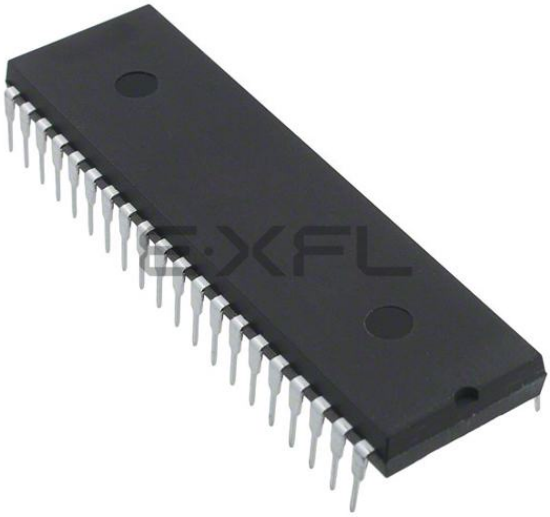


Welcome to E-XFL.COM

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"

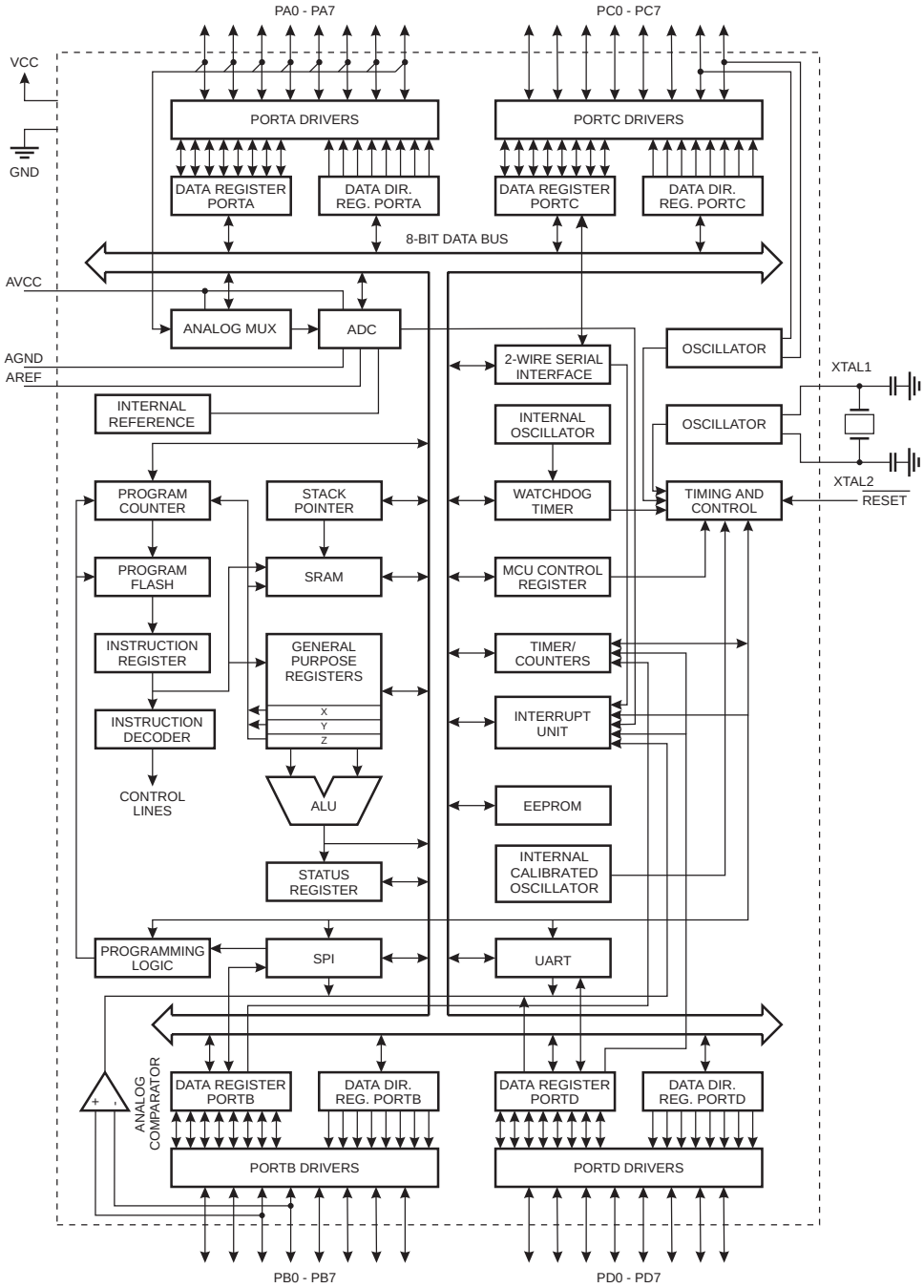
Details	
Product Status	Obsolete
Core Processor	AVR
Core Size	8-Bit
Speed	8MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	32
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C
Mounting Type	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/atmega163-8pc

Description

The ATmega163 is a low-power CMOS 8-bit microcontroller based on the AVR architecture. By executing powerful instructions in a single clock cycle, the ATmega163 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 1. Block Diagram



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

Table 6. Number of Watchdog Oscillator Cycles[1]

BODLEVEL	V_{CC} Condition	Time-out	Number of Cycles
Unprogrammed	2.7V	30 µs	8
Unprogrammed	2.7V	130 µs	32
Unprogrammed	2.7V	4.2 ms	1K
Unprogrammed	2.7V	67 ms	16K
Programmed	4.0V	10 µs	8
Programmed	4.0V	35 µs	32
Programmed	4.0V	5.8 ms	4K
Programmed	4.0V	92 ms	64K

