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	SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	32
Program Memory Size	128KB (64K x 16)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega103l-4ai

Pin Descriptions

VCC Supply voltage.

GND Ground.

Port A (PA7..PA0) Port A is an 8-bit bi-directional I/O port. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers can sink 20 mA 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.

Port A serves as Multiplexed Address/Data bus when using external SRAM.

The Port A pins are tri-stated 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. 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.

The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port C (PC7..PC0) Port C is an 8-bit output port. The Port C output buffers can sink 20 mA.

Port C also serves as Address output when using external SRAM.

Since Port C is an output only port, the Port C pins are *not* tri-stated when a reset condition becomes active.

Port D (PD7..PD0) Port D is an 8-bit bi-directional I/O port with internal pull-up resistors. The Port D output buffers can sink 20 mA. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated.

Port D also serves the functions of various special features.

The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port E (PE7..PE0) Port E is an 8-bit bi-directional I/O port with internal pull-up resistors. The Port E output buffers can sink 20 mA. As inputs, Port E pins that are externally pulled low will source current if the pull-up resistors are activated.

Port E also serves the functions of various special features.

The Port E pins are tri-stated when a reset condition becomes active, even if the clock is not running

Port F (PF7..PF0) Port F is an 8-bit input port. Port F also serves as the analog inputs for the ADC.

$\overline{\text{RESET}}$ Reset input. An external reset is generated by a low level on the $\overline{\text{RESET}}$ pin. Reset pulses longer than 50 ns will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset.

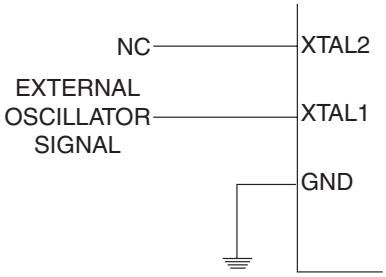
XTAL1 Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

XTAL2 Output from the inverting Oscillator amplifier.

External Clock

To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 3.

Figure 3. External Clock Drive Configuration



Timer Oscillator

For the Timer Oscillator pins, TOSC1 and TOSC2, the crystal is connected directly between the pins. No external capacitors are needed. The Oscillator is optimized for use with a 32,768 Hz watch crystal. Applying an external clock source to TOSC1 is not recommended.

Program execution continues at address PC + k + 1. The relative address k is -2048 to 2047.

EEPROM Data Memory

The EEPROM memory is organized as a separate Data space in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU is described on page 57 specifying the EEPROM Address Register, the EEPROM Data Register and the EEPROM Control Register.

Memory Access Times and Instruction Execution Timing

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

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

Figure 20 shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast-access Register File concept. This is the basic pipelining concept to obtain up to 1 MIPS per MHz with the corresponding unique results for functions per cost, functions per clocks and functions per power unit.

Figure 20. The Parallel Instruction Fetches and Instruction Executions

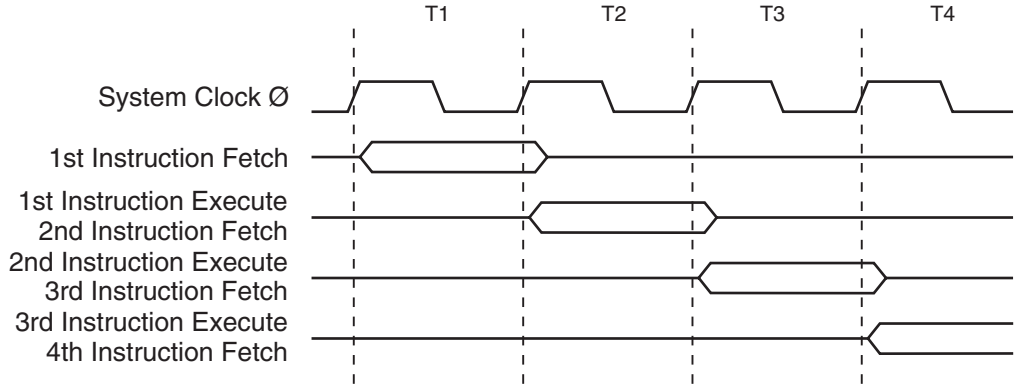
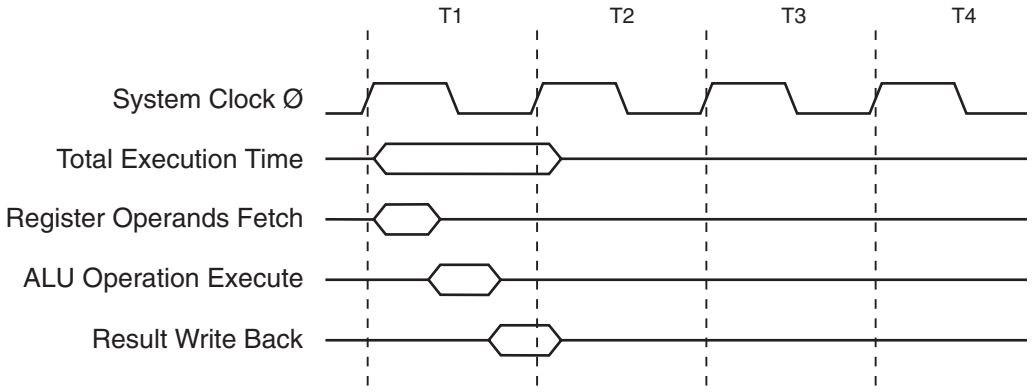


