Modes (Up/Down and Overflow)

The two different PWM modes are selected by the CTC2 bit in the Timer/Counter Control Register – TCCR2.

If CTC2 is cleared and PWM mode is selected, the Timer/Counter acts as an up/down counter, counting up from $00 to $FF, where it turns and counts down again to zero before the cycle is repeated. When the counter value matches the contents of the Output Compare Register, the PD7(OC2) pin is set or cleared according to the settings of the COM21/COM20 bits in the Timer/Counter Control Register TCCR2.

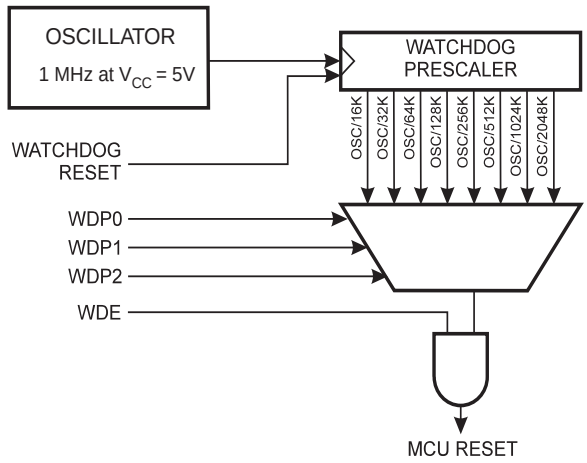
If CTC2 is set and PWM mode is selected, the Timer/Counters will wrap and start counting from $00 after reaching $FF. The PD7(OC2) pin will be set or cleared according to the settings of COM21/COM20 on a Timer/Counter overflow or when the counter value matches the contents of the Output Compare Register. Refer to Table 21 for details.

Watchdog Timer

The Watchdog Timer is clocked from a separate On-chip Oscillator which runs at 1 MHz. This is the typical value at V_{CC} = 5V. See characterization data for typical values at other V_{CC} levels. By controlling the Watchdog Timer prescaler, the Watchdog Reset interval can be adjusted as shown in Table 23 on page 61. The WDR – Watchdog Reset – instruction resets the Watchdog Timer. Eight different clock cycle periods can be selected to determine the reset period. If the reset period expires without another Watchdog Reset, the ATmega163 resets and executes from the Reset Vector. For timing details on the Watchdog Reset, refer to page 28.

To prevent unintentional disabling of the Watchdog, a special turn-off sequence must be followed when the Watchdog is disabled. Refer to the description of the Watchdog Timer Control Register for details.

Figure 40. Watchdog Timer



The Watchdog Timer Control Register – WDTCR

Bit	7	6	5	4	3	2	1	0	
$21 ($41)	–	–	–	WDTOE	WDE	WDP2	WDP1	WDP0	WDTCR
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..5 – Res: Reserved Bits**

These bits are reserved bits in the ATmega163 and will always read as zero.

- **Bit 4 – WDTOE: Watchdog Turn-off Enable**

This bit must be set (one) when the WDE bit is cleared. Otherwise, the Watchdog will not be disabled. Once set, hardware will clear this bit to zero after four clock cycles. Refer to the description of the WDE bit for a Watchdog disable procedure.

