Microchip Technology - ATMEGA163-8AC Datasheet

Welcome to E-XFL.COM

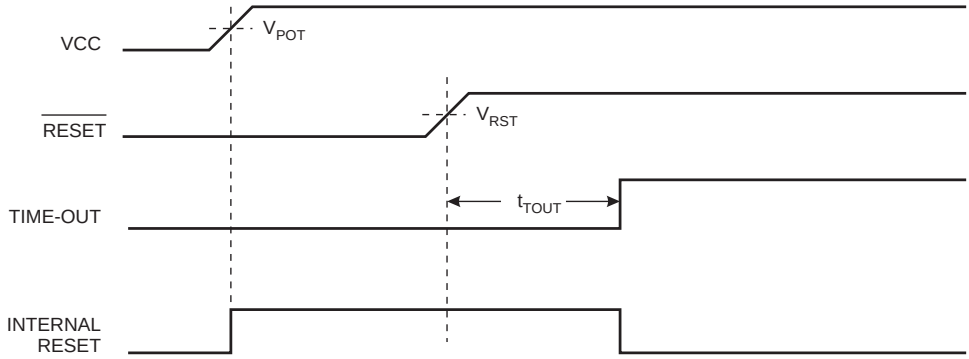
What is "Embedded - Microcontrollers"?

"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	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	0°C ~ 70°C
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega163-8ac

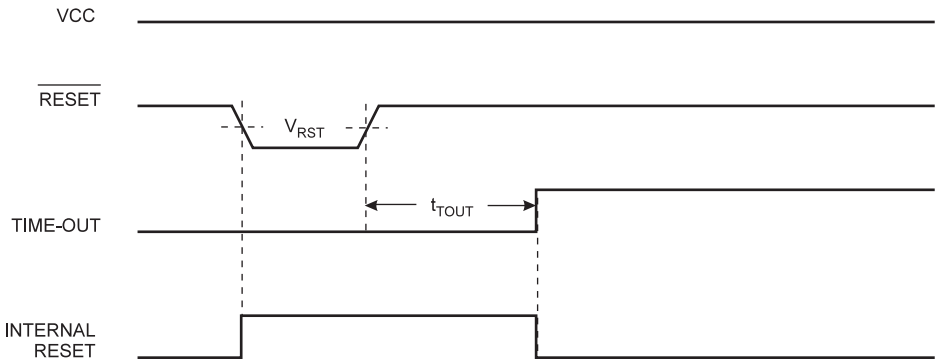
Figure 26. MCU Start-up, $\overline{\text{RESET}}$ Extended Externally



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.

- **Bit 1, 0 – ISC01, ISC00: Interrupt Sense Control 0 Bit 1 and Bit 0**

The External Interrupt 0 is activated by the external pin INT0 if the SREG I-Flag and the corresponding interrupt mask are set. The level and edges on the external INT0 pin that activate the interrupt are defined in Table 9. The value on the INT0 pin is sampled before detecting edges. If edge or toggle interrupt is selected, pulses that last longer than one clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt.

Table 9. Interrupt 0 Sense Control

ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request.
0	1	Any logical change on INT0 generates an interrupt request.
1	0	The falling edge of INT0 generates an interrupt request.
1	1	The rising edge of INT0 generates an interrupt request.

Sleep Modes

To enter any of the four sleep modes, the SE bit in MCUCR must be set (one) and a SLEEP instruction must be executed. The SM1 and SM0 bits in the MCUCR Register select which sleep mode (Idle, ADC Noise Reduction, Power-down, or Power-save) will be activated by the SLEEP instruction. See Table 7 for a summary. If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU wakes up. The MCU is then halted for four cycles, executes the interrupt routine, and resumes execution from the instruction following SLEEP. The contents of the Register File, SRAM, and I/O memory are unaltered when the device wakes up from sleep. If a Reset occurs during sleep mode, the MCU wakes up and executes from the Reset Vector.

Idle Mode

When the SM1/SM0 bits are set to 00, the SLEEP instruction makes the MCU enter Idle mode, stopping the CPU but allowing SPI, UART, Analog Comparator, ADC, Two-wire Serial Interface, Timer/Counters, Watchdog, and the interrupt system to continue operating (if enabled). This enables the MCU to wake up from external triggered interrupts as well as internal ones like the Timer Overflow and UART Receive Complete interrupts. If wake-up from the Analog Comparator interrupt is not required, the Analog Comparator can be powered down by setting the ACD bit in the Analog Comparator Control and Status Register – ACSR. This will reduce power consumption in Idle Mode. If the ADC is enabled, a conversion starts automatically when this mode is entered.

ADC Noise Reduction Mode

When the SM1/SM0 bits are set to 01, the SLEEP instruction makes the MCU enter ADC Noise Reduction mode, stopping the CPU but allowing the ADC, the external interrupts, the Two-wire Serial Interface address watch, Timer/Counter2 and the Watchdog to continue operating (if enabled). This improves the noise environment for the ADC, enabling higher resolution measurements. If the ADC is enabled, a conversion starts automatically when this mode is entered. Apart from the ADC Conversion Complete interrupt, only an External Reset, a Watchdog Reset (if enabled), a Brown-out Reset, a Two-wire Serial Interface address match interrupt, or an external level interrupt can wake up the MCU from ADC Noise Reduction Mode. A Timer/Counter2 Output Compare or overflow event will wake up the MCU, but will not generate an interrupt unless Timer/Counter2 is clocked asynchronously.

In future devices this is subject to change. It is recommended for future code compatibility to disable Timer/Counter2 interrupts during ADC Noise Reduction mode if the Timer/Counter2 is clocked synchronously.

Timer/Counter1 Control Register A – TCCR1A

Bit	7	6	5	4	3	2	1	0	
$2F ($4F)	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	PWM11	PWM10	TCCR1A
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	

- **Bits 7, 6 – COM1A1, COM1A0: Compare Output Mode1A, Bits 1, and 0**

The COM1A1 and COM1A0 control bits determine any output pin action following a Compare Match in Timer/Counter1. Any output pin actions affect pin OC1A – Output Compare A. This is an alternative function to an I/O Port, and the corresponding direction control bit must be set (one) to control an output pin. The control configuration is shown in Table 10.

