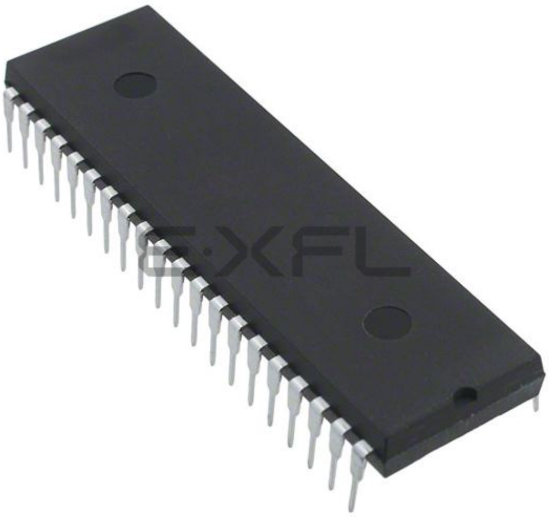
Microchip Technology - ATMEGA163L-4PI Datasheet



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	4MHz
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)	2.7V ~ 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/atmega163l-4pi

cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega163 provides the following features: 16K bytes of In-System Self-Programmable Flash, 512 bytes EEPROM, 1024 bytes SRAM, 32 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC, a programmable Watchdog Timer with internal Oscillator, a programmable serial UART, an SPI serial port, and four software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or Hardware Reset. In Power-save mode, the asynchronous Timer Oscillator continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions.

The On-chip ISP Flash can be programmed through an SPI serial interface or a conventional programmer. By installing a Self-Programming Boot Loader, the microcontroller can be updated within the application without any external components. The Boot Program can use any interface to download the application program in the Application Flash memory. By combining an 8-bit CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega163 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega163 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, In-Circuit Emulators, and evaluation kits.

Pin Descriptions

VCC

Digital supply voltage.

GND

Digital ground.

Port A (PA7..PA0)

Port A serves as the analog inputs to the A/D Converter.

Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers can sink 20mA and can drive LED displays directly. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tristated when a reset condition becomes active, even if the clock is not running.

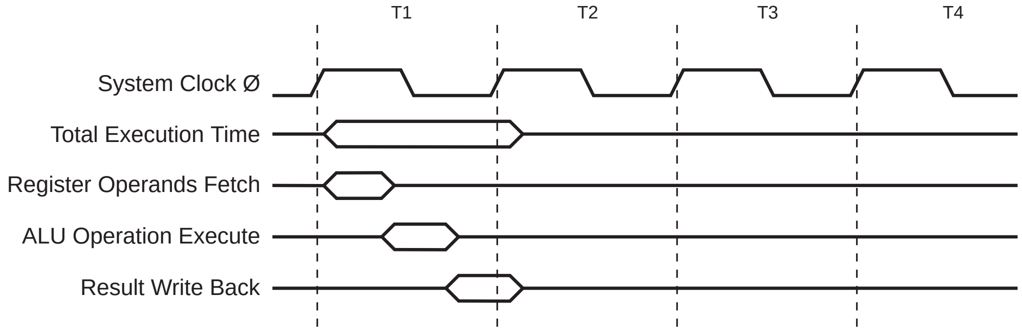
Port B (PB7..PB0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers can sink 20 mA. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. Port B also serves the functions of various special features of the ATmega83/163 as listed on page 117. The Port B pins are tristated when a reset condition becomes active, even if the clock is not running.

Port C (PC7..PC0)

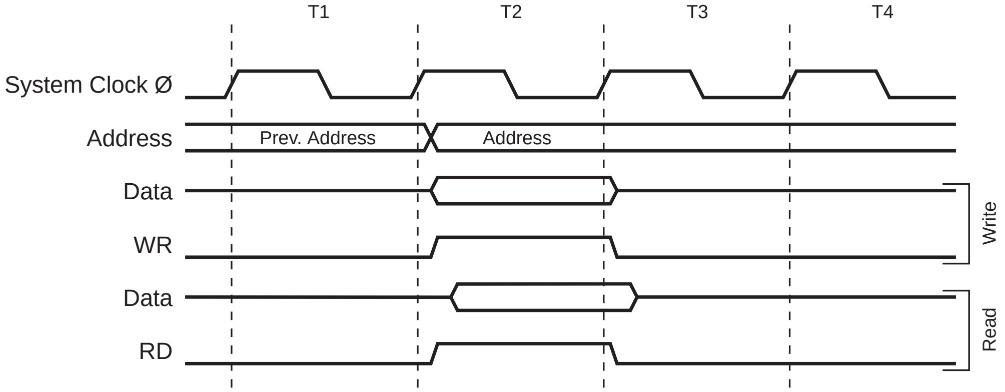
Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers can sink 20 mA. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tristated when a reset condition becomes active, even if the clock is not running.

Figure 22. Single Cycle ALU Operation



The internal data SRAM access is performed in two System Clock cycles as described in Figure 23.

Figure 23. On-chip Data SRAM Access Cycles



I/O Memory

The I/O space definition of the ATmega163 is shown in the following table:

Table 2. ATmega163 I/O Space [(1)]

I/O Address (SRAM Address)	Name	Function
$3F ($5F)	SREG	Status REGister
$3E ($5E)	SPH	Stack Pointer High
$3D ($5D)	SPL	Stack Pointer Low
$3B ($5B)	GIMSK	General Interrupt MaSK Register
$3A ($5A)	GIFR	General Interrupt Flag Register
$39 ($59)	TIMSK	Timer/Counter Interrupt MaSK Register
$38 ($58)	TIFR	Timer/Counter Interrupt Flag Register
$37 ($57)	SPMCR	SPM Control Register
$36 ($56)	TWCR	Two-wire Serial Interface Control Register
$35 ($55)	MCUCR	MCU general Control Register
$34 ($54)	MCUSR	MCU general Status Register
$33 ($53)	TCCR0	Timer/Counter0 Control Register

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}.