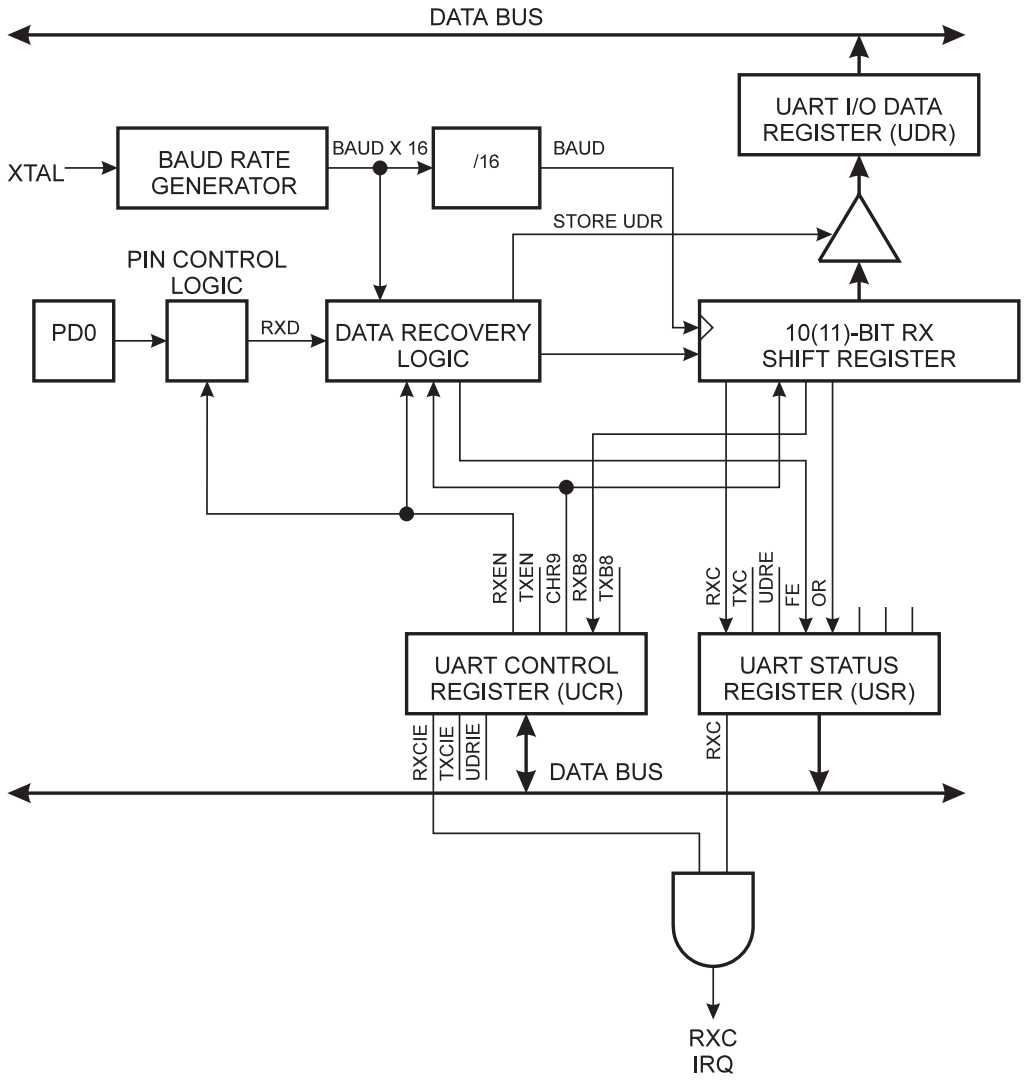
- **Bit 3 – WDE: Watchdog Enable**

When the WDE is set (one) the Watchdog Timer is enabled, and if the WDE is cleared (zero) the Watchdog Timer function is disabled. WDE can only be cleared if the WDTOE bit is set(one). To disable an enabled Watchdog Timer, the following procedure must be followed:

Data Reception

Figure 46 shows a block diagram of the UART Receiver

Figure 46. UART Receiver



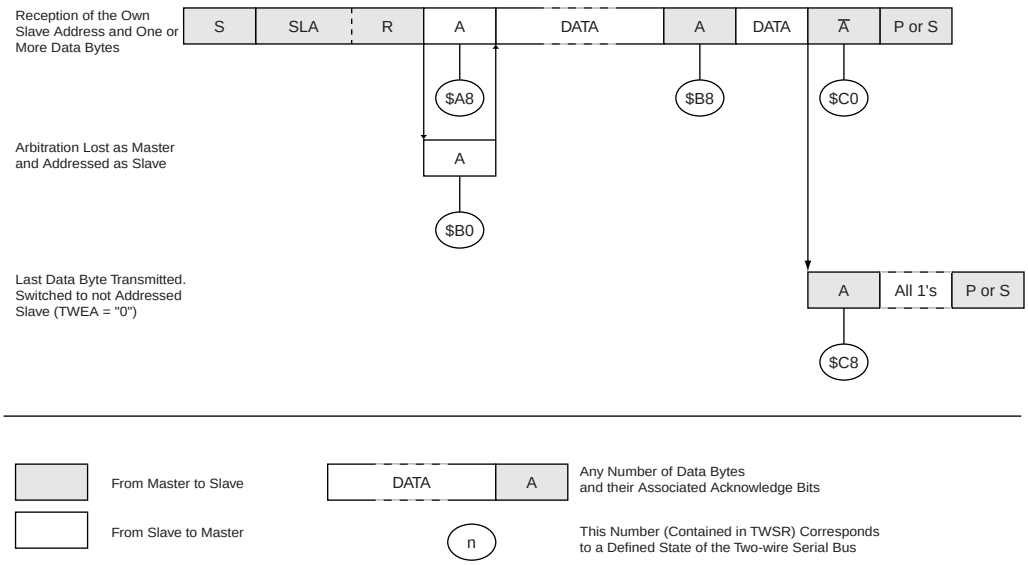
The Receiver front-end logic samples the signal on the RXD pin at a frequency 16 times the baud rate. While the line is idle, one single sample of logical zero will be interpreted as the falling edge of a start bit, and the start bit detection sequence is initiated. Let sample 1 denote the first zero-sample. Following the 1 to 0-transition, the receiver samples the RXD pin at samples 8, 9, and 10. If two or more of these three samples are found to be logical ones, the start bit is rejected as a noise spike and the receiver starts looking for the next 1 to 0-transition.