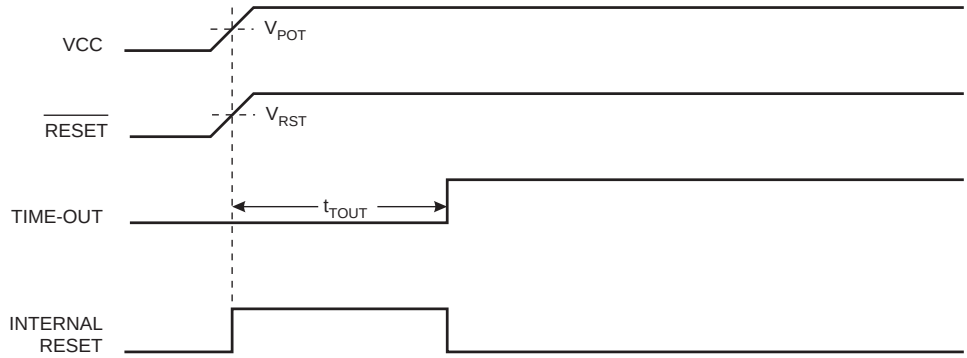
Note: 1. The Bodlevel Fuse can be used to select start-up times even if the Brown-out Detection is disabled (BODEN Fuse unprogrammed).

Power-on Reset

A Power-on Reset (POR) pulse is generated by an On-chip detection circuit. The detection level is defined in Table 4. The POR is activated whenever V_{CC} is below the detection level. The POR circuit can be used to trigger the Start-up Reset, as well as to detect a failure in supply voltage.

A Power-on Reset (POR) circuit ensures that the device is reset from Power-on. Reaching the Power-on Reset threshold voltage invokes a delay counter, which determines the delay, for which the device is kept in RESET after V_{CC} rise. The Time-out Period of the delay counter can be defined by the user through the CKSEL Fuses. The different selections for the delay period are presented in Table 5. The RESET signal is activated again, without any delay, when the V_{CC} decreases below detection level.

Figure 25. MCU Start-up, $\overline{\text{RESET}}$ Tied to V_{CC}.



- **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

(at vector $00A) is executed if a capture triggering event occurs on PD6 (ICP), i.e., when the ICF1 bit is set in the Timer/Counter Interrupt Flag Register – TIFR.

- **Bit 4 – OCIE1A: Timer/Counter1 Output CompareA Match Interrupt Enable**

When the OCIE1A bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Compare A Match Interrupt is enabled. The corresponding interrupt (at vector $00C) is executed if a Compare A Match in Timer/Counter1 occurs, i.e., when the OCF1A bit is set in the Timer/Counter Interrupt Flag Register – TIFR.

- **Bit 3 – OCIE1B: Timer/Counter1 Output CompareB Match Interrupt Enable**

When the OCIE1B bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Compare B Match Interrupt is enabled. The corresponding interrupt (at vector $00E) is executed if a Compare B Match in Timer/Counter1 occurs, i.e., when the OCF1B bit is set in the Timer/Counter Interrupt Flag Register – TIFR.

- **Bit 2 – TOIE1: Timer/Counter1 Overflow Interrupt Enable**

When the TOIE1 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Overflow Interrupt is enabled. The corresponding interrupt (at vector $010) is executed if an overflow in Timer/Counter1 occurs, i.e., when the TOV1 bit is set in the Timer/Counter Interrupt Flag Register – TIFR.

- **Bit 1 – Res: Reserved Bit**

This bit is a reserved bit in the ATmega163 and always reads as zero.

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

When the TOIE0 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter0 Overflow Interrupt is enabled. The corresponding interrupt (at vector $012) is executed if an overflow in Timer/Counter0 occurs, i.e., when the TOV0 bit is set in the Timer/Counter Interrupt Flag Register – TIFR.

The Timer/Counter Interrupt Flag Register – TIFR

Bit	7	6	5	4	3	2	1	0	
$38 ($58)	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	–	TOV0	TIFR
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	x	0	

- **Bit 7 – OCF2: Output Compare Flag2**

The OCF2 bit is set (one) when a Compare Match occurs between the Timer/Counter2 and the data in OCR2 – Output Compare Register2. OCF2 is cleared by hardware when executing the corresponding Interrupt Handling Vector. Alternatively, OCF2 is cleared by writing a logic one to the flag. When the I-bit in SREG, and OCIE2 (Timer/Counter2 Compare Match Interrupt Enable), and the OCF2 are set (one), the Timer/Counter2 Compare Match Interrupt is executed.

- **Bit 6 – TOV2: Timer/Counter2 Overflow Flag**

The TOV2 bit is set (one) when an overflow occurs in Timer/Counter2. TOV2 is cleared by hardware when executing the corresponding inteRrupt Handling Vector. Alternatively, TOV2 is cleared by writing a logic one to the flag. When the SREG I-bit, and TOIE2 (Timer/Counter2 Overflow Interrupt Enable), and TOV2 are set (one), the

MCU Control Register – MCUCR

The MCU Control Register contains control bits for general MCU functions.

Bit	7	6	5	4	3	2	1	0	
$35 ($55)	–	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00	MCUCR
Read/Write	R	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 – Res: Reserved Bit**

This bit is a reserved bit in the ATmega163 and always reads as zero.

- **Bit 6 – SE: Sleep Enable**

The SE bit must be set (one) to make the MCU enter the sleep mode when the SLEEP instruction is executed. To avoid the MCU entering the sleep mode unless it is the programmers purpose, it is recommended to set the Sleep Enable SE bit just before the execution of the SLEEP instruction.