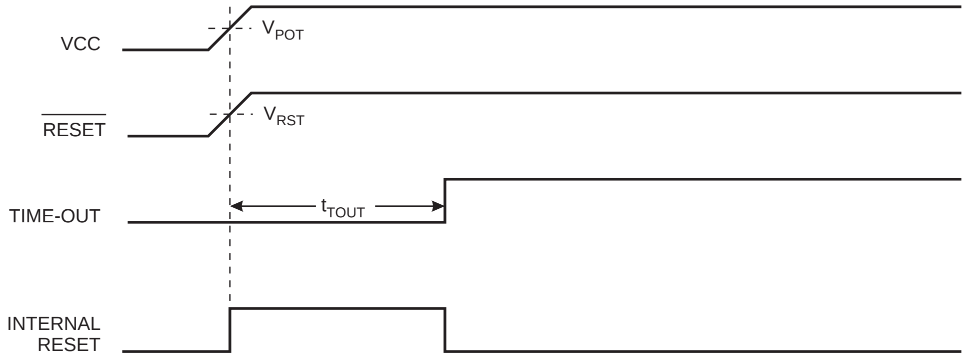
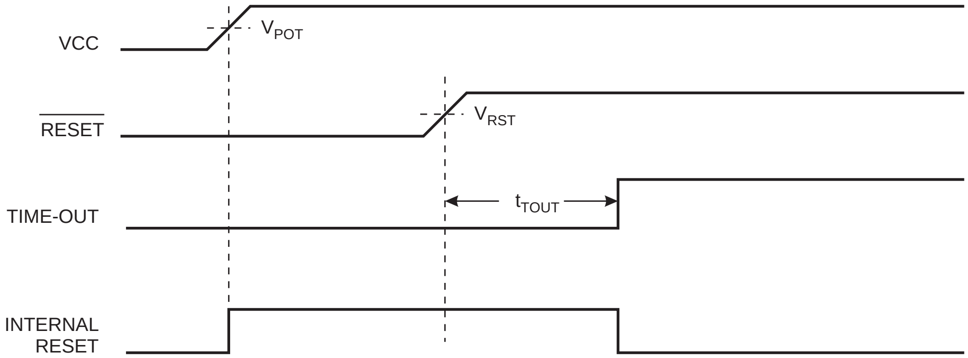


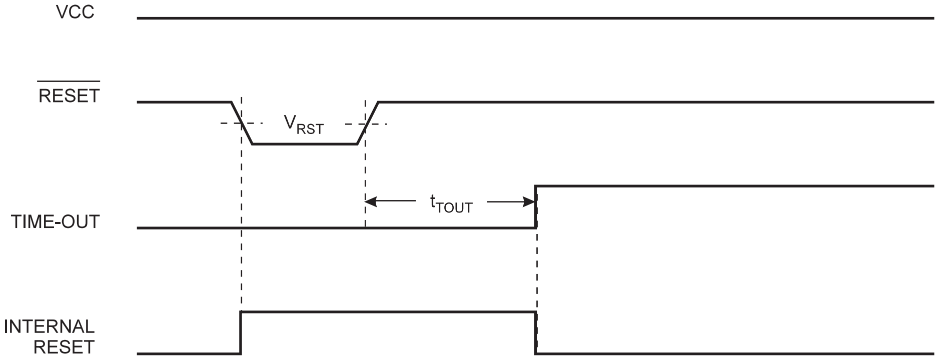
Figure 26. MCU Start-up, $\overline{\text{RESET}}$ Extended Externally



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.

Asynchronous Operation of Timer/Counter2

When Timer/Counter2 operates asynchronously, some considerations must be taken.

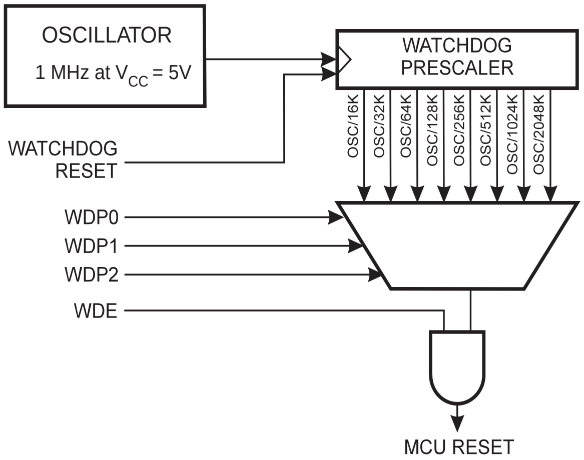
- Warning: When switching between asynchronous and synchronous clocking of Timer/Counter2, the Timer Registers TCNT2, OCR2, and TCCR2 might be corrupted. A safe procedure for switching clock source is:
  1. Disable the Timer/Counter2 interrupts by clearing OCIE2 and TOIE2.
  2. Select clock source by setting AS2 as appropriate.
  3. Write new values to TCNT2, OCR2, and TCCR2.
  4. To switch to asynchronous operation: Wait for TCN2UB, OCR2UB, and TCR2UB.
  5. Clear the Timer/Counter2 Interrupt Flags.
  6. Enable interrupts, if needed.
- The Oscillator is optimized for use with a 32.768 kHz watch crystal. Applying an external clock to the TOSC1 pin may result in incorrect Timer/Counter2 operation. The CPU main clock frequency must be more than four times the Oscillator frequency.
- When writing to one of the registers TCNT2, OCR2, or TCCR2, the value is transferred to a temporary register, and latched after two positive edges on TOSC1. The user should not write a new value before the contents of the temporary register have been transferred to its destination. Each of the three mentioned registers have their individual temporary register, which means that e.g. writing to TCNT2 does not disturb an OCR2 write in progress. To detect that a transfer to the destination register has taken place, the Asynchronous Status Register – ASSR has been implemented.
- When entering Power-save mode after having written to TCNT2, OCR2, or TCCR2, the user must wait until the written register has been updated if Timer/Counter2 is used to wake up the device. Otherwise, the MCU will enter sleep mode before the changes are effective. This is particularly important if the Output Compare2 interrupt is used to wake up the device, since the output compare function is disabled during writing to OCR2 or TCNT2. If the write cycle is not finished, and the MCU enters sleep mode before the OCR2UB bit returns to zero, the device will never receive a compare match interrupt, and the MCU will not wake up.
- If Timer/Counter2 is used to wake the device up from Power-save mode, precautions must be taken if the user wants to re-enter Power-save mode: The interrupt logic needs one TOSC1 cycle to be Reset. If the time between wake-up and re-entering Power-save mode is less than one TOSC1 cycle, the interrupt will not occur, and the device will fail to wake up. If the user is in doubt whether the time before re-entering Power-save is sufficient, the following algorithm can be used to ensure that one TOSC1 cycle has elapsed:
  1. Write a value to TCCR2, TCNT2, or OCR2.
  2. Wait until the corresponding Update Busy Flag in ASSR returns to zero.
  3. Enter Power-save mode.
- When the asynchronous operation is selected, the 32.768 kHz Oscillator for Timer/Counter2 is always running, except in Power-down mode. After a Power-up Reset or Wake-up from Power-down, the user should be aware of the fact that this Oscillator might take as long as one second to stabilize. The user is advised to wait for at least one second before using Timer/Counter2 after Power-up or wake-up from Power-down. The contents of all Timer/Counter2 Registers must be considered lost after a wake-up from Power-down due to unstable clock signal upon startup.
- Description of wake-up from Power-save mode when the Timer is clocked asynchronously: When the interrupt condition is met, the wake-up process is started