If however, a valid start bit is detected, sampling of the data bits following the start bit is performed. These bits are also sampled at samples 8, 9, and 10. The logical value found in at least two of the three samples is taken as the bit value. All bits are shifted into the Transmitter Shift Register as they are sampled. Sampling of an incoming character is shown in Figure 47. Note that the description above is not valid when the UART transmission speed is doubled. See “Double Speed Transmission” on page 78 for a detailed description.

Table 35. Status Codes for Slave Transmitter Mode

Status Code (TWSR)	Status of the Two-wire Serial Bus and Two-wire Serial Interface hardware	Application Software Response					Next Action Taken by Two-wire Serial Interface Hardware
		To/from TWDR	To TWCR				
			STA	STO	TWINT	TWEA	
$A8	Own SLA+R has been received; ACK has been returned	Load data byte or	X	0	1	0	Last data byte will be transmitted and NOT ACK should be received
		Load data byte	X	0	1	1	Data byte will be transmitted and ACK should be received
$B0	Arbitration lost in SLA+R/W as master; own SLA+R has been received; ACK has been returned	Load data byte or	X	0	1	0	Last data byte will be transmitted and NOT ACK should be received
		Load data byte	X	0	1	1	Data byte will be transmitted and ACK should be received
$B8	Data byte in TWDR has been transmitted; ACK has been received	Load data byte or	X	0	1	0	Last data byte will be transmitted and NOT ACK should be received
		Load data byte	X	0	1	1	Data byte will be transmitted and ACK should be received
$C0	Data byte in TWDR has been transmitted; NOT ACK has been received	No TWDR action or	0	0	1	0	Switched to the not addressed Slave mode; no recognition of own SLA or GCA
		No TWDR action or	0	0	1	1	Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = “1”
		No TWDR action or	1	0	1	0	Switched to the not addressed Slave mode; no recognition of own SLA or GCA; a START condition will be transmitted when the bus becomes free
		No TWDR action	1	0	1	1	Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = “1”; a START condition will be transmitted when the bus becomes free
$C8	Last data byte in TWDR has been transmitted (TWEA = “0”); ACK has been received	No TWDR action or	0	0	1	0	Switched to the not addressed Slave mode; no recognition of own SLA or GCA
		No TWDR action or	0	0	1	1	Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = “1”
		No TWDR action or	1	0	1	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA; a START condition will be transmitted when the bus becomes free
		No TWDR action	1	0	1	1	Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = “1”; a START condition will be transmitted when the bus becomes free

Figure 55. Formats and States in the Slave Transmitter Mode



Port B

Port B is an 8-bit bi-directional I/O port with internal pull-ups.

Three I/O memory address locations are allocated for Port B, one each for the Data Register – PORTB, $18($38), Data Direction Register – DDRB, $17($37) and the Port B Input Pins – PINB, $16($36). The Port B Input Pins address is read only, while the Data Register and the Data Direction Register are read/write.

All port pins have individually selectable pull-up resistors. The Port B output buffers can sink 20 mA and thus drive LED displays directly. When pins PB0 to PB7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated.

The Port B pins with alternate functions are shown in Table 45.