Figure 21 shows the internal timing concept for the Register File. In a single clock cycle, an ALU operation using two register operands is executed and the result is stored back to the destination register.

Figure 21. Single Cycle ALU Operation



Timer/Counter Interrupt Flag Register – TIFR

Bit	7	6	5	4	3	2	1	0	
$36 ($56)	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	TIFR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – OCF2: Output Compare Flag 2:**

The OCF2 bit is set (one) when compare match occurs between Timer/Counter2 and the data in OCR2 – Output Compare Register 2. OCF2 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF2 is cleared by writing a logical “1” to the flag. When the I-bit in SREG, and OCIE2 (Timer/Counter2 Compare Interrupt Enable) and the OCF2 are set (one), the Timer/Counter2 Output Compare 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 logical “1” to the flag. When the I-bit in SREG, and TOIE2 (Timer/Counter1 Overflow Interrupt Enable) and TOV2 are set (one), the Timer/Counter2 Overflow interrupt is executed. In PWM mode, this bit is set when Timer/Counter2 advances from $00.

- **Bit 5 – ICF1: Input Capture Flag 1**

The ICF1 bit is set (one) to flag an Input Capture event, indicating that the Timer/Counter1 value has been transferred to the Input Capture Register (ICR1). ICF1 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ICF1 is cleared by writing a logical “1” to the flag. When the SREG I-bit, TICIE1 (Timer/Counter1 Input Capture Interrupt Enable) and ICF1 are set (one), the Timer/Counter1 Capture interrupt is executed.

- **Bit 4 – OCF1A: Output Compare Flag 1A**

The OCF1A bit is set (one) when compare match occurs between the Timer/Counter1 and the data in OCR1A – Output Compare Register 1A. OCF1A is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF1A is cleared by writing a logical “1” to the flag. When the I-bit in SREG and OCIE1A (Timer/Counter1 Compare Interrupt Enable) and the OCF1A are set (one), the Timer/Counter1 Compare A Match interrupt is executed.

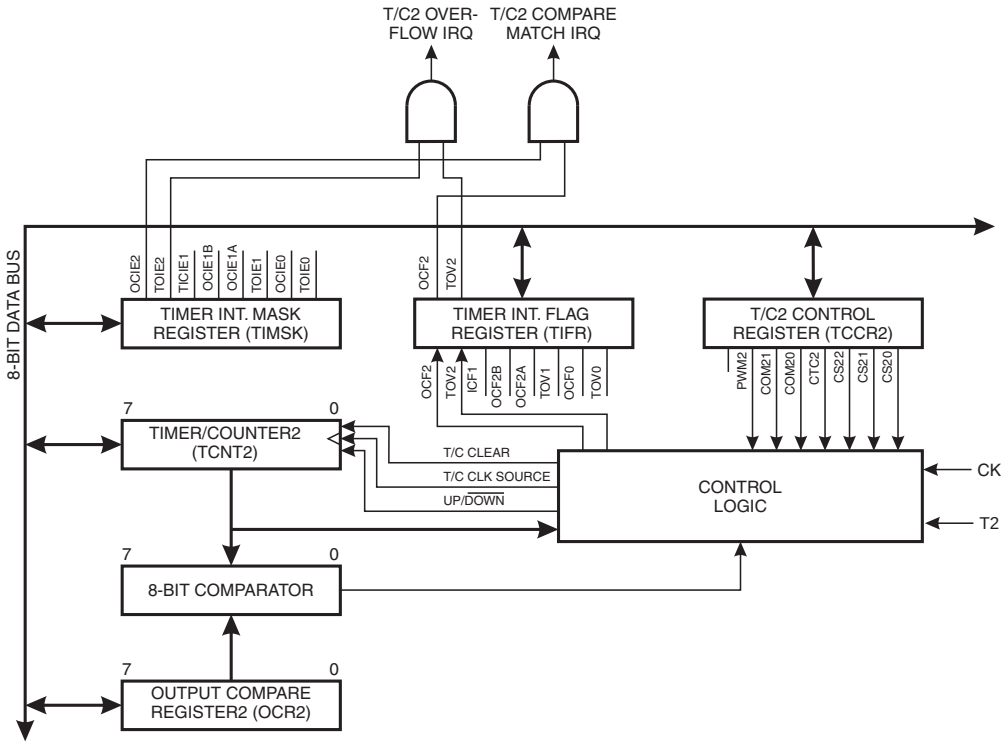
- **Bit 3 – OCF1B: Output Compare Flag 1B**