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:

set), the 9th bit is one for an address byte and zero for a data byte, whereas the stop bit is always high.

The following procedure should be used to exchange data in Multi-Processor Communication mode:

1. All Slave MCUs are in Multi-Processor Communication mode (MPCM in UCSRA is set).
2. The Master MCU sends an address byte, and all slaves receive and read this byte. In the Slave MCUs, the RXC Flag in UCSRA will be set as normal.
3. Each Slave MCU reads the UDR Register and determines if it has been selected. If so, it clears the MPCM bit in UCSRA, otherwise it waits for the next address byte.
4. For each received data byte, the receiving MCU will set the Receive Complete Flag (RXC in UCSRA). In 8-bit mode, the receiving MCU will also generate a Framing Error (FE in UCSRA set), since the stop bit is zero. The other slave MCUs, which still have the MPCM bit set, will ignore the data byte. In this case, the UDR Register and the RXC or FE Flags will not be affected.
5. After the last byte has been transferred, the process repeats from step 2.

UART Control

UART I/O Data Register – UDR

Bit	7	6	5	4	3	2	1	0	
$0C ($2C)	MSB							LSB	UDR
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 UDR Register is actually two physically separate registers sharing the same I/O address. When writing to the register, the UART Transmit Data Register is written. When reading from UDR, the UART Receive Data Register is read.

UART Control and Status Register A – UCSRA

Bit	7	6	5	4	3	2	1	0	
$0B ($2B)	RXC	TXC	UDRE	FE	OR	–	U2X	MPCM	UCSRA
Read/Write	r	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – RXC: UART Receive Complete**

This bit is set (one) when a received character is transferred from the Receiver Shift Register to UDR. The bit is set regardless of any detected framing errors. When the RXCIE bit in UCR is set, the UART Receive Complete interrupt will be executed when RXC is set(one). RXC is cleared by reading UDR. When interrupt-driven data reception is used, the UART Receive Complete Interrupt routine must read UDR in order to clear RXC, otherwise a new interrupt will occur once the interrupt routine terminates.

- **Bit 6 – TXC: UART Transmit Complete**

This bit is set (one) when the entire character (including the stop bit) in the Transmit Shift Register has been shifted out and no new data has been written to UDR. This Flag is especially useful in half-duplex communications interfaces, where a transmitting application must enter receive mode and free the communications bus immediately after completing the transmission.

- **Bit 0 – TWIE: Two-wire Serial Interface Interrupt Enable**

When this bit is enabled, and the I-bit in SREG is set, the Two-wire Serial Interface interrupt will be activated for as long as the TWINT Flag is high.

The TWCR is used to control the operation of the Two-wire Serial Interface. It is used to enable the Two-wire Serial Interface, to initiate a Master access by applying a START condition to the bus, to generate a receiver acknowledge, to generate a stop condition, and to control halting of the bus while the data to be written to the bus are written to the TWDR. It also indicates a write collision if data is attempted written to TWDR while the register is inaccessible.

The Two-wire Serial Interface Status Register – TWSR

Bit	7	6	5	4	3	2	1	0	
$01 ($21)	TWS7	TWS6	TWS5	TWS4	TWS3	–	–	–	TWSR
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	1	1	1	1	1	0	0	0	

- **Bits 7..3 – TWS: Two-wire Serial Interface Status**

These five bits reflect the status of the Two-wire Serial Interface logic and the Two-wire Serial Bus.

- **Bits 2..0 – Res: Reserved bits**

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

The TWSR is read only. It contains a status code which reflects the status of the Two-wire Serial Interface logic and the Two-wire Serial Bus. There are 26 possible status codes. When TWSR contains $F8, no relevant state information is available and no Two-wire Serial Interface interrupt is requested. A valid status code is available in TWSR one CPU clock cycle after the Two-wire Serial Interface Interrupt Flag (TWINT) is set by hardware and is valid until one CPU clock cycle after TWINT is cleared by software. Table 32 to Table 36 give the status information for the various modes.