Table 45. Port B Pins Alternate Functions

Port Pin	Alternate Functions
PB0	T0 (Timer/Counter0 External Counter Input)
PB1	T1 (Timer/Counter1 External Counter Input)
PB2	AIN0 (Analog Comparator Positive Input)
PB3	AIN1 (Analog Comparator Negative Input)
PB4	$\overline{SS}$ (SPI Slave Select Input)
PB5	MOSI (SPI Bus Master Output/Slave Input)
PB6	MISO (SPI Bus Master Input/Slave Output)
PB7	SCK (SPI Bus Serial Clock)

When the pins are used for the alternate function, the DDRB and PORTB Registers have to be set according to the alternate function description.

The Port B Data Register – PORTB

Bit	7	6	5	4	3	2	1	0	
$18 ($38)	**PORTB7**	**PORTB6**	**PORTB5**	**PORTB4**	**PORTB3**	**PORTB2**	**PORTB1**	**PORTB0**	**PORTB**
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 Port B Data Direction Register – DDRB

Bit	7	6	5	4	3	2	1	0	
$17 ($37)	**DDB7**	**DDB6**	**DDB5**	**DDB4**	**DDB3**	**DDB2**	**DDB1**	**DDB0**	**DDRB**
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 Port B Input Pins Address – PINB

Bit	7	6	5	4	3	2	1	0	
$16 ($36)	**PINB7**	**PINB6**	**PINB5**	**PINB4**	**PINB3**	**PINB2**	**PINB1**	**PINB0**	**PINB**
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

The Port B Input Pins Address – PINB – is not a register, and this address enables access to the physical value on each Port B pin. When reading PORTB, the PORTB Data Latch is read, and when reading PINB, the logical values present on the pins are read.

Entering the Boot Loader Program

The SPM instruction can access the entire Flash, but can only be executed from the Boot Loader Flash section. If no Boot Loader capability is needed, the entire Flash is available for application code. Entering the Boot Loader takes place by a jump or call from the application program. This may be initiated by some trigger such as a command received via UART or SPI interface, for example. 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 start executing the application code. Note that the fuses cannot be changed by the MCU itself. This means that once the Boot Reset Fuse is programmed, the Reset Vector will always point to the Boot Loader Reset and the fuse can only be changed through the serial or parallel programming interface.

Table 52. Boot Reset Fuse

BOOTRST	Reset Address
1	Reset Vector = Application Reset (address $0000)
0	Reset Vector = Boot Loader Reset (see Table 51)

Capabilities of the Boot Loader

The program code within the Boot Loader section has the capability to read from and write into the entire Flash, including the Boot Loader memory. This allows the user to update both the Application code and the Boot Loader code that handles the software update. The Boot Loader can thus even modify itself, and it can also erase itself from the code if the feature is not needed anymore.

Self-Programming the Flash

Programming of the Flash is executed one page at a time. The Flash page must be erased first for correct programming. The general Write Lock (Lock bit 2) does not control the programming of the Flash memory by SPM instruction. Similarly, the general Read/Write Lock (Lock bit 1) does not control reading nor writing by LPM/SPM, if it is attempted.

The Program memory can only be updated page by page, not word by word. One page is 128 bytes (64 words). The Program memory will be modified by first performing Page Erase, then filling the temporary page buffer one word at a time using SPM, and then executing Page Write. If only part of the page needs to be changed, the other parts must be stored (for example in internal SRAM) before the erase, and then be rewritten. The temporary page buffer can be accessed in a random sequence. It is essential that the page address used in both the Page Erase and Page Write operation is addressing the same page. See “Assembly code example for a Boot Loader” on page 141 for an assembly code example.

See Table 60 on page 156 for typical programming times when using Self-Programming.

Performing Page Erase by SPM

To execute Page Erase, set up the address in the Z-pointer, write “00011” to the five LSB in SPMCR and execute SPM within four clock cycles after writing SPMCR. The data in R1 and R0 is ignored. The page address must be written to Z13:Z7. Other bits in the Z-pointer will be ignored during this operation. It is recommended that the interrupts are disabled during the page erase operation.

Fill the Temporary Buffer (Page Load)

To write an instruction word, set up the address in the Z-pointer and data in R1:R0, write “00001” to the five LSB in SPMCR and execute SPM within four clock cycles after writing SPMCR. The content of Z6:Z1 is used to address the data in the temporary buffer. Z13:Z7 must point to the page that is supposed to be written.

Store Program Memory Control Register – SPMCR

The Store Program Memory Control Register contains the control bits needed to control the programming of the Flash from internal code execution.