- **Bits 5, 4 – SM1/SM0: Sleep Mode Select Bits 1 and 0**

These bits select between the three available sleep modes as shown in Table 7.

Table 7. Sleep Mode Select

SM1	SM0	Sleep Mode
0	0	Idle
0	1	ADC Noise Reduction
1	0	Power-down
1	1	Power-save

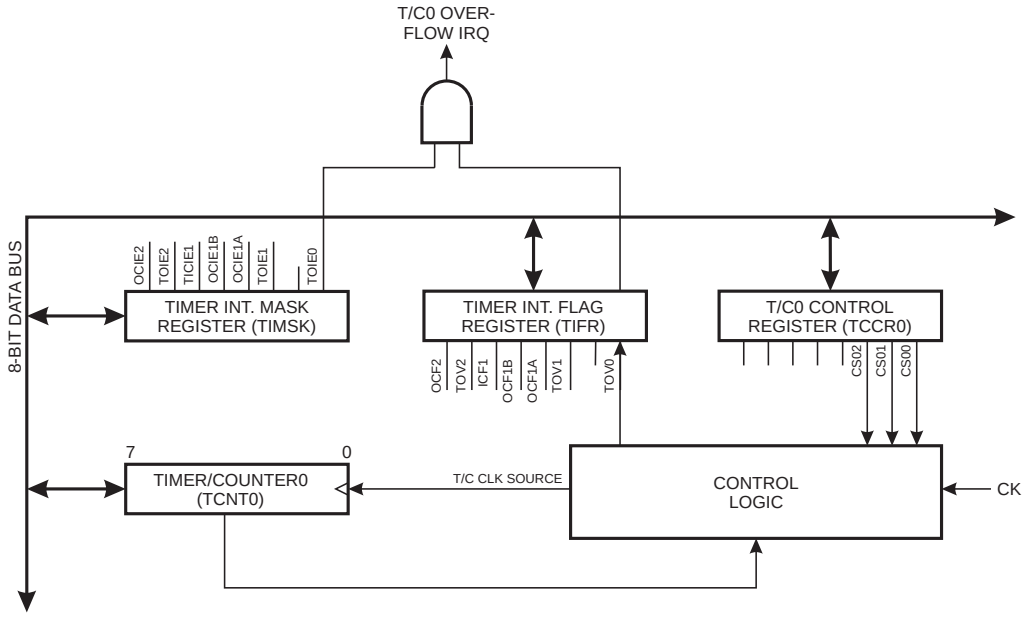
- **Bits 3, 2 – ISC11, ISC10: Interrupt Sense Control 1 Bit 1 and Bit 0**

The External Interrupt 1 is activated by the external pin INT1 if the SREG I-Flag and the corresponding interrupt mask in the GIMSK are set. The level and edges on the external INT1 pin that activate the interrupt are defined in Table 8. The value on the INT1 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 8. Interrupt 1 Sense Control

ISC11	ISC10	Description
0	0	The low level of INT1 generates an interrupt request.
0	1	Any logical change on INT1 generates an interrupt request.
1	0	The falling edge of INT1 generates an interrupt request.
1	1	The rising edge of INT1 generates an interrupt request.

Figure 32. Timer/Counter0 Block Diagram



Timer/Counter0 Control Register – TCCR0

Bit	7	6	5	4	3	2	1	0	
$33 ($53)	–	–	–	–	–	CS02	CS01	CS00	TCCR0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..3 – Res: Reserved Bits**

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

- **Bits 2..0 – CS02, CS01, CS00: Clock Select0, Bit 2, 1, and 0**

The Clock Select0 bits 2,1, and 0 define the prescaling source of Timer0.

Table 11. Clock0 Prescale Select

CS02	CS01	CS00	Description
0	0	0	Stop, Timer/Counter0 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 T0, falling edge
1	1	1	External Pin T0, rising edge

The Stop condition provides a Timer Enable/Disable function. The prescaled CK modes are scaled directly from the CK Oscillator clock. If the external pin modes are used for Timer/Counter0, transitions on PB0/(T0) will clock the counter even if the pin is configured as an output. This feature can give the user SW control of the counting.

Table 15. Timer TOP Values and PWM Frequency

CTC1	PWM11	PWM10	PWM Resolution	Timer TOP Value	Frequency
0	0	1	8-bit	$00FF (255)	$f_{TCK1}/510$
0	1	0	9-bit	$01FF (511)	$f_{TCK1}/1022$
0	1	1	10-bit	$03FF(1023)	$f_{TCK1}/2046$
1	0	1	8-bit	$00FF (255)	$f_{TCK1}/256$
1	1	0	9-bit	$01FF (511)	$f_{TCK1}/512$
1	1	1	10-bit	$03FF(1023)	$f_{TCK1}/1024$

As shown in Table 15, the PWM operates at either 8, 9, or 10 bits resolution. Note the unused bits in OCR1A, OCR1B, and TCNT1 will automatically be written to zero by hardware. For example, bit 9 to 15 will be set to zero in OCR1A, OCR1B, and TCNT1 if the 9-bit PWM resolution is selected. This makes it possible for the user to perform read-modify-write operations in any of the three resolution modes and the unused bits will be treated as don't care.

Table 16. Timer TOP Values and PWM Frequency

PWM Resolution	Timer TOP Value	Frequency
8-bit	$00FF (255)	$f_{TC1}/510$
9-bit	$01FF (511)	$f_{TC1}/1022$
10-bit	$03FF(1023)	$f_{TC1}/2046$

Table 17. Compare1 Mode Select in PWM Mode(1)

CTC1	COM1X 1	COM1X 0	Effect on OCX1
0	0	0	Not connected
0	0	1	Not connected
0	1	0	Cleared on Compare Match, up-counting. Set on Compare Match, down-counting (non-inverted PWM).
0	1	1	Cleared on Compare Match, down-counting. Set on Compare Match, up-counting (inverted PWM).
1	0	0	Not connected
1	0	1	Not connected
1	1	0	Cleared on Compare Match, set on overflow.
1	1	1	Set on Compare Match, cleared on overflow.