The Two-wire Serial Interface Data Register – TWDR

Bit	7	6	5	4	3	2	1	0	
$03 ($23)	TWD7	TWD6	TWD5	TWD4	TWD3	TWD2	TWD1	TWD0	TWDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	1	1	1	1	1	1	1	

- **Bits 7..0 – TWD: Two-wire Serial Interface Data Register**

These eight bits constitute the next data byte to be transmitted, or the latest data byte received on the Two-wire Serial Bus.

In Transmit mode, TWDR contains the next byte to be transmitted. In Receive mode, the TWDR contains the last byte received. It is writeable while the Two-wire Serial Interface is not in the process of shifting a byte. This occurs when the Two-wire Serial Interface Interrupt Flag (TWINT) is set by hardware. Note that the Data Register cannot be initialized by the user before the first interrupt occurs. The data in TWDR remain stable as long as TWINT is set. While data is shifted out, data on the bus is simultaneously shifted in. TWDR always contains the last byte present on the bus, except after a wake up from ADC Noise Reduction mode, Power-down mode, or Power-save mode by the Two-wire Serial Interface interrupt. For example, in the case of a lost bus arbitration, no data is lost in the transition from Master to Slave. Handling of the ACK Flag is controlled automatically by the Two-wire Serial Interface logic, the CPU cannot access the ACK bit directly.

After a repeated START condition (state $10), the Two-wire Serial Interface may switch to the Master Transmitter mode by loading TWDR with SLA+W or access a new Slave as Master Receiver or Transmitter.

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

Slave Receiver Mode

In the Slave Receiver mode, a number of data bytes are received from a Master Transmitter (see Figure 54). To initiate the Slave Receiver mode, TWAR and TWCR must be initialized as follows:

Table 30. TWAR: Slave Receiver Mode Initialization

TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE
Value	Device’s Own Slave Address							

The upper seven bits are the address to which the Two-wire Serial Interface will respond when addressed by a Master. If the LSB is set, the Two-wire Serial Interface will respond to the general call address ($00), otherwise it will ignore the general call address.

Table 31. WCR: Slave Receiver Mode Initialization

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

TWEN must be set to enable the Two-wire Serial Interface. The TWEA bit must be set to enable the acknowledgement of the device’s own slave address or the general call address. TWSTA and TWSTO must be cleared.

When TWAR and TWCR have been initialized, the Two-wire Serial Interface waits until it is addressed by its own slave address (or the general call address if enabled) followed by the Data Direction bit which must be “0” (write) for the Two-wire Serial Interface to operate in the Slave Receiver mode. After its own slave address and the write bit have been received, the Two-wire Serial Interface Interrupt Flag is set and a valid status code can be read from TWSR. The status code is used to determine the appropriate software action. The appropriate action to be taken for each status code is detailed in Table 34. The Slave Receiver mode may also be entered if arbitration is lost while the Two-wire Serial Interface is in the Master mode (see states $68 and $78).

If the TWEA bit is reset during a transfer, the Two-wire Serial Interface will return a “Not Acknowledge” (“1”) to SDA after the next received data byte. While TWEA is Reset, the Two-wire Serial Interface does not respond to its own slave address. However, the Two-wire Serial Bus is still monitored and address recognition may resume at any time by setting TWEA. This implies that the TWEA bit may be used to temporarily isolate the Two-wire Serial Interface from the Two-wire Serial Bus.

In ADC Noise Reduction mode, Power-down mode, and Power-save mode, the clock system to the Two-wire Serial Interface is turned off. If the Slave Receive mode is enabled, the interface can still acknowledge a general call and its own slave address by using the Two-wire Serial Bus clock as a clock source. The part will then wake-up from sleep and the Two-wire Serial Interface will hold the SCL clock wil low during the wake-up and until the TWINT Flag is cleared.

Note that the Two-wire Serial Interface Data Register – TWDR – does not reflect the last byte present on the bus when waking up from these sleep modes.

Assembly code illustrating operation of the Slave Receiver mode is given at the end of the TWI section.

Slave Transmitter Mode

In the Slave Transmitter mode, a number of data bytes are transmitted to a Master Receiver (see Figure 55). The transfer is initialized as in the Slave Receiver mode. When TWAR and TWCR have been initialized, the Two-wire Serial Interface waits until it is addressed by its own slave address (or the general call address if enabled) followed by the Data Direction bit which must be “1” (read) for the Two-wire Serial Interface to operate in the Slave Transmitter mode. After its own slave address and the read bit have been received, the Two-wire Serial Interface Interrupt Flag is set and a valid status code can be read from TWSR. The status code is used to determine the appropriate software action. The appropriate action to be taken for each status code is detailed in Table 35. The slave transmitter mode may also be entered if arbitration is lost while the Two-wire Serial Interface is in the Master mode (see state $B0).

If the TWEA bit is reset during a transfer, the Two-wire Serial Interface will transmit the last byte of the transfer and enter state $C0 or state $C8. the Two-wire Serial Interface is switched to the not addressed Slave mode, and will ignore the Master if it continues the transfer. Thus the Master Receiver receives all “1” as serial data. While TWEA is reset, the Two-wire Serial Interface does not respond to its own slave address. However, the Two-wire Serial Bus is still monitored and address recognition may resume at any time by setting TWEA. This implies that the TWEA bit may be used to temporarily isolate the Two-wire Serial Interface from the Two-wire Serial Bus.

Assembly code illustrating operation of the Slave Receiver mode is given at the end of the TWI section.

Miscellaneous States

There are two status codes that do not correspond to a defined Two-wire Serial Interface state, see Table 36.