The OCF1B bit is set (one) when compare match occurs between the Timer/Counter1 and the data in OCR1B – Output Compare Register 1B. OCF1B is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF1B is cleared by writing a logical “1” to the flag. When the I-bit in SREG and OCIE1B (Timer/Counter1 Compare Match Interrupt Enable) and the OCF1B are set (one), the Timer/Counter1 Compare B Match interrupt is executed.

- **Bit 2 – TOV1: Timer/Counter1 Overflow Flag**

The TOV1 is set (one) when an overflow occurs in Timer/Counter1. TOV1 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV1 is cleared by writing a logical “1” to the flag. When the I-bit in SREG and TOIE1

Figure 31. Timer/Counter2 Block Diagram



The 8-bit Timer/Counter0 can select clock source from PCK0 or prescaled PCK0. The 8-bit Timer/Counter2 can select clock source from CK, prescaled CK or an external pin. Both Timer/Counters can be stopped as described in the specification for the Timer/Counter Control Registers – TCCR0 and TCCR2.

The different Status Flags (Overflow, Compare Match and Capture Event) are found in the Timer/Counter Interrupt Flag Register (TIFR). Control signals are found in the Timer/Counter Control Registers – TCCR0 and TCCR2. The interrupt enable/disable settings are found in the Timer/Counter Interrupt Mask Register (TIMSK).

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 clock period. The external clock signal is sampled on the rising edge of the internal CPU clock.

The 8-bit Timer/Counters feature a high-resolution and a high-accuracy usage with the lower prescaling opportunities. Similarly, the high prescaling opportunities make these units useful for lower speed functions or exact timing functions with infrequent actions.

Both Timer/Counters support two Output Compare functions using the Output Compare Registers (OCR0 and OCR2) as the data source to be compared to the Timer/Counter contents. The Output Compare functions include optional clearing of the counter on compare match and action on the Output Compare pins – PB4(OC0/PWM0) and PB7(OC2/PWM2) – on compare match.

Timer/Counters 0 and 2 can also be used as 8-bit Pulse Width Modulators. In this mode the Timer/Counter and the Output Compare Register serve as a glitch-free, stand-alone PWM with centered pulses. Refer to page 43 for a detailed description of this function.

In PWM mode, this bit has no effect.

- **Bits 2, 1, 0 – CS02, CS01, CS00/CS22, CS21, CS20: Clock Select Bits 2, 1 and 0**

The Clock Select2 bits 2, 1 and 0 define the prescaling source of the Timer/Counter.

Table 11. Timer/Counter0 Prescale Select

CS02	CS01	CS00	Description
0	0	0	Timer/Counter0 is stopped.
0	0	1	PCK0
0	1	0	PCK0/8
0	1	1	PCK0/32
1	0	0	PCK0/64
1	0	1	PCK0/128
1	1	0	PCK0/256
1	1	1	PCK0/1024

Table 12. Timer/Counter2 Prescale Select

CS22	CS21	CS20	Description
0	0	0	Timer/Counter2 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 PD7(T2), falling edge
1	1	1	External Pin PD7(T2), rising edge

The Stop condition provides a Timer Enable/Disable function. The CK down divided modes are scaled directly from the CK CPU clock. If the external pin modes are used for Timer/Counter2, transitions on PD7/(T2) will clock the counter even if the pin is configured as an output. This feature can give the user software control of the counting.

Timer/Counter0 – TCNT0

Bit	7	6	5	4	3	2	1	0	
$32 ($42)	MSB							LSB	TCNT0
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	

Timer/Counter2 – TCNT2

Bit	7	6	5	4	3	2	1	0	
$24 ($44)	MSB							LSB	TCNT2
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	

These 8-bit registers contain the value of the Timer/Counters.

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

Timer/Counter0 Output Compare Register – OCR0

Bit	7	6	5	4	3	2	1	0	
$31 ($51)	MSB							LSB	OCR0
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	

Timer/Counter2 Output Compare Register – OCR2

Bit	7	6	5	4	3	2	1	0	
$23 ($43)	MSB							LSB	OCR2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Output Compare Registers are 8-bit read/write registers.

The Timer/Counter Output Compare Registers contain the data to be continuously compared with the Timer/Counter. Actions on compare matches are specified in TCCR0 and TCCR2. A compare match does only occur if the Timer/Counter counts to the OCR value. A software write that sets the Timer/Counter and Output Compare Register to the same value does not generate a compare match.