Bit	7	6	5	4	3	2	1	0	
$37 ($57)	–	ASB	–	ASRE	BLBSET	PGWRT	PGERS	SPMEN	SPMCR
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	x	0	0	0	0	0	0	0	

- **Bit 7 – Res: Reserved Bit**

This bit is a reserved bit in the ATmega163 and always reads as zero. This bit should be written to zero when writing SPMCR.

- **Bit 6 – ASB: Application Section Busy**

Before entering the Application section after a Boot Loader operation (Page Erase or Page Write) the user software must verify that this bit is cleared. In future devices, this bit will be set to “1” by Page Erase and Page Write. In ATmega163, this bit always reads as zero.

- **Bit 5 – Res: Reserved Bit**

This bit is a reserved bit in the ATmega163 and always reads as zero. This bit should be written to zero when writing SPMCR.

- **Bit 4 – ASRE: Application Section Read Enable**

Before re-entering the Application section, the user software must set this bit together with the SPMEN bit and execute SPM within four clock cycles.

- **Bit 3 – BLBSET: Boot Lock Bit Set**

If this bit is set at the same time as SPMEN, the next SPM instruction within four clock cycles will set Boot Lock bits. Alternatively, an LPM instruction within five cycles will read either the Lock bBits or the Fuse bits. The BLBSET bit will auto-clear upon completion of the SPM or LPM instruction, or if no SPM, or LPM, instruction is executed within four, respectively five, clock cycles.

- **Bit 2 – PGWRT: Page Write**

If this bit is set at the same time as SPMEN, the next SPM instruction within four clock cycles executes Page Write, with the data stored in the temporary buffer. The page address is taken from the high part of the Z-pointer. The data in R1 and R0 are ignored. The PGWRT bit will auto-clear upon completion of a page write, or if no SPM instruction is executed within four clock cycles. The CPU is halted during the entire Page Write operation.

- **Bit 1 – PGERS: Page Erase**

If this bit is set at the same time as SPMEN, the next SPM instruction within four clock cycles executes Page Erase. The page address is taken from the high part of the Z-pointer. The data in R1 and R0 are ignored. The PGERS bit will auto-clear upon completion of a Page Erase, or if no SPM instruction is executed within four clock cycles. The CPU is halted during the entire Page Erase operation.

Parallel Programming Characteristics

Figure 85. Parallel Programming Timing

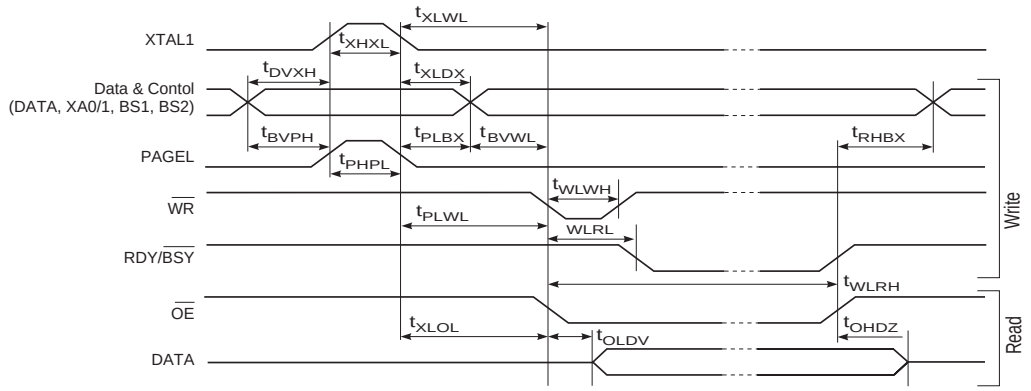
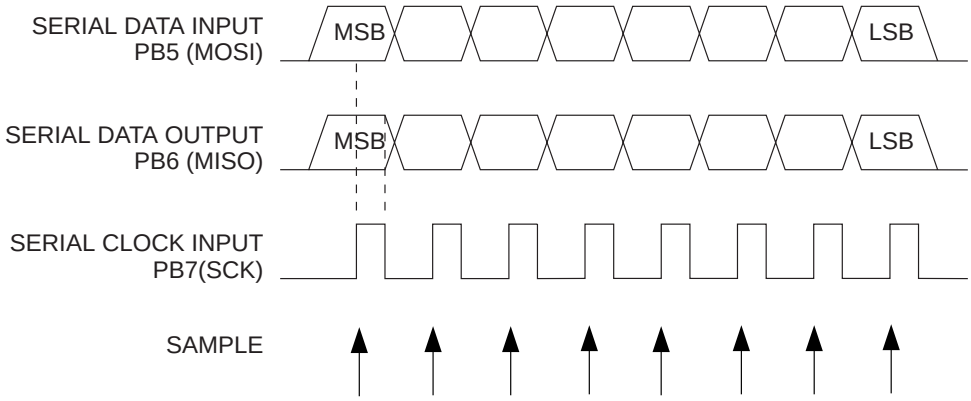