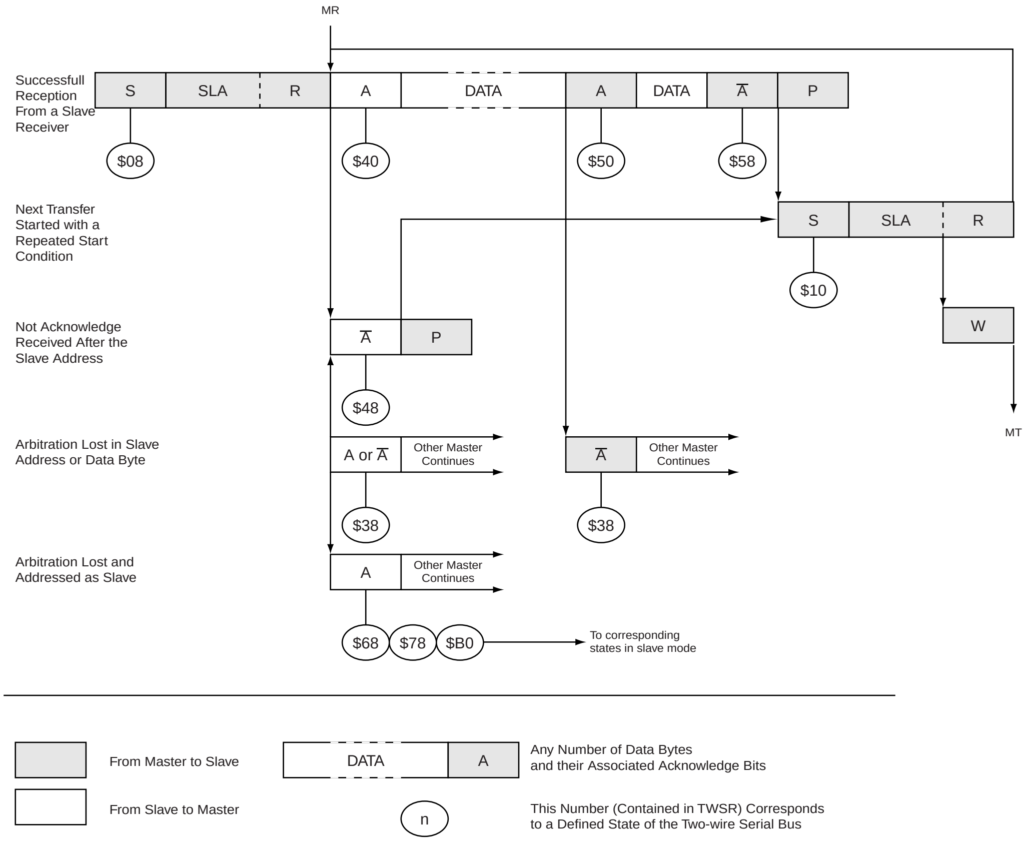
Status $F8 indicates that no relevant information is available because the Two-wire Serial Interface Interrupt Flag (TWINT) is not set yet. This occurs between other states, and when the Two-wire Serial Interface is not involved in a serial transfer.

Status $00 indicates that a bus error has occured during a Two-wire Serial Bus transfer. A bus error occurs when a START or STOP condition occurs at an illegal position in the format frame. Examples of such illegal positions are during the serial transfer of an address byte, a data byte or an acknowledge bit. When a bus error occurs, TWINT is set. To recover from a bus error, the TWSTO Flag must set and TWINT must be cleared by writing a logic one to it. This causes the Two-wire Serial Interface to enter the not addressed Slave mode and to clear the TWSTO Flag (no other bits in TWCR are affected). The SDA and SCL lines are released and no STOP condition is transmitted.

Table 33. Status Codes for Master Receiver Mode

		Application Software Response					
			To TWCR				
Status Code (TWSR)	Status of the Two-wire Serial Bus and Two-wire Serial Interface hardware	To/from TWDR	STA	STO	TWINT	TWEA	Next Action Taken by Two-wire Serial Interface Hardware
$08	A START condition has been transmitted	Load SLA+R	X	0	1	X	SLA+R will be transmitted ACK or NOT ACK will be received
$10	A repeated START condition has been transmitted	Load SLA+R or Load SLA+W	X X	0 0	1 1	X X	SLA+R will be transmitted ACK or NOT ACK will be received SLA+W will be transmitted Logic will switch to Master Transmitter mode.
$38	Arbitration lost in SLA+R or NOT ACK bit	No TWDR action or No TWDR actio	0 1	0 0	1 1	X X	Two-wire Serial Bus will be released and not addressed Slave mode will be entered A START condition will be transmitted when the bus becomes free
$40	SLA+R has been transmitted; ACK has been received	No TWDR action or No TWDR action	0 0	0 0	1 1	0 1	Data byte will be received and NOT ACK will be returned Data byte will be received and ACK will be returned
$48	SLA+R has been transmitted; NOT ACK has been received	No TWDR action or No TWDR action or No TWDR action	1 0 1	0 1 1	1 1 1	X X X	Repeated START will be transmitted STOP condition will be transmitted and TWSTO Flag will be Reset STOP condition followed by a START condition will be transmitted and TWSTO Flag will be Reset
$50	Data byte has been received; ACK has been returned	Read data byte or Read data byte	0 0	0 0	1 1	0 1	Data byte will be received and NOT ACK will be returned Data byte will be received and ACK will be returned
$58	Data byte has been received; NOT ACK has been returned	Read data byte or Read data byte or Read data byte	1 0 1	0 1 1	1 1 1	X X X	Repeated START will be transmitted STOP condition will be transmitted and TWSTO Flag will be Reset STOP condition followed by a START condition will be transmitted and TWSTO Flag will be Reset

Figure 53. Formats and States in the Master Receiver Mode


MR
Successfull Reception From a Slave Receiver
S
SLA
R
A
DATA
A
DATA
A̅
P
$08
$40
$50
$58
Next Transfer Started with a Repeated Start Condition
S
SLA
R
$10
W
Not Acknowledge Received After the Slave Address
A̅
P
$48
MT
Arbitration Lost in Slave Address or Data Byte
A or A̅
Other Master Continues
A̅
Other Master Continues
$38
$38
Arbitration Lost and Addressed as Slave
A
Other Master Continues
$68
$78
$B0
To corresponding states in slave mode
From Master to Slave
DATA
A
Any Number of Data Bytes and their Associated Acknowledge Bits
From Slave to Master
n
This Number (Contained in TWSR) Corresponds to a Defined State of the Two-wire Serial Bus