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

Timer/Counters 0 and 2 in PWM Mode

When the PWM mode is selected, the Timer/Counter and the Output Compare Register (OCR0 or OCR2) form an 8-bit, free-running, glitch-free and phase correct PWM with outputs on the PB4(OC0/PWM0) or PB7(OC2/PWM2) pin. The Timer/Counter acts as an up/down counter, counting up from $00 to $FF, where it turns and counts down again to zero before the cycle is repeated. When the counter value matches the contents of the Output Compare Register, the PB4(OC0/PWM0) or PB7(OC2/PWM2) pin is set or cleared according to the settings of the COM01/COM00 or COM21/COM20 bits in the Timer/Counter Control Registers TCCR0 and TCCR2. Refer to Table 13 for details.

Table 13. Compare Mode Select in PWM Mode

COMn1	COMn0	Effect on Compare/PWM Pin
0	0	Not connected
0	1	Not connected
1	0	Cleared on compare match, up-counting. Set on compare match, down-counting (non-inverted PWM).
1	1	Cleared on compare match, down-counting. Set on compare match, up-counting (inverted PWM).

Note: n = 0 or 2

Note that in PWM mode, the Output Compare Register is transferred to a temporary location when written. The value is latched when the Timer/Counter reaches $FF. This prevents the occurrence of odd-length PWM pulses (glitches) in the event of an unsynchronized OCR0 or OCR2 write. See Figure 32 for an example.

- When writing to one of the registers TCNT0, OCR0 or TCCR0, 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 TCNT0 does not disturb an OCR0 write in progress. To detect that a transfer to the destination register has taken place, an Asynchronous Status Register (ASSR) has been implemented.
- When entering Power-save mode after having written to TCNT0, OCR0 or TCCR0, the user must wait until the written register has been updated if Timer/Counter0 is used to wake up the device. Otherwise, the MCU will go to sleep before the changes have had any effect. This is extremely important if the Output Compare0 interrupt is used to wake up the device; Output Compare is disabled during write to OCR0 or TCNT0. If the write cycle is not finished (i.e., the user goes to sleep before the OCR0UB bit returns to zero), the device will never get a compare match and the MCU will not wake up.
- If Timer/Counter0 is used to wake up the device from Power-save mode, precautions must be taken if the user wants to reenter Power-save mode: The interrupt logic needs one TOSC1 cycle to get reset. If the time between wake-up and reentering 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 TCCR0, TCNT0 or OCR0.
  2. Wait until the corresponding Update Busy Flag in ASSR returns to zero.
  3. Enter Power-save mode.
- When asynchronous operation is selected, the 32 kHz Oscillator for Timer/Counter0 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/Counter0 after Power-up or wake-up from Power-down. The content of all Timer/Counter0 Registers must be considered lost after a wake-up from Power-down due to the unstable clock signal upon start-up, no matter whether the Oscillator is in use or a clock signal is applied to the TOSC pin.
- 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 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. To execute the corresponding Timer/Counter0 interrupt routine, the Global Interrupt bit in SREG must have been set. Otherwise, the part will still wake up from Power-down, but continues to execute the Sleep command. The Interrupt Flags are updated three processor cycles after the processor clock has started. During these cycles, the processor executes instructions, but the interrupt condition is not readable and the interrupt routine has not started yet.
- 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 of which TCNT0 will read as the same value as before Power-

save mode was entered. After an edge on the asynchronous clock, TCNT0 will read correctly. (The compare and overflow functions of the Timer are not affected by this behavior.) Safe procedure to ensure correct value is read:

1. Write any value to either of the registers OCR0 or TCCR0
2. Wait for the corresponding Update Busy Flag to be cleared
3. Read TCNT0

Note that OCR0 and TCCR0 are never modified by hardware, and will always read correctly.

16-bit Timer/Counter1

Figure 33 shows the block diagram for Timer/Counter1.

The 16-bit Timer/Counter1 can select clock source from CK, prescaled CK or an external pin. In addition, it can be stopped as described in the specification for the Timer/Counter1 Control Register (TCCR1B). The different Status Flags (Overflow, Compare Match and Capture Event) are found in the Timer/Counter Interrupt Flag Register (TIFR). Control signals are found in the Timer/Counter1 Control Registers – TCCR1A and TCCR1B. The interrupt enable/disable settings for Timer/Counter1 are found in the Timer/Counter Interrupt Mask Register (TIMSK).

When Timer/Counter1 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 clock period. The external clock signal is sampled on the rising edge of the internal CPU clock.

The 16-bit Timer/Counter1 features both a high-resolution and a high-accuracy usage with the lower prescaling opportunities. Similarly, the high prescaling opportunities makes the Timer/Counter1 useful for lower speed functions or exact timing functions with infrequent actions.