- **Bits 5, 4 – COM1B1, COM1B0: Compare Output Mode1B, Bits 1, and 0**

The COM1B1 and COM1B0 control bits determine any output pin action following a Compare Match in Timer/Counter1. Any output pin actions affect pin OC1B – Output Compare B. This is an alternative function to an I/O port, and the corresponding direction control bit must be set (one) to control an output pin. The control configuration is shown in Table 10.

Table 12. Compare 1 Mode Select[1]

COM1X1	COM1X0	Description
0	0	Timer/Counter1 disconnected from output pin OC1X
0	1	Toggle the OC1X output line.
1	0	Clear the OC1X output line (to zero).
1	1	Set the OC1X output line (to one).

Note: 1. X = A or B.

In PWM mode, these bits have a different function. Refer to Table 14 for a detailed description.

- **Bit 3 – FOC1A: Force Output Compare1A**

Writing a logical one to this bit, forces a change in the Compare Match Output pin PD5 according to the values already set in COM1A1 and COM1A0. If the COM1A1 and COM1A0 bits are written in the same cycle as FOC1A, the new settings will not take effect until next Compare Match or Forced Compare Match occurs. The Force Output Compare bit can be used to change the output pin without waiting for a Compare Match in the Timer. The automatic action programmed in COM1A1 and COM1A0 happens as if a Compare Match had occurred, but no interrupt is generated and it will not clear the timer even if CTC1 in TCCR1B is set. The corresponding I/O pin must be set as an output pin for the FOC1A bit to have effect on the pin. The FOC1A bit will always be read as zero. The setting of the FOC1A bit has no effect in PWM mode.

- **Bit 2 – FOC1B: Force Output Compare1B**

Writing a logical one to this bit, forces a change in the Compare Match Output pin PD4 according to the values already set in COM1B1 and COM1B0. If the COM1B1 and COM1B0 bits are written in the same cycle as FOC1B, the new settings will not take effect until next Compare Match or Forced Compare Match occurs. The Force Output Compare bit can be used to change the output pin without waiting for a Compare Match

in the Timer. The automatic action programmed in COM1B1 and COM1B0 happens as if a Compare Match had occurred, but no interrupt is generated. The corresponding I/O pin must be set as an output pin for the FOC1B bit to have effect on the pin. The FOC1B bit will always be read as zero. The setting of the FOC1B bit has no effect in PWM mode.

- **Bits 1..0 – PWM11, PWM10: Pulse Width Modulator Select Bits**

These bits select PWM operation of Timer/Counter1 as specified in Table 11. This mode is described on page 48.

Table 13. PWM Mode Select

PWM11	PWM10	Description
0	0	PWM operation of Timer/Counter1 is disabled
0	1	Timer/Counter1 is an 8-bit PWM
1	0	Timer/Counter1 is a 9-bit PWM
1	1	Timer/Counter1 is a 10-bit PWM

Timer/Counter1 Control Register B – TCCR1B

Bit	7	6	5	4	3	2	1	0	
$2E ($4E)	ICNC1	ICES1	–	–	CTC1	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – ICNC1: Input Capture1 Noise Canceler (4 CKs)**

When the ICNC1 bit is cleared (zero), the Input Capture trigger noise canceler function is disabled. The Input Capture is triggered at the first rising/falling edge sampled on the ICP – Input Capture Pin – as specified. When the ICNC1 bit is set (one), four successive samples are measures on the ICP – Input Capture Pin, and all samples must be high/low according to the Input Capture trigger specification in the ICES1 bit. The actual sampling frequency is XTAL clock frequency.

- **Bit 6 – ICES1: Input Capture1 Edge Select**

While the ICES1 bit is cleared (zero), the Timer/Counter1 contents are transferred to the Input Capture Register – ICR1 – on the falling edge of the Input Capture Pin – ICP. While the ICES1 bit is set (one), the Timer/Counter1 contents are transferred to the Input Capture Register – ICR1 – on the rising edge of the Input Capture Pin – ICP.

- **Bits 5, 4 – Res: Reserved Bits**

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

- **Bit 3 – CTC1: Clear Timer/Counter1 on Compare Match**

When the CTC1 control bit is set (one), the Timer/Counter1 is Reset to $0000 in the clock cycle after a Compare A Match. If the CTC1 control bit is cleared, Timer/Counter1 continues counting and is unaffected by a Compare Match. When a prescaling of 1 is used, and the Compare A Register is set to C, the timer will count as follows if CTC1 is set:

... | C-1 | C | 0 | 1 |...

When the prescaler is set to divide by eight, the Timer will count like this:

... | C-1, C-1, C-1, C-1, C-1, C-1, C-1, C-1 | C, C, C, C, C, C, C, C | 0, 0, 0, 0, 0, 0, 0, 0 |1,1,1,1,1,1,1,1|...

In PWM mode, this bit has a different function. If the CTC1 bit is cleared in PWM mode, the Timer/Counter1 acts as an up/down counter. If the CTC1 bit is set (one), the Timer/Counter wraps when it reaches the TOP value. Refer to page 48 for a detailed description.

- **Bits 2..0 – CS12, CS11, CS10: Clock Select1, Bit 2, 1, and 0**

The Clock Select1 bits 2, 1, and 0 define the prescaling source of Timer/Counter1.

Table 14. Clock 1 Prescale Select

CS12	CS11	CS10	Description
0	0	0	Stop, the Timer/Counter1 is stopped.
0	0	1	CK
0	1	0	CK/8
0	1	1	CK/64
1	0	0	CK/256
1	0	1	CK/1024
1	1	0	External Pin T1, falling edge
1	1	1	External Pin T1, rising edge

The Stop condition provides a Timer Enable/Disable function. The prescaled modes are scaled directly from the CK Oscillator clock. If the external pin modes are used for Timer/Counter1, transitions on PB1/(T1) will clock the counter even if the pin is configured as an output. This feature can give the user SW control of the counting.