Note: 1. X = A or B

Note that in the PWM mode, the 8, 9, or 10 least significant OCR1A/OCR1B bits (depending on resolution), when written, are transferred to a temporary location. They are latched when Timer/Counter1 reaches the value TOP. This prevents the occurrence of odd-length PWM pulses (glitches) in the event of an unsynchronized OCR1A/OCR1B write. See Figure 35 and Figure 36 for an example in each mode.

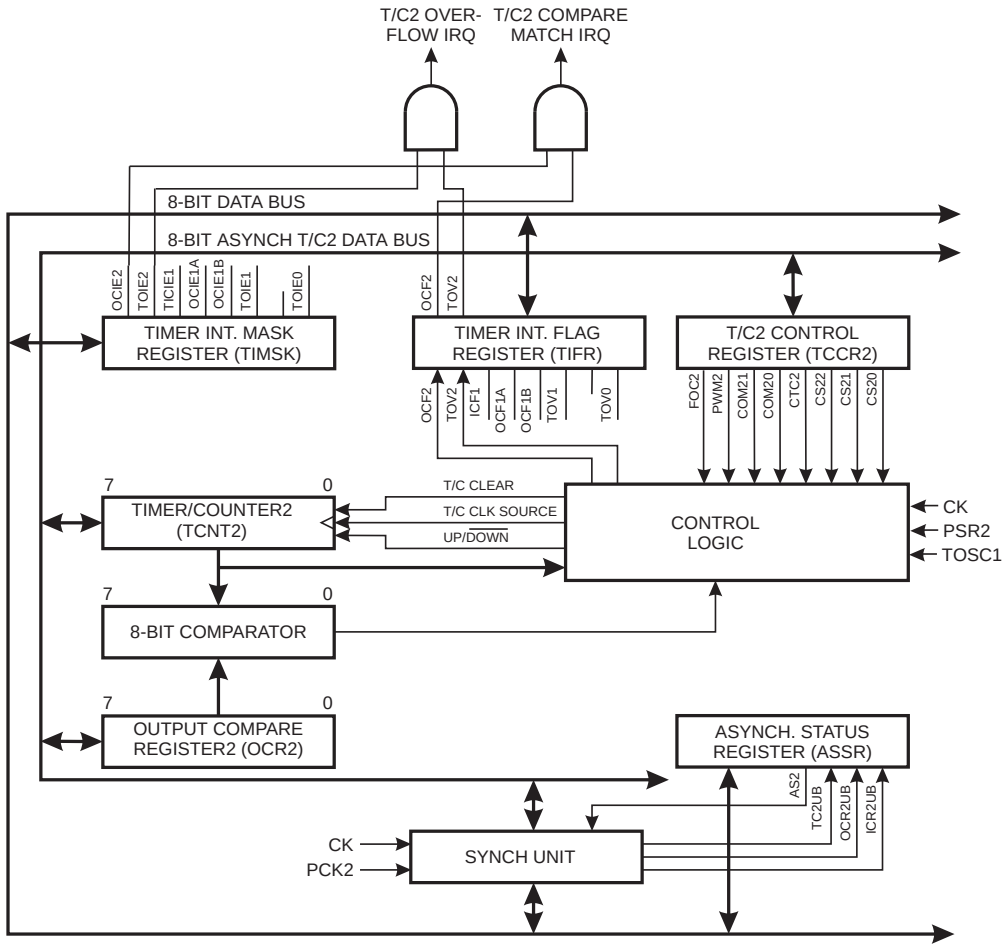
In overflow PWM mode, the table above is only valid for OCR1X = TOP.

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

The Two-wire Serial Interface Bit Rate Register – TWBR

Bit	7	6	5	4	3	2	1	0	
$00 ($20)	TWBR7	TWBR6	TWBR5	TWBR4	TWBR3	TWBR2	TWBR1	TWBR0	TWBR
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..0 – Two-wire Serial Interface Bit Rate Register**

TWBR selects the division factor for the bit rate generator. The bit rate generator is a frequency divider which generates the SCL clock frequency in the Master modes according to the following equation:

$$\text{Bit Rate} = \frac{f_{CK}}{16 + 2(\text{TWBR}) + t_A f_{CK}}$$

- Bit Rate = SCL frequency
- f_{CK} = CPU Clock frequency
- TWBR = Contents of the Two-wire Serial Interface Bit Rate Register
- t_A = Bus alignment adjustion

Note: Both the Receiver and the Transmitter can stretch the low period of the SCL line when waiting for user response, thereby reducing the average bit rate.

TWBR should be set to a value higher than seven to ensure correct Two-wire Serial Bus functionality. The bus alignment adjustion is automatically inserted by the Two-wire Serial Interface, and ensures the validity of setup and hold times on the bus for any TWBR value higher than seven. This adjustment may vary from 200 ns to 600 ns depending on bus loads and drive capabilities of the devices connected to the bus.