The Timer/Counter1 supports two Output Compare functions using the Output Compare Registers 1A and 1B (OCR1A and OCR1B) as the data sources to be compared to the Timer/Counter1 contents. The Output Compare functions include optional clearing of the counter on Compare A Match, and actions on the Output Compare pins on both compare matches.

1. In the same operation, write a logical “1” to WDTOE and WDE. A logical “1” must be written to WDE even though it is set to one before the disable operation starts.
2. Within the next four clock cycles, write a logical “0” to WDE. This disables the Watchdog.

- **Bits 2..0 – WDP2, WDP1, WDP0: Watchdog Timer Prescaler 2, 1 and 0**

The WDP2, WDP1 and WDP0 bits determine the Watchdog Timer prescaling when the Watchdog Timer is enabled. The different prescaling values and their corresponding Time-out periods are shown in Table 21.

Table 21. Watchdog Timer Prescale Select

WDP2	WDP1	WDP0	Number of WDT Oscillator Cycles	Typical Time-out at V_{CC} = 3.0V	Typical Time-out at V_{CC} = 5.0V
0	0	0	16K cycles	47 ms	15 ms
0	0	1	32K cycles	94 ms	30 ms
0	1	0	64K cycles	0.19 s	60 ms
0	1	1	128K cycles	0.38 s	0.12 s
1	0	0	256K cycles	0.75 s	0,24 s
1	0	1	512K cycles	1.5 s	0.49 s
1	1	0	1,024K cycles	3.0 s	0.97 s
1	1	1	2,048K cycles	6.0 s	1.9 s

Note: The frequency of the Watchdog Oscillator is voltage-dependent as shown in the Electrical Characteristics section.
The WDR (Watchdog Reset) instruction should always be executed before the Watchdog Timer is enabled. This ensures that the reset period will be in accordance with the Watchdog Timer prescale settings. If the Watchdog Timer is enabled without reset, the Watchdog Timer may not start counting from zero.
To avoid unintentional MCU Reset, the Watchdog Timer should be disabled or reset before changing the Watchdog Timer Prescale Select.

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 Status Register – USR

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

The USR Register is a read-only register providing information on the UART Status.

- **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 the 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.

When the TXCIE bit in UCR is set, setting of TXC causes the UART Transmit Complete interrupt to be executed. TXC is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, the TXC bit is cleared (zero) by writing a logical “1” to the bit.

- **Bit 5 – UDRE: UART Data Register Empty**

This bit is set (one) when a character written to UDR is transferred to the Transmit Shift Register. Setting of this bit indicates that the Transmitter is ready to receive a new character for transmission.

When the UDRIE bit in UCR is set, the UART Transmit Complete interrupt to be executed as long as UDRE is set. UDRE is cleared by writing UDR. When interrupt-driven data transmittal is used, the UART Data Register Empty Interrupt routine must write UDR in order to clear UDRE, otherwise a new interrupt will occur once the interrupt routine terminates.

UDRE is set (one) during reset to indicate that the Transmitter is ready.

- **Bit 1 – RXB8: Receive Data Bit 8**

When CHR9 is set (one), RXB8 is the ninth data bit of the received character.

- **Bit 0 – TXB8: Transmit Data Bit 8**

When CHR9 is set (one), TXB8 is the ninth data bit in the character to be transmitted.

- **Bit 5 – Res: Reserved Bit**

This bit is reserved in the ATmega103(L). **Warning:** When writing ADCSR, a logical “0” must be written to this bit.

- **Bit 4 – ADIF: ADC Interrupt Flag**

This bit is set (one) when an ADC conversion is complete and the result is written to the ADC Data Registers are updated. The ADC Conversion Complete interrupt is executed if the ADIE bit and the I-bit in SREG are set (one). ADIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ADIF is cleared by writing a logical “1” to the flag. Beware that if doing a Read-Modify-Write on ADCSR, a pending interrupt can be disabled. This also applies if the SBI and CBI instructions are used.

- **Bit 3 – ADIE: ADC Interrupt Enable**

When this bit is set (one) and the I-bit in SREG is set (one), the ADC Conversion Complete interrupt is activated.

- **Bits 2..0 – ADPS2..ADPS0: ADC Prescaler Select Bits**

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

Table 27. ADC Prescaler Selections

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	Invalid
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

ADC Data Register – ADCL and ADCH

Bit	15	14	13	12	11	10	9	8	
$05 ($25)	–	–	–	–	–	–	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	

When an ADC conversion is complete, the result is found in these two registers. It is essential that both registers are read and that ADCL is read before ADCH.