Timer/Counter1 – TCNT1H and TCNT1L

Bit	15	14	13	12	11	10	9	8	
$2D ($4D)	**MSB**								**TCNT1H**
$2C ($4C)								**LSB**	**TCNT1L**
	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	

This 16-bit register contains the prescaled value of the 16-bit Timer/Counter1. To ensure that both the high and low bytes are read and written simultaneously when the CPU accesses these registers, the access is performed using an 8-bit temporary register (TEMP). This temporary register is also used when accessing OCR1A, OCR1B, and ICR1. If the main program and also interrupt routines perform access to registers using TEMP, interrupts must be disabled during access from the main program and interrupt routines.

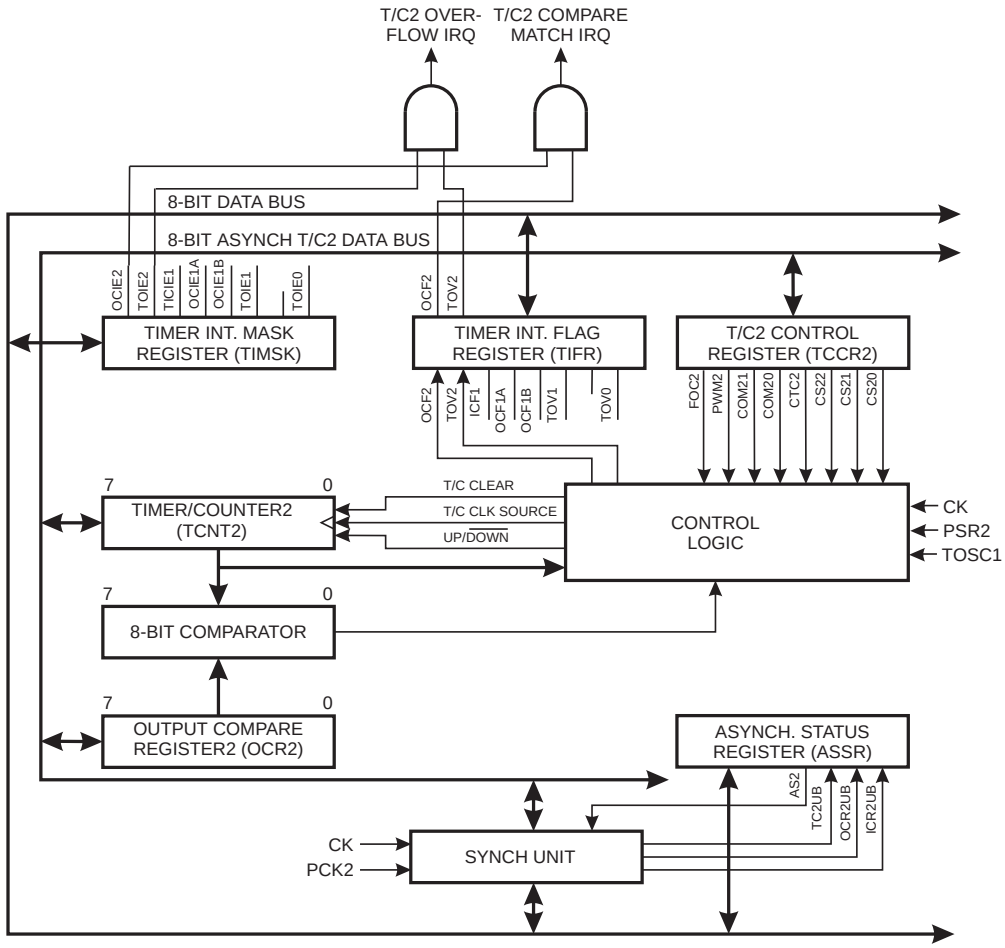
In overflow PWM mode, the table above is only valid for OCR1X = TOP.

In PWM mode, the Timer Overflow Flag1, TOV1, is set when the counter advances from $0000. In overflow PWM mode, the Timer Overflow Flag is set as in Normal Timer/Counter mode. Timer Overflow Interrupt1 operates exactly as in Normal Timer/Counter mode, i.e., it is executed when TOV1 is set provided that Timer Overflow Interrupt1 and global interrupts are enabled. This also applies to the Timer Output Compare1 Flags and interrupts.

8-bit Timer/Counter 2

Figure 37 shows the block diagram for Timer/Counter2.

Figure 37. Timer/Counter2 Block Diagram



The 8-bit Timer/Counter2 can select clock source from CK, prescaled CK, or external crystal input TOSC1. It can also be stopped as described in the section “Timer/Counter2 Control Register – TCCR2” on page 52.

The Status Flags (Overflow and Compare Match) are found in the Timer/Counter Interrupt Flag Register – TIFR. Control signals are found in the Timer/Counter Control Register TCCR2. The interrupt enable/disable settings are found in “The Timer/Counter Interrupt Mask Register – TIMSK” on page 31.

When Timer/Counter2 is externally clocked, the external signal is synchronized with the Oscillator frequency of the CPU. To assure proper sampling of the external clock, the minimum time between two external clock transitions must be at least one internal CPU

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 clock cycles, it executes the interrupt routine, and resumes execution from the instruction following SLEEP.

- During asynchronous operation, the synchronization of the interrupt flags for the asynchronous timer takes three 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.
- After waking up from Power-save mode with the asynchronous timer enabled, there will be a short interval in which TCNT2 will read as the same value as before Power-save mode was entered. After an edge on the asynchronous clock, TCNT2 will read correctly (The compare and overflow functions of the Timer are not affected by this behavior.). Safe procedure to ensure that the correct value is read:
  1. Write any value to either of the registers OCR2 or TCCR2.
  2. Wait for the corresponding Update Busy Flag to be cleared.
  3. Read TCNT2.

  Note that OCR2 and TCCR2 are never modified by hardware, and will always read correctly.