Analog Comparator Multiplexed Input

It is possible to select any of the PA7..0 (ADC7..0) pins to replace the negative input to the Analog Comparator. The ADC multiplexer is used to select this input, and consequently, the ADC must be switched off to utilize this feature. If the Analog Comparator Multiplexer Enable bit (ACME in SFIOR) is set (one) and the ADC is switched off (ADEN in ADCSR is zero), MUX2..0 in ADMUX select the input pin to replace the negative input to the Analog Comparator, as shown in Table 38. If ACME is cleared (zero) or ADEN is set (one), PB3 (AIN1) is applied to the negative input to the Analog Comparator.

Table 38. Analog Comparator Multiplexed Input

ACME	ADEN	MUX2..0	Analog Comparator Negative Input
0	x	xxx	AIN1
1	1	xxx	AIN1
1	0	000	ADC0
1	0	001	ADC1
1	0	010	ADC2
1	0	011	ADC3
1	0	100	ADC4
1	0	101	ADC5
1	0	110	ADC6
1	0	111	ADC7

When initiating a conversion by setting the ADSC bit in ADCSR, the conversion starts at the following rising edge of the ADC clock cycle.

A normal conversion takes 13 ADC clock cycles. In certain situations, the ADC needs more clock cycles to initalization and minimize offset errors. Extended conversions take 25 ADC clock cycles and occur as the first conversion after the ADC is switched on (ADEN in ADCSR is set). Additionally, when changing voltage reference, the user may improve accuracy by disregarding the first conversion result after the reference or MUX setting was changed.

The actual sample-and-hold takes place 1.5 ADC clock cycles after the start of a normal conversion and 13.5 ADC clock cycles after the start of an extended conversion. When a conversion is complete, the result is written to the ADC Data Registers, and ADIF is set. In Single Conversion mode, ADSC is cleared simultaneously. The software may then set ADSC again, and a new conversion will be initated on the first rising ADC clock edge. In Free Running mode, a new conversion will be started immediately after the conversion completes, while ADSC remains high. Using Free Running mode and an ADC clock frequency of 200 kHz gives the lowest conversion time with a maximum resolution, 65 µs, equivalent to 15 kSPS. For a summary of conversion times, see Table 39.

Figure 59. ADC Timing Diagram, Extended Conversion (Single Conversion Mode)

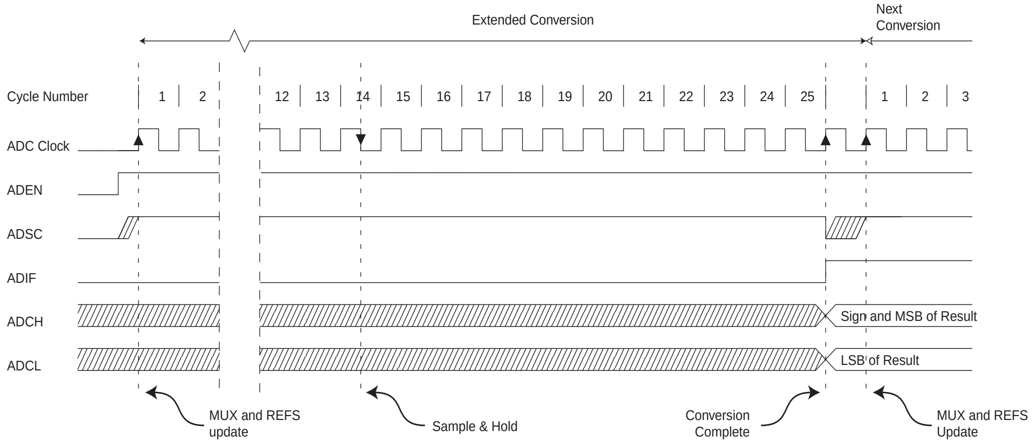
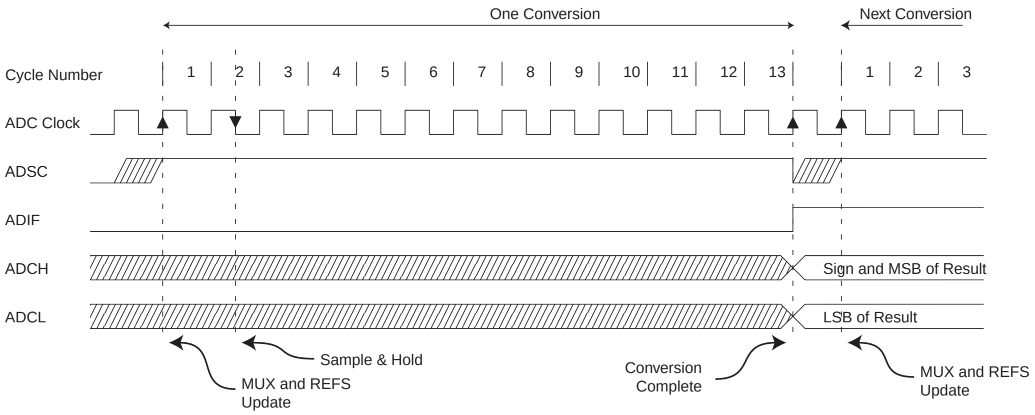


Figure 60. ADC Timing Diagram, Single Conversion



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 67. PORTB Schematic Diagram (Pin PB5)