Table 30. DDBn Effects on Port B Pins

DDBn	PORTBn	I/O	Pull-up	Comment
0	0	Input	No	Tri-state (high-Z)
0	1	Input	Yes	PBn will source current if ext. pulled low
1	0	Output	No	Push-pull Zero Output
1	1	Output	No	Push-pull One Output

Note: n: 7,6...0, pin number

Alternate Functions of Port B

The alternate pin configuration is as follows:

- **OC2/PWM2, Bit 7**

OC2/PWM2, Output Compare output for Timer/Counter2 or PWM output when Timer/Counter2 is in PWM mode. The pin has to be configured as an output to serve this function.

- **OC1B/PWM1B, Bit 6**

OC1B/PWM1B, Output Compare output B for Timer/Counter1 or PWM output B when Timer/Counter1 is in PWM mode. The pin has to be configured as an output to serve this function.

- **OC1A/PWM1A, Bit 5**

OC1A/PWM1A, Output Compare output A for Timer/Counter1 or PWM output A when Timer/Counter1 is in PWM mode. The pin has to be configured as an output to serve this function.

- **OC0/PWM0, Bit 4**

OC0/PWM0, Output Compare output for Timer/Counter0 or PWM output when Timer/Counter0 is in PWM mode. The pin has to be configured as an output to serve this function.

- **MISO – Port B, Bit 3**

MISO: Master Data input, Slave Data output pin for SPI channel. When the SPI is enabled as a Master, this pin is configured as an input regardless of the setting of DDB3. When the SPI is enabled as a Slave, the data direction of this pin is controlled by DDB3. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB3 bit. See the description of the SPI port for further details.

- **MOSI – Port B, Bit 2**

MOSI: SPI Master Data output, Slave Data input for SPI channel. When the SPI is enabled as a Slave, this pin is configured as an input regardless of the setting of DDB2. When the SPI is enabled as a Master, the data direction of this pin is controlled by DDB2. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB2 bit. See the description of the SPI port for further details.

- **SCK – Port B, Bit 1**

SCK: Master Clock output, Slave Clock input pin for SPI channel. When the SPI is enabled as a Slave, this pin is configured as an input regardless of the setting of DDB1. When the SPI is enabled as a Master, the data direction of this pin is controlled by

DDB1. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB1 bit. See the description of the SPI port for further details.

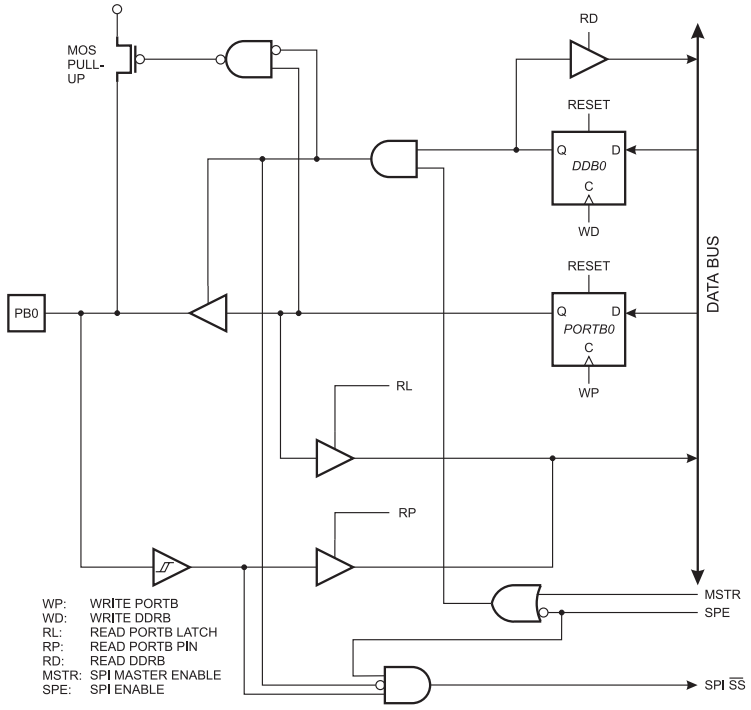
- **$\overline{SS}$ – Port B, Bit 0**

$\overline{SS}$: Slave Port Select input. When the SPI is enabled as a Slave, this pin is configured as an input regardless of the setting of DDB0. As a Slave, the SPI is activated when this pin is driven low. When the SPI is enabled as a Master, the data direction of this pin is controlled by DDB0. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB0 bit. See the description of the SPI port for further details.

Port B Schematics

Note that all port pins are synchronized. The synchronization latches are, however, not shown in the figures.

Figure 54. Port B Schematic Diagram (Pin PB0)



Port E Data Register – PORTE

Bit	7	6	5	4	3	2	1	0	
$03 ($23)	PORTE7	PORTE6	PORTE5	PORTE4	PORTE3	PORTE2	PORTE1	PORTE0	PORTE
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	

Port E Data Direction Register – DDRE

Bit	7	6	5	4	3	2	1	0	
$02 ($22)	DDE7	DDE6	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0	DDRE
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	

Port E Input Pins Address – PINE

Bit	7	6	5	4	3	2	1	0	
$01 ($21)	PINE7	PINE6	PINE5	PINE4	PINE3	PINE2	PINE1	PINE0	PINE
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

The Port E Input Pins address (PINE) is not a register; this address enables access to the physical value on each Port E pin. When reading PORTE, the Port E Data Latch is read and when reading PINE, the logical values present on the pins are read.