The EEPROM Control Register – EECR

Bit	7	6	5	4	3	2	1	0	
$1C ($3C)	–	–	–	–	EERIE	EEMWE	EEWE	EERE	EECR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..4 – Res: Reserved Bits**

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

- **Bit 3 – EERIE: EEPROM Ready Interrupt Enable**

When the I bit in SREG and EERIE are set (one), the EEPROM Ready Interrupt is enabled. When cleared (zero), the interrupt is disabled. The EEPROM Ready interrupt generates a constant interrupt when EEWE is cleared (zero).

- **Bit 2 – EEMWE: EEPROM Master Write Enable**

The EEMWE bit determines whether setting EEWE to one causes the EEPROM to be written. When EEMWE is set(one) setting EEWE will write data to the EEPROM at the selected address If EEMWE is zero, setting EEWE will have no effect. When EEMWE has been set (one) by software, hardware clears the bit to zero after four clock cycles. See the description of the EEWE bit for an EEPROM write procedure.

- **Bit 1 – EEWE: EEPROM Write Enable**

The EEPROM Write Enable Signal EEWE is the write strobe to the EEPROM. When address and data are correctly set up, the EEWE bit must be set to write the value into the EEPROM. The EEMWE bit must be set when the logical one is written to EEWE, otherwise no EEPROM write takes place. The following procedure should be followed when writing the EEPROM (the order of steps 2 and 3 is not essential):

1. Wait until EEWE becomes zero.
2. Write new EEPROM address to EEAR (optional).
3. Write new EEPROM data to EEDR (optional).
4. Write a logical one to the EEMWE bit in EECR.
5. Within four clock cycles after setting EEMWE, write a logical one to EEWE.

Caution: An interrupt between step 4 and step 5 will make the write cycle fail, since the EEPROM Master Write Enable will time-out. If an interrupt routine accessing the EEPROM is interrupting another EEPROM access, the EEAR or EEDR Register will be modified, causing the interrupted EEPROM access to fail. It is recommended to have the Global Interrupt Flag cleared during the four last steps to avoid these problems.

When the write access time (see Table 24) has elapsed, the EEWE bit is cleared (zero) by hardware. The user software can poll this bit and wait for a zero before writing the next byte. When EEWE has been set, the CPU is halted for four cycles before the next instruction is executed.

- **Bit 0 – EERE: EEPROM Read Enable**

The EEPROM Read Enable Signal EERE is the read strobe to the EEPROM. When the correct address is set up in the EEAR Register, the EERE bit must be set. When the EERE bit is cleared (zero) by hardware, requested data is found in the EEDR Register. The EEPROM read access takes one instruction, and there is no need to poll the EERE bit. When EERE has been set, the CPU is halted for two cycles before the next instruction is executed.

When the Two-wire Serial Interface Interrupt Flag is set, the status code in TWSR is used to determine the appropriate software action. For each status code, the required software action and details of the following serial transfer are given in Table 32 to Table 36.

Master Transmitter Mode

In the Master Transmitter mode, a number of data bytes are transmitted to a Slave Receiver (see Figure 52). Before Master Transmitter mode can be entered, the TWCR must be initialized as follows:

Table 29. TWCR: Master Transmitter Mode Initialization

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

TWEN must be set to enable the Two-wire Serial Interface, TWSTA and TWSTO must be cleared.

The Master Transmitter mode may now be entered by setting the TWSTA bit. The Two-wire Serial Interface logic will then test the Two-wire Serial Bus and generate a START condition as soon as the bus becomes free. When a START condition is transmitted, the Two-wire Serial Interface Interrupt Flag (TWINT) is set by hardware, and the status code in TWSR will be $08. TWDR must then be loaded with the slave address and the data direction bit (SLA+W). Clearing the TWINT bit in software will continue the transfer. The TWINT Flag is cleared by writing a logic one to the flag.

When the slave address and the direction bit have been transmitted and an acknowledgement bit has been received, TWINT is set again and a number of status codes in TWSR are possible. Possible status codes in Master mode are $18, $20, or $38. The appropriate action to be taken for each of these status codes is detailed in Table 32. The data must be loaded when TWINT is high only. If not, the access will be discarded, and the Write Collision bit – TWWC will be set in the TWCR Register. This scheme is repeated until the last byte is sent and the transfer is ended by generating a STOP condition or a repeated START condition. A STOP condition is generated by setting TWSTO, a repeated START condition is generated by setting TWSTA and TWSTO.

After a repeated START condition (state $10) the Two-wire Serial Interface can access the same Slave again, or a new Slave without transmitting a STOP condition. Repeated START enables the Master to switch between Slaves, Master Transmitter mode and Master Receiver mode without loosing control over the bus.

Assembly code illustrating operation of the Master Transmitter mode is given at the end of the TWI section.

Master Receiver Mode

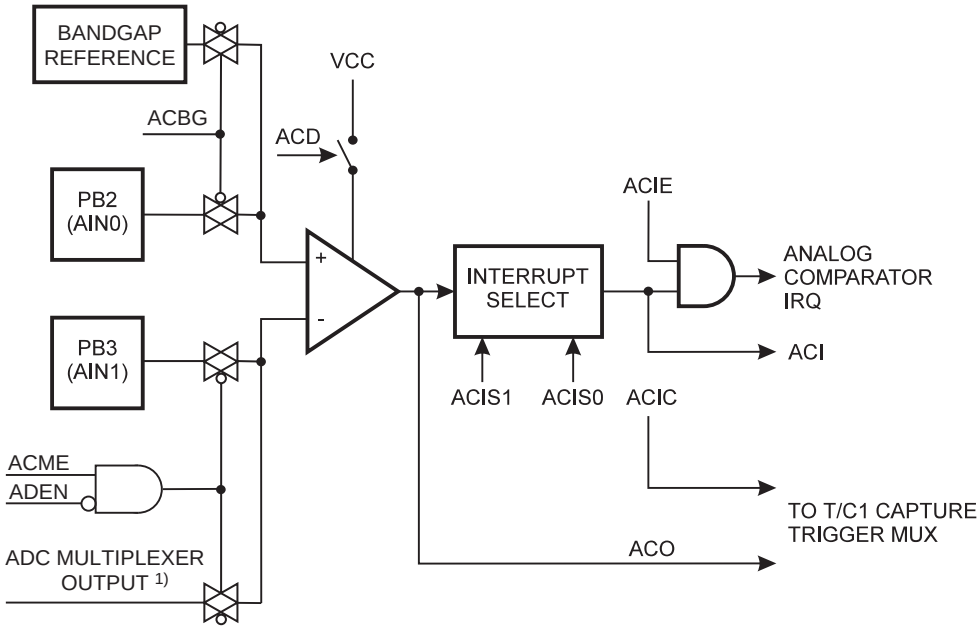
In the Master Receiver mode, a number of data bytes are received from a Slave Transmitter (see Figure 53). The transfer is initialized as in the Master Transmitter mode. When the START condition has been transmitted, the TWINT Flag is set by hardware. The software must then load TWDR with the 7-bit slave address and the Data Direction bit (SLA+R). The transfer will then continue when the TWINT Flag is cleared by software.