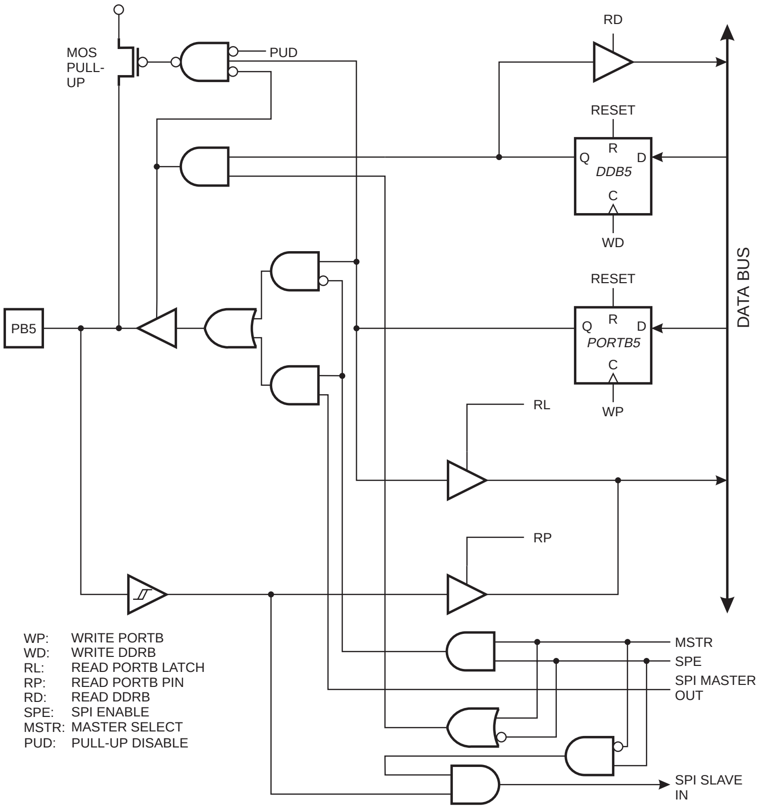
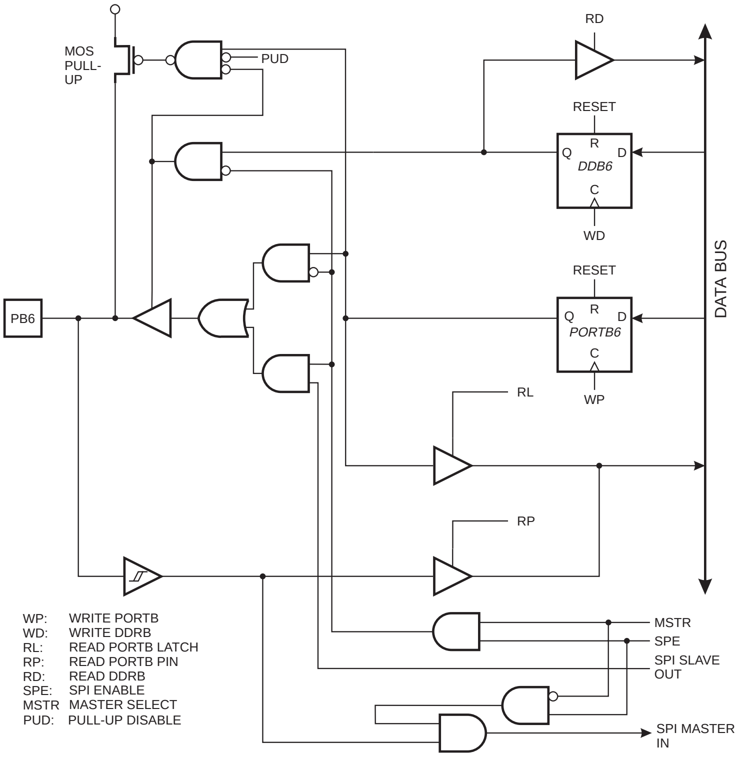


Figure 68. PORTB Schematic Diagram (Pin PB6)



Memory Programming

Boot Loader Support

The ATmega163 provides a mechanism for Programming and Re-programming code by the MCU itself. This feature allows flexible application software updates, controlled by the MCU using a Flash-resident Boot Loader program. This makes it possible to program the AVR in a target system without access to its SPI pins. The Boot Loader program can use any available data interface and associated protocol, such as UART serial bus interface, to input or output program code, and write (program) that code into the Flash memory, or read the code from the Flash memory.

The ATmega163 Flash memory is organized in two main sections:

- The Application Flash section
- The Boot Loader Flash section

The Application Flash section and the Boot Loader Flash section have seperate Boot Lock bits. Thus the user can select different levels of protection for the two sections. The Store Program Memory (SPM) instruction can only be executed from the Boot Loader Flash section.

The Program Flash memory in ATmega163 is divided into 128 pages of 64 words each. The Boot Loader Flash section is located at the high address space of the Flash, and can be configured through the BOOTSZ Fuses as shown in Table 51.

Table 51. Boot Size Configuration

BOOTSZ1	BOOTSZ0	Boot Size	Pages	Application Flash Addresses	Boot Flash Addresses
1	1	128 Words	2	$0000 - $1F7F	$1F80 - $1FFF
1	0	256 Words	4	$0000 - $1EFF	$1F00 - $1FFF
0	1	512 Words	8	$0000 - $1DFF	$1E00 - $1FFF
0	0	1024 Words	16	$0000 - $1BFF	$1C00 - $1FFF

Bit	7	6	5	4	3	2	1	0
Rd	–	–	BLB12	BLB11	BLB02	BLB01	LB2	LB1

The algorithm for reading the Fuse Low bits is similar to the one described above for reading the Lock bits. To read the Fuse Low bits, load the Z-pointer with $0000 and set the BLBSET and SPMEN bits in SPMCR. When an LPM instruction is executed within five cycles after the BLBSET and SPMEN bits are set in the SPMCR, the value of the Fuse Low bits will be loaded in the destination register as shown below.