Port E as General Digital I/O

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

Table 34. DDEn Bits on Port E Pins

DDEn	PORTEn	I/O	Pull-up	Comment
0	0	Input	No	Tri-state (high-Z)
0	1	Input	Yes	PDn will source current if ext. pulled low.
1	0	Output	No	Push-pull Zero Output
1	1	Output	No	Push-pull One Output

Note: n: 7,6...0, pin number

Alternate Functions of Port E

The alternate pin functions of Port E are:

- **PDI/RXD – Port E, Bit 0**

PDI, Serial Programming Data Input. During Serial Program downloading, this pin is used as data input line for the ATmega103(L).

RXD, UART Receive Pin. Receive Data (Data input pin for the UART). When the UART Receiver is enabled, this pin is configured as an input regardless of the value of DDRD0. When the UART forces this pin to be an input, a logical “1” in PORTD0 will turn on the internal pull-up.

- **PDO/TXD – Port E, Bit 1**

PDO, Serial Programming Data Output. During Serial Program downloading, this pin is used as data output line for the ATmega103(L).

TXD, UART Transmit Pin.

- **AC+ – Port E, Bit 2**

AC+, Analog Comparator Positive Input. This pin is directly connected to the positive input of the analog comparator.

- **AC- – Port E, Bit 3**

AC-, Analog Comparator Negative Input. This pin is directly connected to the negative input of the analog comparator.

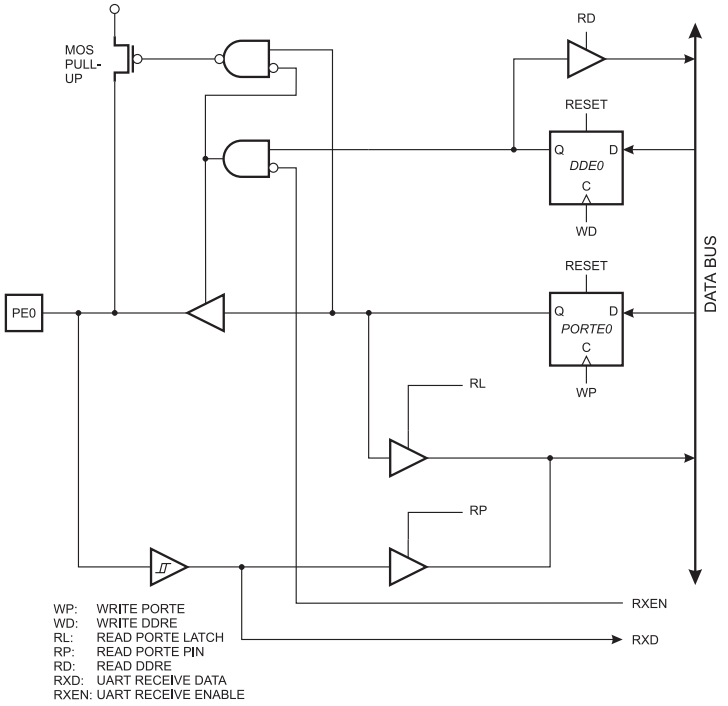
- **INT4..INT7 – Port E, Bits 4 - 7**

INT4..INT7, External Interrupt sources 4 - 7: The PE4 - PE7 pins can serve as external interrupt sources to the MCU. Interrupts can be triggered by low level or positive or negative edge on these pins. The internal pull-up MOS resistors can be activated as described above. See the interrupt description for further details, and how to enable the sources.

Port E Schematics

Note that all port pins are synchronized. The synchronization latches are, however, not shown in the figures.

Figure 66. Port E Schematic Diagram (Pin PE0)



Memory Programming

Program and Data Memory Lock Bits

The ATmega103(L) MCU provides two Lock bits that can be left unprogrammed (“1”) or can be programmed (“0”) to obtain the additional features listed in Table 35. The Lock bits can only be erased to “1” with the Chip Erase command.

Table 35. Lock Bit Protection Modes

Memory Lock Bits			Protection Type
Mode	LB1	LB2	
1	1	1	No memory lock features enabled.
2	0	1	Further programming of the Flash and EEPROM is disabled.(1)
3	0	0	Same as mode 2, and verify is also disabled.

Note: 1. In Parallel mode, programming of the Fuse bits are also disabled. Program the Fuse bits before programming the Lock bits.

Fuse Bits

The ATmega103(L) has four Fuse bits, SPIEN, SUT1..0 and EESAVE.