When the slave address and the direction bit have been transmitted and an acknowledgement bit has been received, TWINT is set again and a number of status codes in TWSR are possible. Possible status codes in Master mode are $40, $48, or $38. The appropriate action to be taken for each of these status codes is detailed in Table 52. Received data can be read from the TWDR Register when the TWINT Flag is set high by hardware. This scheme is repeated until the last byte has been received and a STOP condition is transmitted by writing a logic one to the TWSTO bit in the TWCR Register.

The Analog Comparator

The Analog Comparator compares the input values on the positive pin PB2 (AIN0) and negative pin PB3 (AIN1). When the voltage on the positive pin PB2 (AIN0) is higher than the voltage on the negative pin PB3 (AIN1), the Analog Comparator Output, ACO, is set (one). The comparator's output can be set to trigger the Timer/Counter1 Input Capture function. In addition, the comparator can trigger a separate interrupt, exclusive to the Analog Comparator. The user can select Interrupt triggering on comparator output rise, fall or toggle. A block diagram of the comparator and its surrounding logic is shown in Figure 56.

Figure 56. Analog Comparator Block Diagram



Note: 1. See Figure 57 on page 106.

The Analog Comparator Control And Status Register – ACSR

Bit	7	6	5	4	3	2	1	0	
$08 ($28)	**ACD**	**ACBG**	**ACO**	**ACI**	**ACIE**	**ACIC**	**ACIS1**	**ACIS0**	**ACSR**
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	N/A	0	0	0	0	0	

- **Bit 7 – ACD: Analog Comparator Disable**

When this bit is set(one), the power to the Analog Comparator is switched off. This bit can be set at any time to turn off the Analog Comparator. This will reduce power consumption in active and Idle mode. When changing the ACD bit, the Analog Comparator Interrupt must be disabled by clearing the ACIE bit in ACSR. Otherwise an interrupt can occur when the bit is changed.

- **Bit 6 – ACBG: Analog Comparator Bandgap Select**

When this bit is set and the BOD is enabled (BODEN Fuse is programmed), a fixed bandgap voltage of nominally 1.22V replaces the positive input to the Analog Comparator. When this bit is cleared, AIN0 is applied to the positive input of the Analog Comparator.

- **Bits 2..0 – ADPS2..0: ADC Prescaler Select Bits**

These bits determine the division factor between the XTAL frequency and the input clock to the ADC.

Table 42. ADC Prescaler Selections

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

The ADC Data Register – ADCL and ADCH

ADLAR = 0

Bit	15	14	13	12	11	10	9	8	
$05 ($25)	SIGN	–	–	–	–	–	ADC9	ADC8	ADCH
$04 ($24)	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

ADLAR = 1

Bit	15	14	13	12	11	10	9	8	
$05 ($25)	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
$04 ($24)	ADC1	ADC0	–	–	–	–	–	–	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

When an ADC conversion is complete, the result is found in these two registers.

When ADCL is read, the ADC Data Register is not updated until ADCH is read. Consequently, if the result is left adjusted and no more than 8-bit precision is required, it is sufficient to read ADCH. Otherwise, ADCL must be read first, then ADCH.

The ADLAR bit in ADMUX affects the way the result is read from the registers. If ADLAR is set, the result is left adjusted. If ADLAR is cleared (default), the result is right adjusted.

- **ADC9..0: ADC Conversion result**

These bits represent the result from the conversion. $000 represents analog ground, and $3FF represents the selected reference voltage minus one LSB.

ADC Characteristics

Table 43. ADC Characteristics

Symbol	Parameter	Condition	Min	Typ	Max	Units
	Resolution	Single-ended Conversion		10		Bits
	Absolute accuracy	V_{REF} = 4V ADC clock = 200 kHz		1	2	LSB
	Absolute accuracy	V_{REF} = 4V ADC clock = 1 MHz		4		LSB
	Absolute accuracy	V_{REF} = 4V ADC clock = 2 MHz		16		LSB
	Integral Non-linearity	V_{REF} > 2V		0.5		LSB
	Differential Non-linearity	V_{REF} > 2V		0.5		LSB
	Zero Error (Offset)	V_{REF} > 2V		1		LSB
	Conversion Time	Free Running Conversion	65		260	µs
	Clock Frequency		50		200	kHz
AV_{CC}	Analog Supply Voltage		V_{CC} - 0.3(1)		V_{CC} + 0.3(2)	V
V_{REF}	Reference Voltage		2 V		AV_{CC}	V
VINT	Internal Voltage Reference		2.35	2.56	2.77	V
V_{BG}	Bandgap Voltage Reference		1.12	1.22	1.32	V
R_{REF}	Reference Input Resistance		6	10	13	kΩ
V_{IN}	Input Voltage		AGND		AREF	V
R_{AIN}	Analog Input Resistance			100		MΩ

Notes: 1. Minimum for AVCC is 2.7V.
2. Maximum for AVCC is 5.5V.