Table 59. Parallel Programming Characteristics, T_A = 25°C ± 10%, V_{CC} = 5 V ± 10%

Symbol	Parameter	Min	Typ	Max	Units
V_{PP}	Programming Enable Voltage	11.5		12.5	V
I_{PP}	Programming Enable Current			250	µA
t_{DVXH}	Data and Control Valid before XTAL1 High	67			ns
t_{XHXL}	XTAL1 Pulse Width High	67			ns
t_{XLDX}	Data and Control Hold after XTAL1 Low	67			ns
t_{XLWL}	XTAL1 Low to $\overline{WR}$ Low	67			ns
t_{BVPH}	BS1 Valid before PAGEL High	67			ns
t_{PHPL}	PAGEL Pulse Width High	67			ns
t_{PLBX}	BS1 Hold after PAGEL Low	67			ns
t_{PLWL}	PAGEL Low to $\overline{WR}$ Low	67			ns
t_{BVWL}	BS1 Valid to $\overline{WR}$ Low	67			ns
t_{RHBX}	BS1 Hold after RDY/$\overline{BSY}$ High	67			ns
t_{WLWH}	$\overline{WR}$ Pulse Width Low	67			ns
t_{WLRL}	$\overline{WR}$ Low to RDY/$\overline{BSY}$ Low	0		2.5	µs
t_{WLRH}	$\overline{WR}$ Low to RDY/$\overline{BSY}$ High(1)	1	1.5	1.9	ms
t_{WLRH_CE}	$\overline{WR}$ Low to RDY/$\overline{BSY}$ High for Chip Erase(2)	16	23	30	ms
t_{WLRH_FLASH}	$\overline{WR}$ Low to RDY/$\overline{BSY}$ High for Write Flash(3)	8	12	15	ms
t_{XLOL}	XTAL1 Low to $\overline{OE}$ Low	67			ns
t_{OLDV}	$\overline{OE}$ Low to DATA Valid		20		ns
t_{OHDZ}	$\overline{OE}$ High to DATA Tri-stated			20	ns

Notes: 1. t_{WLRH} is valid for the Write EEPROM, Write Fuse Bits and Write Lock Bits commands.
2. t_{WLRH_CE} is valid for the Chip Erase command.
3. t_{WLRH_FLASH} is valid for the Write Flash command.

Figure 87. Serial Programming Waveforms



Serial Programming Characteristics

Figure 88. Serial Programming Timing

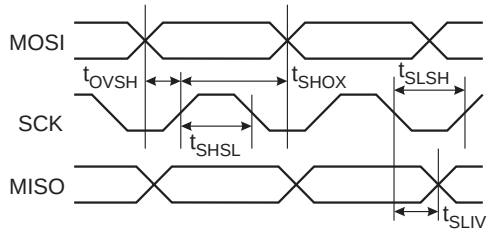
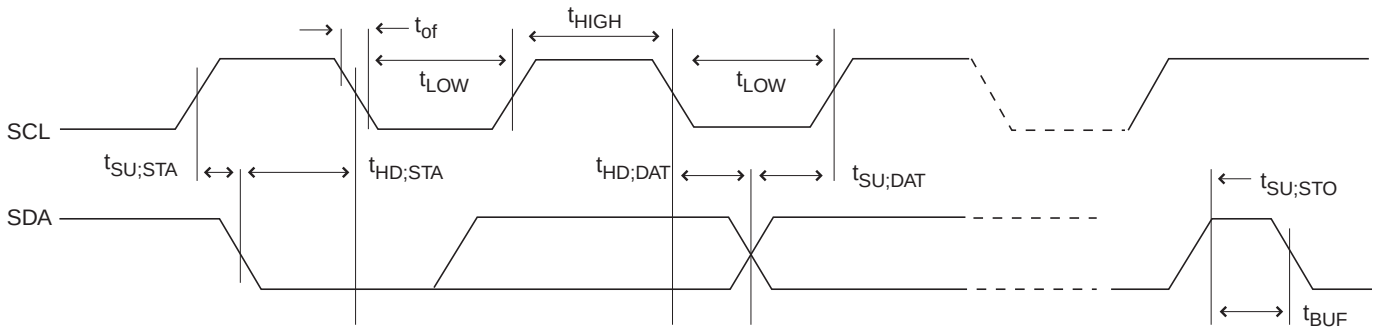