The Two-wire Serial Interface Control Register – TWCR

Bit	7	6	5	4	3	2	1	0	
$36 ($56)	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	TWCR
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

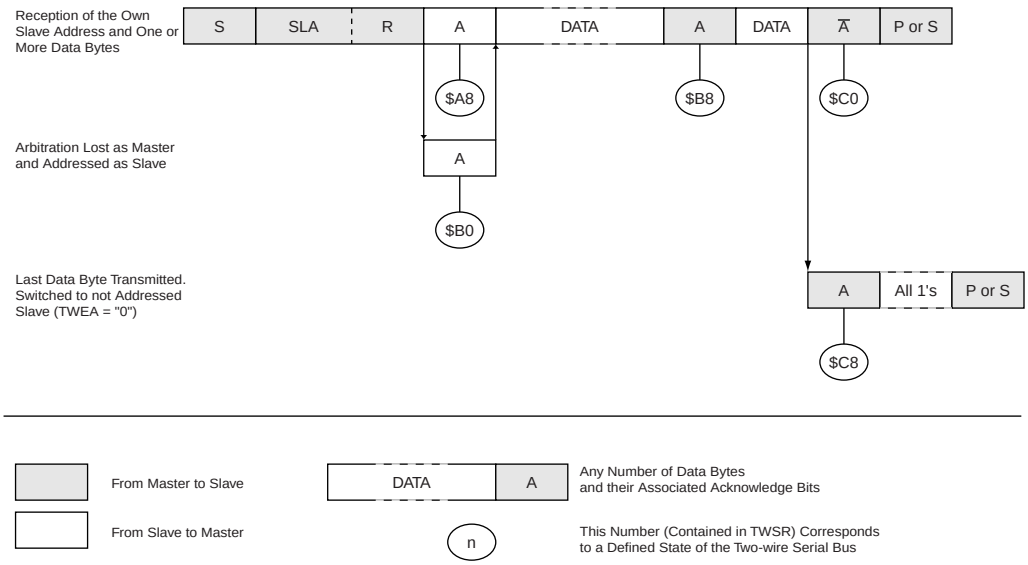
- **Bit 7 – TWINT: Two-wire Serial Interface Interrupt Flag**

This bit is set by hardware when the Two-wire Serial Interface has finished its current job and expects application software response. If the I-bit in the SREG and TWIE in the TWCR Register are set (one), the MCU will jump to the Interrupt Vector at address $22. While the TWINT Flag is set, the bus SCL clock line low period is stretched. The TWINT Flag must be cleared by software by writing a logic one to it. Note that this flag is not automaticaly cleared by hardware when executing the interrupt routine. Also note that clearing this flag starts the operation of the Two-wire Serial Interface, so all accesses to the Two-wire Serial Interface Address Register – TWAR, Two-wire Serial Interface Status Register – TWSR, and Two-wire Serial Interface Data Register – TWDR must be complete before clearing this flag.

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



```
.equ    ST_DATA_NACK    =$C0    ;Data byte has been tramsmitted and NACK
                                ;received
.equ    ST_LAST_DATA    =$C8    ;Last byte in I2DR has been transmitted (TWEA =
                                ;'0'), ACK has been received

;***** Slave Receiver staus codes *****
.equ    SR_SLA_ACK      =$60    ;SLA+R has been received and ACK returned
.equ    SR_ARB_LOST_SLA_ACK=$68;Arbitration lost in SLA+R/W as Master. Own
                                ;SLA+R has been received and ACK returned
.equ    SR_GCALL_ACK    =$70    ;Generall call has been received and ACK
                                ;returned
.equ    SR_ARB_LOST_GCALL_ACK=$78;Arbitration lost in SLA+R/W as Master.
                                ;General Call has been received and ACK
                                ;returned
.equ    SR_DATA_ACK     =$80    ;Previously addressed with own SLA+W. Data byte
                                ;has been received and ACK returned
.equ    SR_DATA_NACK    =$88    ;Previously addressed with own SLA+W. Data byte
                                ;has been received and NACK returned
.equ    SR_GCALL_DATA_ACK=$90;Previously addressed with General Call.Data
                                ;byte has been received and ACK returned
.equ    SR_GCALL_DATA_NACK=$98;Previously addressed with General Call. Data
                                ;byte has been received and NACK returned
.equ    SR_STOP         =$A0    ;A STOP condition or repeated START condition
                                ;has been received while still addressed as a
                                ;slave
;***** Miscellanous States *****
.equ    NO_INFO         =$F8    ;No relevant state information; TWINT = '0'
.equ    BUS_ERROR       =$00    ;Bus error due to illegal START or STOP
                                ;condition
```

- **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.

Scanning Multiple Channels

Since change of analog channel always is delayed until a conversion is finished, the Free Running mode can be used to scan multiple channels without interrupting the converter. Typically, the ADC Conversion Complete interrupt will be used to perform the channel shift. However, the user should take the following fact into consideration:

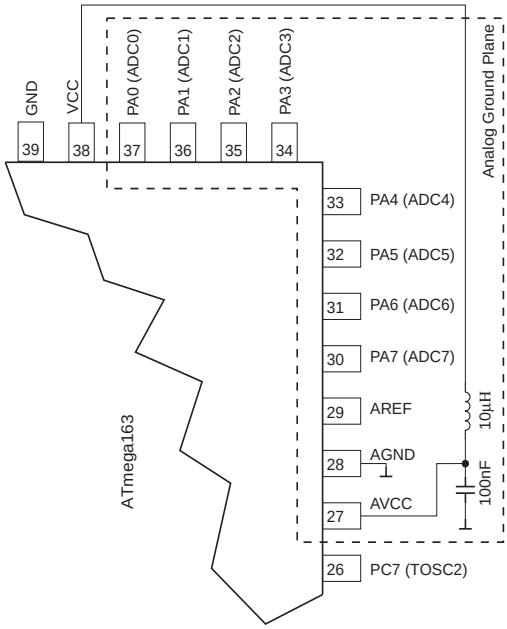
The interrupt triggers once the result is ready to be read. In Free Running mode, the next conversion will start immediately when the interrupt triggers. If ADMUX is changed after the interrupt triggers, the next conversion has already started, and the old setting is used.