Figure 65. PORTB Schematic Diagram (Pins PB2 and PB3)

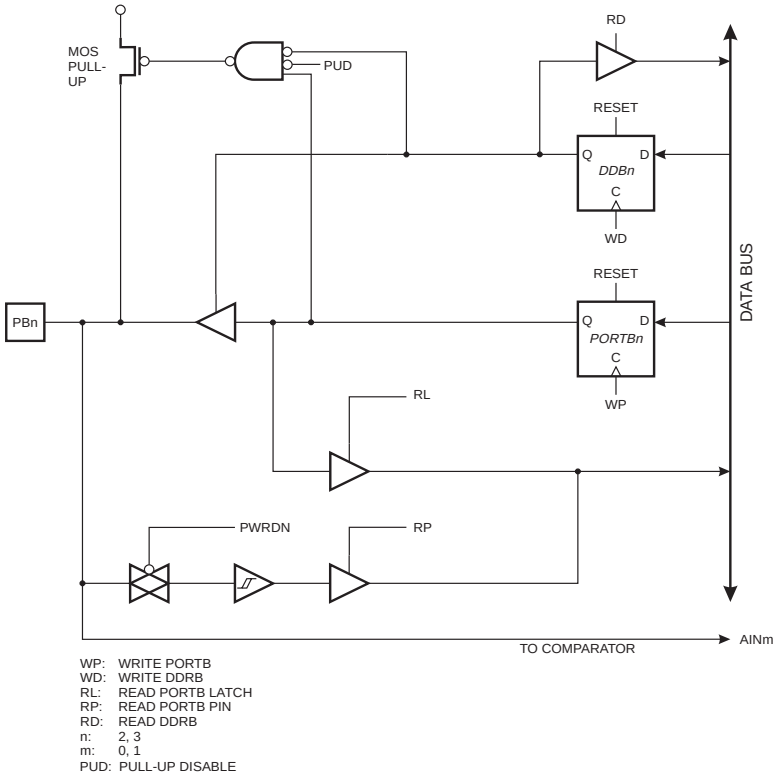
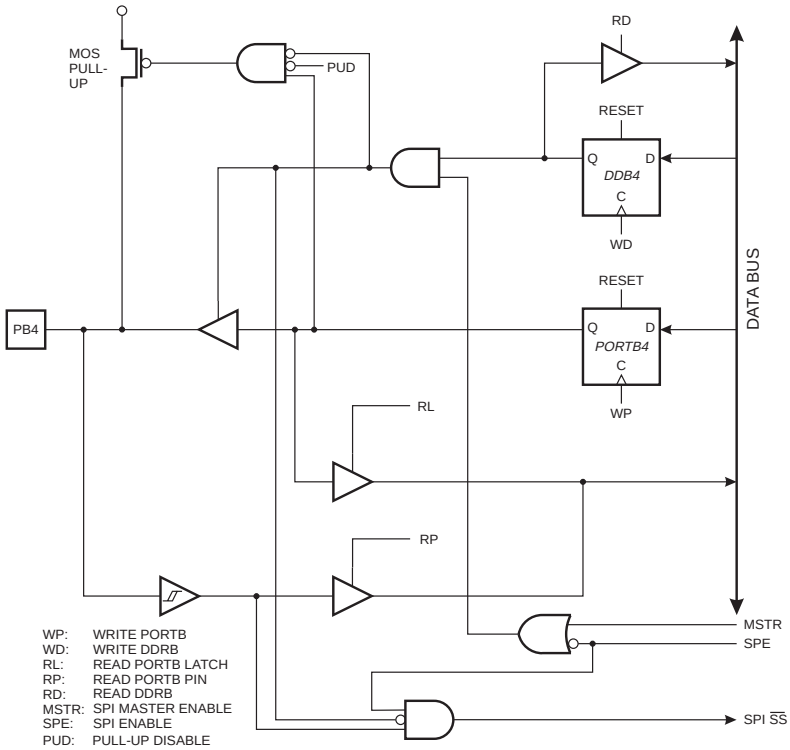


Figure 66. PORTB Schematic Diagram (Pin PB4)



Perform a Page Write

To execute Page Write, set up the address in the Z-pointer, write “00101” 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. During this operation, Z6:Z0 must be zero to ensure that the page is written correctly. It is recommended that the interrupts are disabled during the page write operation.

Consideration while Updating the Boot Loader Section

Special care must be taken if the user allows the Boot Loader section to be updated by leaving Boot Lock bit 11 unprogrammed. An accidental write to the Boot Loader itself can corrupt the entire Boot Loader, and further software updates might be impossible. If it is not necessary to change the Boot Loader software itself, it is recommended to program the Boot Lock Bit 11 to protect the Boot Loader software from any internal software changes.

Wait for SPM Instruction to Complete

Though the CPU is halted during Page Write, Page Erase or Lock bit write, for future compatibility, the user software must poll for SPM complete by reading the SPMCR Register and loop until the SPMEN bit is cleared after a programming operation. See “Assembly code example for a Boot Loader” on page 141 for a code example.

Instruction Word Read after Page Erase, Page Write, and Lock Bit Write

To ensure proper instruction pipelining after programming action (Page Erase, Page Write, or Lock bit write), the SPM instruction must be followed with the sequence (.dw $FFFF - NOP) as shown below:

```
spm
.dw $FFFF
nop
```

If not, the instruction following SPM might fail. It is not necessary to add this sequence when the SPM instruction only loads the temporary buffer.

Avoid Reading the Application Section During Self-Programming

During Self-Programming (either Page Erase or Page Write), the user software should not read the application section. The user software itself must prevent addressing this section during the Self-Programming operations. This implies that interrupts must be disabled. Before addressing the application section after the programming is completed, for future compatibility, the user software must write “10001” to the five LSB in SPMCR and execute SPM within four clock cycles. Then the user software should verify that the ASB bit is cleared. See “Assembly code example for a Boot Loader” on page 141 for an example. Though the ASB and ASRE bits have no special function in this device, it is important for future code compatibility that they are treated as described above.

Boot Loader Lock Bits

ATmega163 has two separate sets of Boot Lock bits which can be set independently. This gives the user a unique flexibility to select different levels of protection.