Table 62. Serial Programming Characteristics, T_A = -40°C to 85°C, V_{CC} = 2.7V - 5.5V (Unless otherwise noted)

Symbol	Parameter	Min	Typ	Max	Units
$1/t_{CLCL}$	Oscillator Frequency (V_{CC} = 2.7 - 5.5 V)	0		4	MHz
t_{CLCL}	Oscillator Period (V_{CC} = 2.7 - 5.5 V)	250			ns
$1/t_{CLCL}$	Oscillator Frequency (V_{CC} = 4.0 - 5.5 V)	0		8	MHz
t_{CLCL}	Oscillator Period (V_{CC} = 4.0 - 5.5 V)	125			ns
t_{SHSL}	SCK Pulse Width High	2 t_{CLCL}			ns
t_{SLSH}	SCK Pulse Width Low	2 t_{CLCL}			ns
t_{OVSH}	MOSI Setup to SCK High	t_{CLCL}			ns
t_{SHOX}	MOSI Hold after SCK High	2 t_{CLCL}			ns
t_{SLIV}	SCK Low to MISO Valid	10	16	32	ns

Figure 90. Two-wire Serial Bus Timing



Typical Characteristics

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. All pins on Port F are pulled high externally. A sine wave generator with rail-to-rail output is used as clock source.

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 91. Analog Comparator Offset Voltage vs, Common Mode Voltage (V_{CC} = 5V)

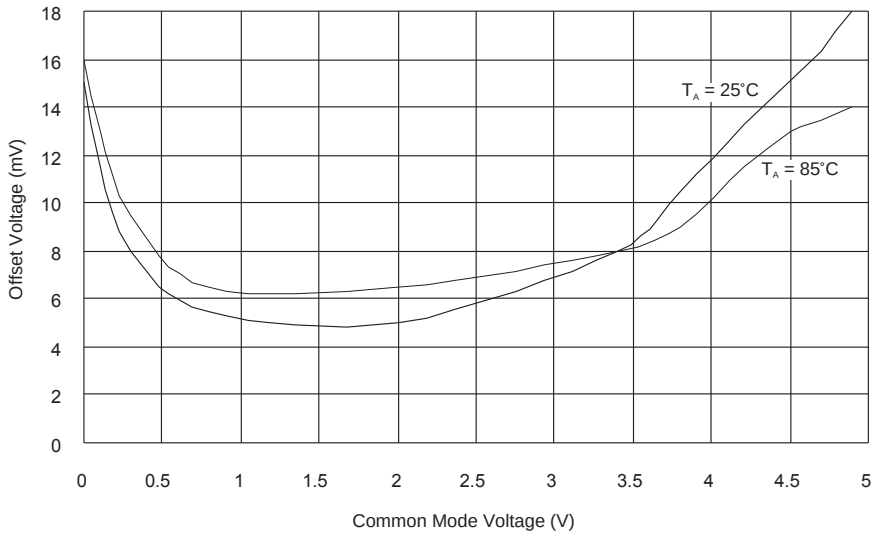


Figure 92. Analog Comparator Offset Voltage vs. Common Mode Voltage (V_{CC} = 2.7V)