Bit	7	6	5	4	3	2	1	0
Rd	BODLEVEL	BODEN	SPIEN	–	CKSEL3	CKSEL2	CKSEL1	CKSEL0

Similarly, when reading the Fuse High bits, load $0003 in the Z-pointer. When an LPM instruction is executed within five cycles after the BLBSET and SPMEN bits are set in the SPMCR, the value of the Fuse High bits will be loaded in the destination register as shown below.

Bit	7	6	5	4	3	2	1	0
Rd	–	–	–	–	–	BOOTSZ1	BOOTSZ0	BOOTRST

Fuse and Lock bits that are programmed, will be read as zero. Fuse and Lock bits that are unprogrammed, will be read as one.

In all cases, the read value of unused bit positions are undefined.

EEPROM Write Prevents Writing to SPMCR

Note that an EEPROM write operation will block all software programming to Flash. Reading the Fuses and Lock bits from software will also be prevented during the EEPROM write operation. It is recommended that the user checks the status bit (EEWE) in the EECR Register and verifies that the bit is cleared before writing to the SPMCR Register. If EEPROM writing is performed inside an interrupt routine, the user software should disable that interrupt before checking the EEWE status bit.

Addressing the Flash During Self-Programming

The Z-pointer is used to address the SPM commands.

Bit	15	14	13	12	11	10	9	8
ZH (R31)	**Z15**	**Z14**	**Z13**	**Z12**	**Z11**	**Z10**	**Z9**	**Z8**
ZL (R30)	**Z7**	**Z6**	**Z5**	**Z4**	**Z3**	**Z2**	**Z1**	**Z0**
	7	6	5	4	3	2	1	0

Z15:Z14 always ignored

Z13:Z7 page select, for page erase and page write

Z6:Z1 word select, for filling temp buffer (must be zero during page write operation)

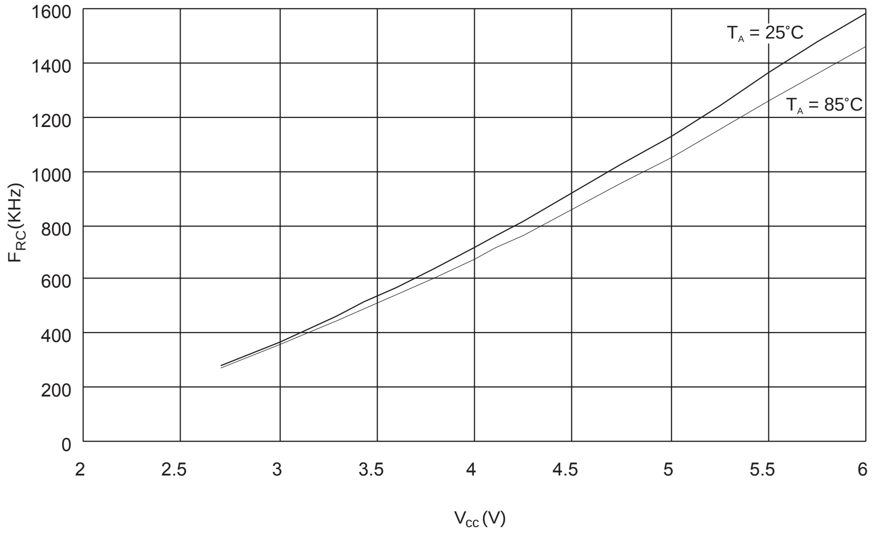
Z0 should be zero for all SPM commands, byte select for the LPM instruction.

The only operation that does not use the Z-pointer is Setting the Boot Loader Lock bits. The content of the Z-pointer is ignored and will have no effect on the operation.

Note that the Page Erase and Page Write operation is addressed independently. Therefore it is of major importance that the Boot Loader software addresses the same page in both the page erase and page write operation.

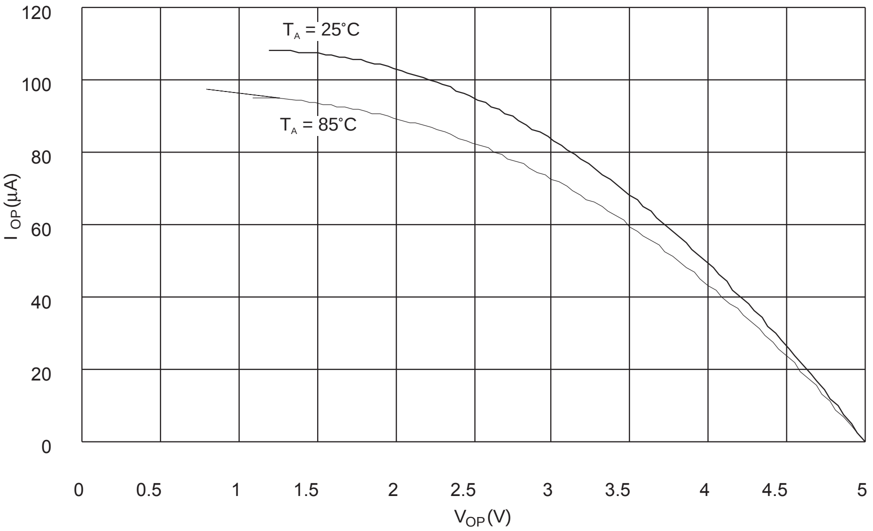
The LPM instruction also uses the Z-pointer to store the address. Since this instruction addresses the Flash byte-by-byte, also the LSB (bit Z0) of the Z-pointer is used. See page 15 for a detailed description.

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)



Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 487-2600

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
TEL (41) 26-426-5555
FAX (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
TEL (33) 2-40-18-18-18
FAX (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
TEL (33) 4-42-53-60-00
FAX (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
TEL (44) 1355-803-000
FAX (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
TEL (49) 71-31-67-0
FAX (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
TEL (33) 4-76-58-30-00
FAX (33) 4-76-58-34-80

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

© Atmel Corporation 2003.

Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

ATMEL® and AVR® are the registered trademarks of Atmel.

Other terms and product names may be the trademarks of others.

Printed on recycled paper.