The user can select:

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

See Table and Table for further details. The Boot Lock bits can be set in software and in Serial or Parallel Programming mode, but they can only be cleared by a chip erase command.

```
  ; re-enable the Application Section
      ldi     spmcrval, (1<<ASRE) + (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 Application Section
      ldi     spmcrval, (1<<ASRE) + (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

  ; return to Application Section
  ; verify that Application Section is safe to read
Return:
      in      temp1, SPMCR
      sbrs    temp1, ASB              ; If ASB is set, the AS is not ready yet
      ret
  ; re-enable the Applicaiton Section
      ldi     spmcrval, (1<<ASRE) + (1<<SPMEN)
      call    Do_spm
      rjmp    Return

Do_spm:
  ; input: spmcrval determines SPM action
  ; check that no EEPROM write access is running
Wait_ee:
      sbic    EECR, EEWE
      rjmp    Wait_ee
  ; SPM timed sequence
      out     SPMCR, spmcrval
      spm
      .dw $FFFF                       ; ensure proper pipelining
      nop                             ; of next instruction
  ; check for SPM complete
Wait_spm:
      in      temp1, SPMCR
```

Table 61. Serial Programming Instruction Set

Instruction	Instruction Format				Operation
	Byte 1	Byte 2	Byte 3	Byte4	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx	Enable Serial Programming after RESET goes low.
Chip Erase	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Chip Erase EEPROM and Flash.
Read Program Memory	0010 **H**000	xxxa **aaaa**	**bbbb bbbb**	**oooo oooo**	Read **H** (high or low) data **o** from Program memory at word address **a**:**b**.
Load Program Memory Page	0100 **H**000	xxxx xxxx	xx**bb bbbb**	**iiii iiii**	Write **H** (high or low) data **i** to Program Memory page at word address **b**.
Write Program Memory Page	0100 1100	xxxa **aaaa**	**bb**xx xxxx	xxxx xxxx	Write Program Memory Page at address **a**:**b**.
Read EEPROM Memory	1010 0000	xxxx xxx**a**	**bbbb bbbb**	**oooo oooo**	Read data **o** from EEPROM memory at address **a**:**b**.
Write EEPROM Memory	1100 0000	xxxx xxx**a**	**bbbb bbbb**	iiii iiii	Write data **i** to EEPROM memory at address **a**:**b**.
Read Lock Bits	0101 1000	0000 0000	xxxx 0xxx	xx***65 4321***	Read Lock bits. “0” = programmed, “1” = unprogrammed.
Write Lock Bits	1010 1100	111x xxxx	xxxx xxxx	11***65 4321***	Write Lock bits. Set bits ***6*** - ***1*** = “0” to program Lock bits.
Read Signature Byte	0011 0000	xxxx xxxx	xxxx xx**bb**	**oooo oooo**	Read Signature Byte **o** at address **b**.
Write Fuse Bits	1010 1100	1010 0000	xxxx xxxx	**CB**11 **A987**	Set bits **C** - **A**, **9** - **7** = “0” to program, “1” to unprogram
Write Fuse High Bits	1010 1100	1010 1000	xxxx xxxx	1111 1**FED**	Set bits **F** - **D** = “0” to program, “1” to unprogram
Read Fuse Bits	0101 0000	0000 0000	xxxx xxxx	**CB**xx **A987**	Read Fuse bits. “0” = programmed, “1” = unprogrammed
Read Fuse High Bits	0101 1000	0000 1000	xxxx xxxx	xxxx 1**FED**	Read Fuse high bits. “0” = programmed, “1” = unprogrammed
Read Calibration Byte	0011 1000	xxxx xxxx	0000 0000	**oooo oooo**	Read Signature Byte **o** at address **b**.

Note: **a** = address high bits, **b** = address low bits, **H** = 0 - Low byte, 1 - High Byte, **o** = data out, **i** = data in, x = don’t care
1 = lock bit 1, ***2*** = lock bit 2, ***3*** = Boot Lock Bit01, ***4*** = Boot Lock Bit02, ***5*** = Boot Lock Bit11, ***6*** = Boot Lock Bit12, **7** = CKSEL0 Fuse, **8** = CKSEL1 Fuse, **9** = CKSEL2 Fuse, **A** = CKSEL3 Fuse, **B**= BODEN Fuse, **C**= BODLEVEL Fuse, **D**= BOOTRST Fuse, **E**= BOOTSZ0 Fuse, **F**= BOOTSZ1 Fuse

Electrical Characteristics

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground	-1.0V to V_{CC}+0.5V
Voltage on $\overline{\text{RESET}}$ with respect to Ground	-1.0V to +13.0V
Maximum Operating Voltage	6.6V
DC Current per I/O Pin	40.0 mA
DC Current V_{CC} and GND Pins	200.0 mA

*NOTICE: Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

T_A = -40°C to 85°C, V_{CC} = 2.7V to 5.5V (unless otherwise noted)

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{IL}	Input Low-voltage	(Except XTAL1)	-0.5		0.3 V_{CC}[1]	V
V_{IL1}	Input Low-voltage	(XTAL1), CKSEL3 fuse programmed	-0.5		0.3 V_{CC}[1]	V
		(XTAL1), CKSEL3 fuse unprogrammed	-0.5		0.2 V_{CC}[1]	V
V_{IH}	Input High-voltage	(Except XTAL1, $\overline{\text{RESET}}$)	0.6 V_{CC}[2]		V_{CC} + 0.5	V
V_{IH1}	Input High-voltage	(XTAL1), CKSEL3 fuse programmed	0.6 V_{CC}[2]		V_{CC} + 0.5	V
		(XTAL1), CKSEL3 fuse unprogrammed	0.8 V_{CC}[2]		V_{CC} + 0.5	V
V_{IH2}	Input High-voltage	($\overline{\text{RESET}}$)	0.9 V_{CC}[2]		V_{CC} + 0.5	V
V_{OL}	Output Low-voltage[3] (Ports A,B,C,D)	I_{OL} = 20 mA, V_{CC} = 5V I_{OL} = 10 mA, V_{CC} = 3V			0.6 0.5	V V
V_{OH}	Output High-voltage[4] (Ports A,B,C,D)	I_{OH} = -3 mA, V_{CC} = 5V I_{OH} = -1.5 mA, V_{CC} = 3V	4.2 2.3			V V
I_{IL}	Input Leakage Current I/O pin	Vcc = 5.5V, pin low (absolute value)			8.0	µA
I_{IH}	Input Leakage Current I/O pin	Vcc = 5.5V, pin high (absolute value)			980	nA
RRST	Reset Pull-up Resistor		100		500	kΩ
$R_{I/O}$	I/O Pin Pull-up Resistor		35		120	kΩ