ADC Noise Canceling Techniques

Digital circuitry inside and outside the ATmega163 generates EMI which might affect the accuracy of analog measurements. If conversion accuracy is critical, the noise level can be reduced by applying the following techniques:

1. The analog part of the ATmega163 and all analog components in the application should have a separate analog ground plane on the PCB. This ground plane is connected to the digital ground plane via a single point on the PCB.
2. Keep analog signal paths as short as possible. Make sure analog tracks run over the analog ground plane, and keep them well away from high-speed switching digital tracks.
3. The AVCC pin on the ATmega163 should be connected to the digital V_{CC} supply voltage via an LC network as shown in Figure 62.
4. Use the ADC noise canceler function to reduce induced noise from the CPU.
5. If some Port A pins are used as digital outputs, it is essential that these do not switch while a conversion is in progress.

Figure 62. ADC Power Connections



Port D as General Digital I/O

PDn, General I/O pin: The DDDn bit in the DDRD Register selects the direction of this pin. If DDDn is set (one), PDn is configured as an output pin. If DDDn is cleared (zero), PDn is configured as an input pin. If PDn is set (one) when configured as an input pin the MOS pull up resistor is activated. To switch the pull up resistor off the PDn has to be cleared (zero), the pin has to be configured as an output pin, or the PUD bit has to be set. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Table 50. DDDn Bits on Port D Pins[1]

DDDn	PORTDn	PUD	I/O	Pull Up	Comment
0	0	x	Input	No	Tri-state (Hi-Z)
0	1	1	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	PDn will source current if ext. pulled low.
1	0	x	Output	No	Push-pull Zero Output
1	1	x	Output	No	Push-pull One Output

Note: 1. n: 7,6…0, pin number.

Alternate Functions of PORTD

- **OC2 – PORTD, Bit 7**

OC2, Timer/Counter2 Output Compare Match output: The PD7 pin can serve as an external output for the Timer/Counter2 Output Compare. The pin has to be configured as an output (DDD7 set (one)) to serve this function. See the Timer description on how to enable this function. The OC2 pin is also the output pin for the PWM mode timer function.

- **ICP – PORTD, Bit 6**

ICP – Input Capture Pin: The PD6 pin can act as an Input Capture pin for Timer/Counter1. The pin has to be configured as an input (DDD6 cleared(zero)) to serve this function. See the timer description on how to enable this function.

- **OC1A – PORTD, Bit 5**

OC1A, Output Compare Match A output: The PD5 pin can serve as an external output for the Timer/Counter1 Output Compare A. The pin has to be configured as an output (DDD5 set (one)) to serve this function. See the timer description on how to enable this function. The OC1A pin is also the output pin for the PWM mode timer function.

- **OC1B – PORTD, Bit 4**

OC1B, Output Compare Match B output: The PD4 pin can serve as an external output for the Timer/Counter1 Output Compare B. The pin has to be configured as an output (DDD4 set (one)) to serve this function. See the timer description on how to enable this function. The OC1B pin is also the output pin for the PWM mode timer function.

- **INT1 – PORTD, Bit 3**

INT1, External Interrupt Source 1: The PD3 pin can serve as an External Interrupt Source to the MCU. See the interrupt description for further details, and how to enable the source.

Figure 75. PORTD Schematic Diagram (Pin PD1)

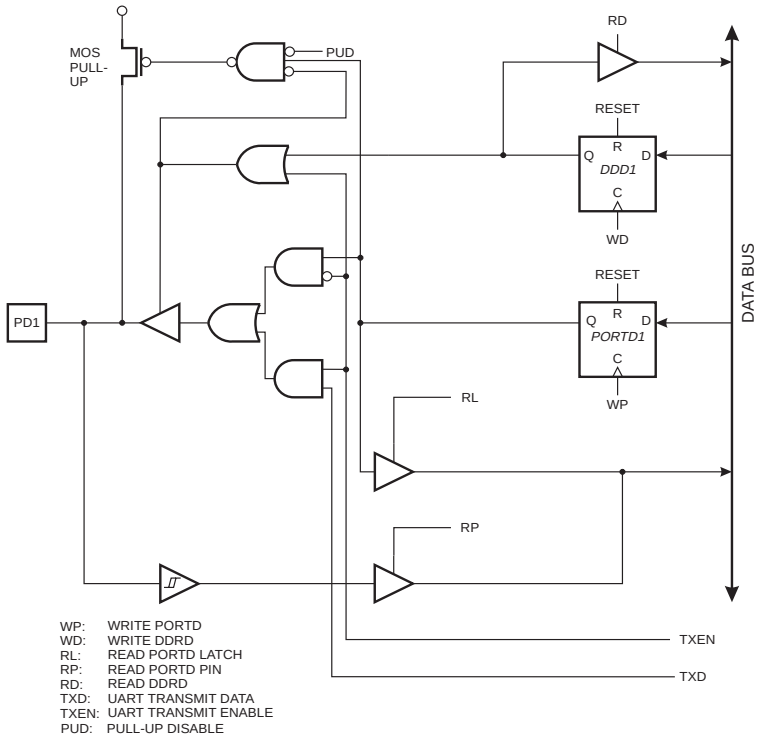
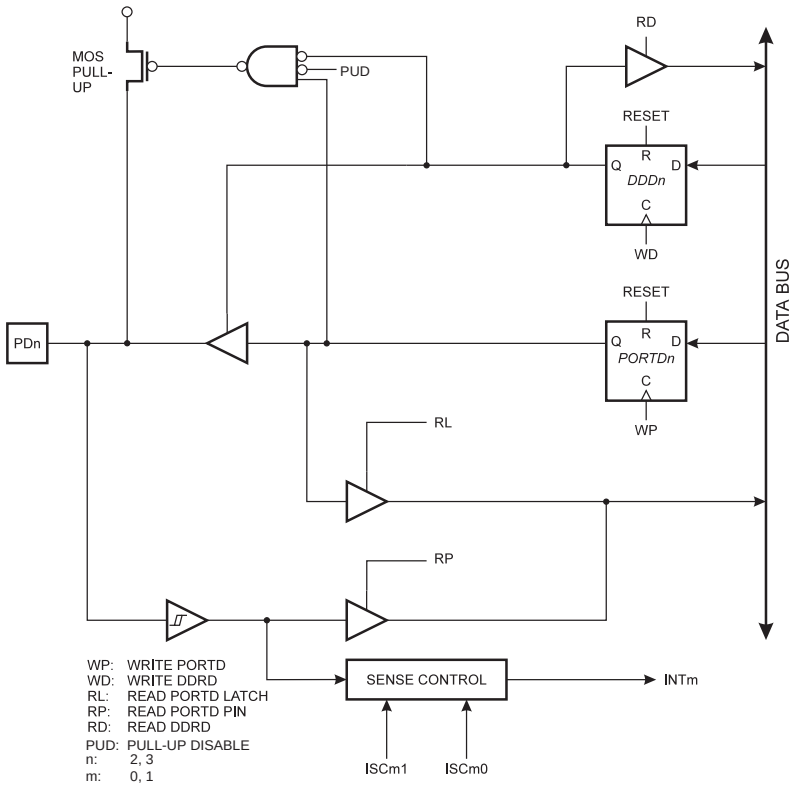