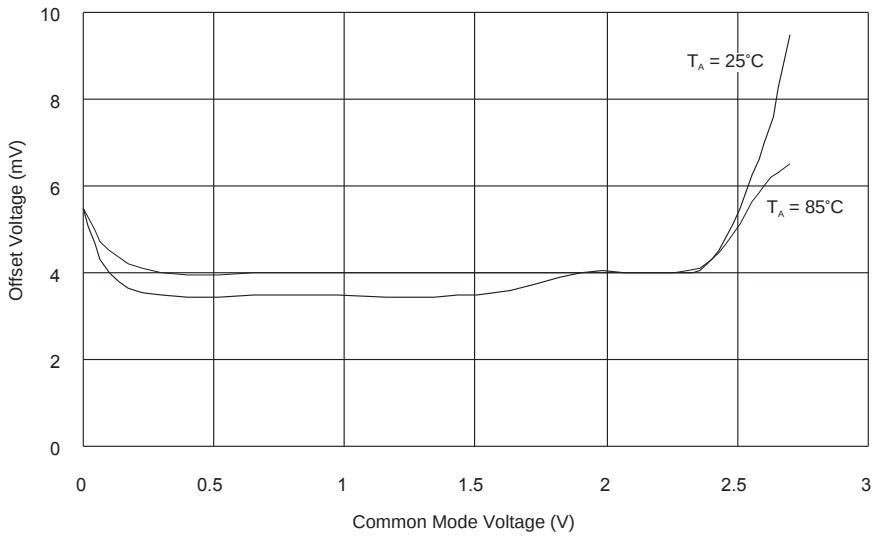


Figure 93. Analog Comparator Input Leakage Current (V_{CC} = 6V; TA = 25°C)

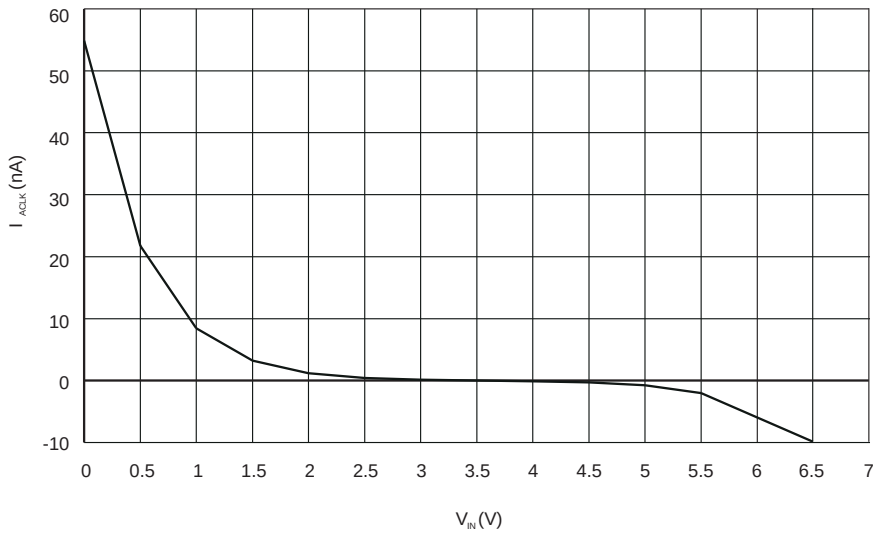


Figure 98. I/O Pin Source Current vs. Output Voltage (V_{CC} = 5V)

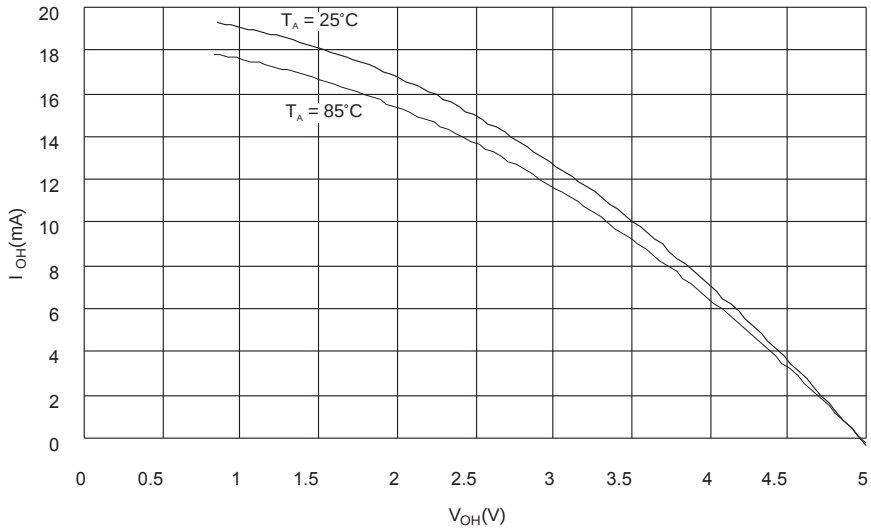


Figure 99. I/O Pin Sink Current vs. Output Voltage (V_{CC} = 2.7V)