Two-wire Serial Interface Characteristics

Table 65 describes the requirements for devices connected to the Two-wire Serial Bus. The ATmega163 Two-wire Serial Interface meets or exceeds these requirements under the noted conditions.

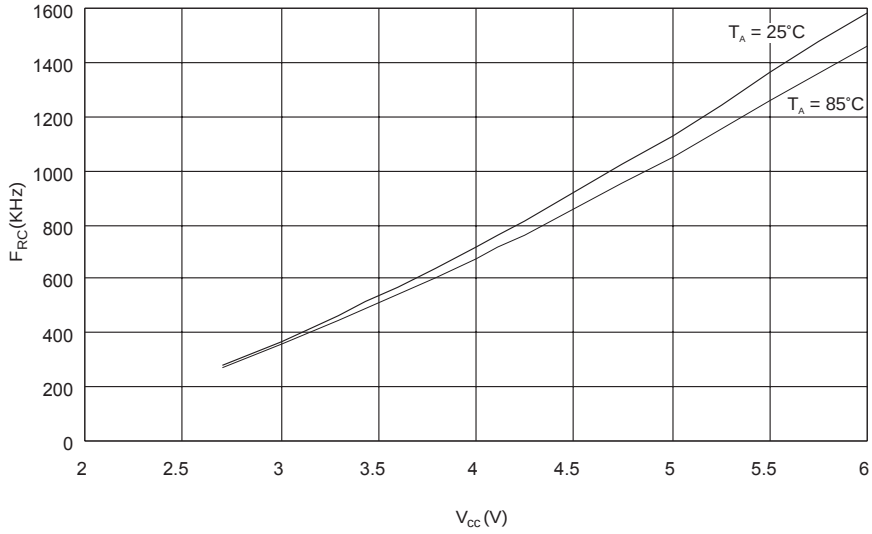
Timing symbols refer to Figure 90.

Table 65. Two-wire Serial Bus Requirements

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low-voltage		-0.5	0.3 V_{CC}	V
V_{IH}	Input High-voltage		0.7 V_{CC}	V_{CC} + 0.5	V
V_{hys}(1)	Hysteresis of Schmitt Trigger Inputs		0.05 V_{CC}(2)	–	V
V_{OL}(1)	Output Low-voltage	3 mA sink current	0	0.4	V
t_{of}(1)	Output Fall Time from V_{IHmin} to V_{ILmax}	10 pF < C_b < 400 pF(3)	20 + 0.1C_b(3)(2)	250	ns
t_{SP}(1)	Spikes Suppressed by Input Filter		0	50(2)	ns
I_i	Input Current each I/O Pin	0.1V_{CC} < V_i < 0.9V_{CC}	-10	10	µA
C_i(1)	Capacitance for each I/O Pin		–	10	pF
f_{SCL}	SCL Clock Frequency	f_{CK}(4) > max(16f_{SCL}, 250kHz)(5)	0	217	kHz
$t_{HD;STA}$	Hold Time (repeated) START Condition	f_{SCL} ≤ 100 kHz	4.0	–	µs
		f_{SCL} > 100 kHz	0.6	–	µs
t_{LOW}	Low Period of the SCL Clock	f_{SCL} ≤ 100 kHz(6)	4.7	–	µs
		f_{SCL} > 100 kHz	1.3	–	µs
t_{HIGH}	High period of the SCL clock	f_{SCL} ≤ 100 kHz	4.0	–	µs
		f_{SCL} > 100 kHz	0.6	–	µs
$t_{SU;STA}$	Set–up time for a repeated START condition	f_{SCL} ≤ 100 kHz	4.7	–	µs
		f_{SCL} > 100 kHz	0.6	–	µs
$t_{HD;DAT}$	Data hold time	f_{SCL} ≤ 100 kHz	0	3.45	µs
		f_{SCL} > 100 kHz	0	0.9	µs
$t_{SU;DAT}$	Data setup time	f_{SCL} ≤ 100 kHz	250	–	ns
		f_{SCL} > 100 kHz	100	–	ns
$t_{SU;STO}$	Setup time for STOP condition	f_{SCL} ≤ 100 kHz	4.0	–	µs
		f_{SCL} > 100 kHz	0.6	–	µs
t_{BUF}	Bus free time between a STOP and START condition	f_{SCL} ≤ 100 kHz	4.7	–	µs
		f_{SCL} > 100 kHz	1.3	–	µs

Notes:
1. In ATmegan163, this parameter is characterized and not 100% tested.
2. Required only for f_{SCL} > 100 kHz.
3. C_b = capacitance of one bus line in pF.
4. f_{CK} = CPU clock frequency
5. This requirement applies to all ATmega163 Two-wire Serial Interface operation. Other devices connected to the Two-wire Serial Bus need only obey the general f_{SCL} requirement.
6. The actual low period generated by the ATmega163 Two-wire Serial Interface is (1/f_{SCL} - 2/f_{CK}), thus f_{CK} must be greater than 6 MHz for the low time requirement to be strictly met at f_{SCL} = 100 kHz.

Figure 94. Watchdog Oscillator Frequency vs. V_{CC}



Sink and source capabilities of I/O ports are measured on one pin at a time.

Figure 95. Pull-up Resistor Current vs. Input Voltage (V_{CC} = 5V)