Figure 76. PORTD Schematic Diagram (Pins PD2 and PD3)



```
sbrc    temp1, SPMEN
rjmp    Wait_spm
ret
```

Program and Data Memory Lock Bits

The ATmega163 provides six Lock bits which can be left unprogrammed (“1”) or can be programmed (“0”) to obtain the additional features listed in Table 55. The Lock bits can only be erased to “1” with the Chip Erase command.

Table 55. Lock Bit Protection Modes

Memory Lock Bits			
LB mode	**LB1**	**LB2**	**Protection Type**
1	1	1	No memory lock features enabled for Parallel and Serial Programming.
2	0	1	Further programming of the Flash and EEPROM is disabled in Parallel and Serial Programming mode. The Fuse bits are locked in both Serial and Parallel Programming mode.(1)
3	0	0	Further programming and verification of the Flash and EEPROM is disabled in Parallel and Serial Programming mode. The Fuse bits are locked in both Serial and Parallel Programming mode.(1)
BLB0 mode	**BLB01**	**BLB02**	
1	1	1	No restrictions for SPM, LPM accessing the Application section.
2	0	1	SPM is not allowed to write to the Application section.
3	0	0	SPM is not allowed to write to the Application section, and LPM executing from the Boot Loader section is not allowed to read from the Application section.
4	1	0	LPM executing from the Boot Loader section is not allowed to read from the Application section.
BLB1 mode	**BLB11**	**BLB12**	
1	1	1	No restrictions for SPM, LPM accessing the Boot Loader section.
2	0	1	SPM is not allowed to write to the Boot Loader section.
3	0	0	SPM is not allowed to write to the Boot Loader section, and LPM executing from the Application section is not allowed to read from the Boot Loader section. If code executed from the Boot Section, the interrupts are disabled when BLB12 is programmed.
4	1	0	LPM executing from the Application section is not allowed to read from the Boot Loader section. If code executed from the Boot Section, the interrupts are disabled when BLB12 is programmed.

Note: 1. Program the Fuse bits before programming the Lock bits.

Fuse Bits

The ATmega163 has ten Fuse bits, divided in two groups. The Fuse High bits are BOOTSZ1..0 and BOOTRST, and the Fuse Low bits are BODLEVEL, BODEN, SPIEN, and CKSEL3..0.

- BOOTSZ1..0 select the size and start address of the Boot Flash section according to Table 51 on page 134. Default value is “11” (both unprogrammed).
- When BOOTRST is programmed (“0”), the Reset Vector is set to the start address of the Boot Flash section, as selected by the BOOTSZ fuses according to Table 51 on page 134. If the BOOTRST is unprogrammed (“1”), the Reset Vector is set to address $0000. Default value is unprogrammed (“1”).
- The BODLEVEL Fuse selects the Brown-out Detection Level and changes the Start-up times, according to Table 4 on page 24 and Table 5 on page 25, respectively. Default value is unprogrammed (“1”).
- When the BODEN Fuse is programmed (“0”), the Brown-out Detector is enabled. See “Reset and Interrupt Handling” on page 21. Default value is unprogrammed (“1”).
- When the SPIEN Fuse is programmed (“0”), Serial Program and Data Downloading are enabled. Default value is programmed (“0”). The SPIEN Fuse is not accessible in serial programming mode.
- CKSEL3..0 select the clock source and the start-up delay after reset, according to Table 1 on page 5 and Table 5 on page 25. Default value is “0010” (Internal RC Oscillator).

The status of the Fuse bits is not affected by Chip Erase. Note that the Fuse bits are locked if Lock bit1 (LB1) is programmed. Program the Fuse bits before programming the Lock bits.

Signature Bytes

All Atmel microcontrollers have a three-byte signature code which identifies the device. This code can be read in both serial and parallel mode. The three bytes reside in a separate address space.

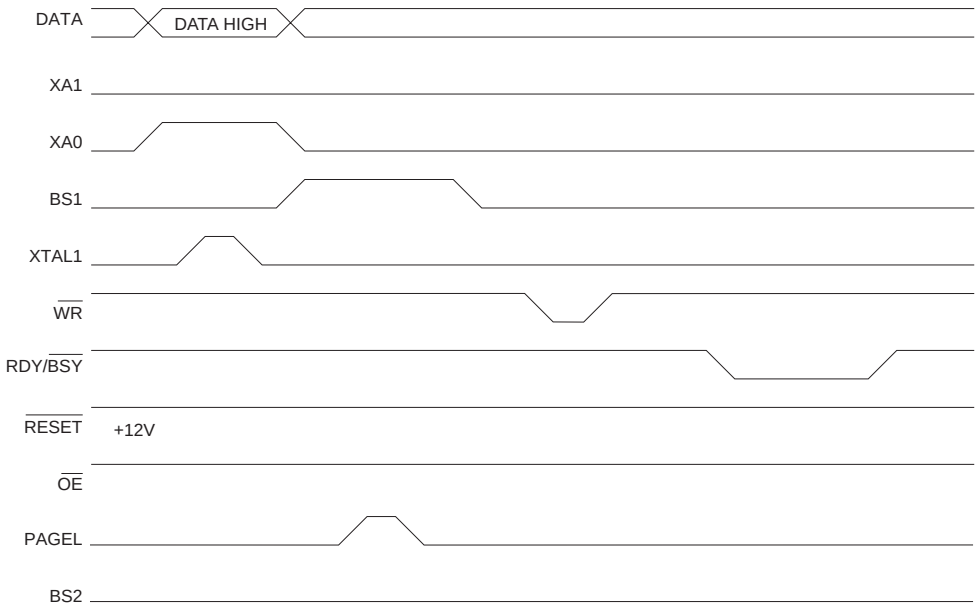
The ATmega163 the signature bytes are:

1. $000: $1E (indicates manufactured by Atmel).
2. $001: $94 (indicates 16KB Flash memory).
3. $002: $02 (indicates ATmega163 device when $001 is $94).

Calibration Byte

The ATmega163 has a one byte calibration value for the internal RC Oscillator. This byte resides in the high byte of address $000 in the signature address space. During Memory Programming, the external programmer must read this location and program it into a selected location in the normal Flash Program memory. At start-up, the user software must read this Flash location and write the value to the OSCCAL Register.

Figure 83. Programming the Flash Waveforms (continued)



Programming the EEPROM

The programming algorithm for the EEPROM Data Memory is as follows (refer to "Programming the Flash" on page 147 for details on Command, Address and Data loading):

1. A: Load Command "0001 0001".
2. H: Load Address High Byte ($00 - $01)
3. B: Load Address Low Byte ($00 - $FF)
4. E: Load Data Low Byte ($00 - $FF)

L: Write Data Low Byte

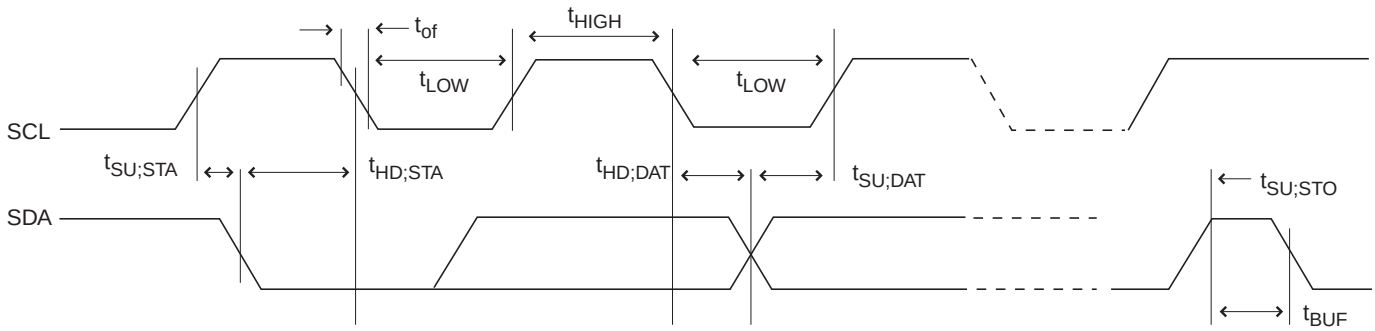
1. Set BS to "0". This selects low data.
2. Give WR a negative pulse. This starts programming of the data byte. RDY/BSY goes low.
3. Wait until to RDY/BSY goes high before programming the next byte. (See Figure 84 for signal waveforms)

The loaded command and address are retained in the device during programming. For efficient programming, the following should be considered.

- The command needs only be loaded once when writing or reading multiple memory locations.
- Address high byte needs only be loaded before programming a new 256 word page in the EEPROM.
- Skip writing the data value $FF, that is the contents of the entire EEPROM after a Chip Erase.

These considerations also applies to Flash, EEPROM and Signature bytes reading.

Figure 90. Two-wire Serial Bus Timing



Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
4	2.7 - 5.5V	ATmega163L-4AC ATmega163L-4PC	44A 40P6	Commercial (0°C to 70°C)
		ATmega163L-4AI ATmega163L-4PI	44A 40P6	Industrial (-40°C to 85°C)
8	4.0 - 5.5V	ATmega163-8AC ATmega163-8PC	44A 40P6	Commercial (0°C to 70°C)
		ATmega163-8AI ATmega163-8PI	44A 40P6	Industrial (-40°C to 85°C)

Package Type	
44A	44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)
40P6	40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)