- When the SPIEN Fuse is programmed (“0”), Serial Program and Data Downloading is enabled. Default value is programmed (“0”). The SPIEN Fuse is not accessible in Serial Programming mode.
- When EESAVE is programmed, the EEPROM memory is preserved through the Chip Erase cycle. Default value is unprogrammed (“1”). The EESAVE Fuse bit cannot be programmed if any of the Lock bits are programmed.
- SUT1..0 Fuses: Determine the MCU start-up time. See Table 5 on page 27 for further details. Default value is unprogrammed (“11”), which gives a nominal start-up time of 16 ms.

The status of the Fuse bits is not affected by Chip Erase.

Signature Bytes

All Atmel microcontrollers have a 3-byte signature code that identifies the device. This code can be read in both Serial and Parallel mode. The three bytes reside in a separate address space.

For the ATmega103 they are:

1. $000: $1E (indicates manufactured by Atmel)
2. $001: $97 (indicates 128K bytes Flash memory)
3. $002: $01 (indicates ATmega103 when signature byte $001 is $97)

Programming the Flash and EEPROM

Atmel’s ATmega103(L) offers 128K bytes of In-System Reprogrammable Flash memory and 4K bytes of EEPROM Data memory.

The ATmega103(L) is shipped with the On-chip Flash Program and EEPROM Data memory arrays in the erased state (i.e., contents = $FF) and ready to be programmed. This device supports a Parallel Programming mode and a Serial Programming mode. The +12V supplied to the $\overline{\text{RESET}}$ pin in Parallel Programming mode is used for programming enable only, and no current of significance is drawn by this pin. The Serial Programming mode provides a convenient way to download program and data into the ATmega103(L) inside the user’s system.

Table 44. Serial Programming Instruction Set

Instruction	Instruction Format				Operation
	Byte 1	Byte 2	Byte 3	Byte 4	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx	Enable serial programming while $\overline{\text{RESET}}$ is low.
Chip Erase	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Chip erase EEPROM and Flash.
Read Program Memory	0010 **H**000	**aaaa 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	**xbbb bbbb**	**iiii iiii**	Write **H** (high or low) data **i** to Program memory page at word address **b**.
Write Program Memory Page	0100 1100	**aaaa aaaa**	**b**xxx xxxx	xxxx xxxx	Write Program memory page at address **a**:**b**.
Read EEPROM Memory	1010 0000	xxxx **aaaa**	**bbbb bbbb**	**oooo oooo**	Read data **o** from EEPROM memory at address **a**:**b**.
Write EEPROM Memory	1100 0000	xxxx **aaaa**	**bbbb bbbb**	**iiii iiii**	Write data **i** to EEPROM memory at address **a**:**b**.
Read Lock Bits	0101 1000	xxxx xxxx	xxxx xxxx	xxxx x***21***x	Read Lock bits. “0” = programmed, “1” = unprogrammed.
Write Lock Bits	1010 1100	1111 1***21***1	xxxx xxxx	xxxx xxxx	Write Lock bits. Set bits ***1,2*** = “0” to Program Lock bits.
Read Fuse Bits	0101 0000	xxxx xxxx	xxxx xxxx	xx**5**x **6**1**43**	Read Fuse bits. “0” = programmed, “1” = unprogrammed.
Write Fuse Bits	1010 1100	1011 **6**1**43**	xxxx xxxx	xxxx xxxx	Write Fuse bits. Set bit **6**,**4,3** = “0” to program, “1” to unprogram.
Read Signature Byte	0011 0000	xxxx xxxx	xxxx xx**bb**	**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 = SUT0 Fuse
4 = SUT1 Fuse
5 = SPIEN Fuse
6 = EESAVE Fuse

Typical Characteristics

The following charts show typical behavior. These figures are not tested during manufacturing. All current consumption measurements are performed with all I/O pins configured as inputs and with internal pull-ups enabled. All pins on Port F are pulled high externally. A sine wave generator with rail-to-rail output is used as clock source.

The power consumption in Power-down mode is independent of clock selection.

The current consumption is a function of several factors such as: operating voltage, operating frequency, loading of I/O pins, switching rate of I/O pins, code executed and ambient temperature. The dominating factors are operating voltage and frequency.

The current drawn from capacitive loaded pins may be estimated (for one pin) as $C_L \bullet V_{CC} \bullet f$, where C_L = load capacitance, V_{CC} = operating voltage and f = average switching frequency of I/O pin.

The parts are characterized at frequencies higher than test limits. Parts are not guaranteed to function properly at frequencies higher than the ordering code indicates.

The difference between current consumption in Power-down mode with Watchdog Timer enabled and Power-down mode with Watchdog Timer disabled represents the differential current drawn by the Watchdog Timer.

Figure 81. Active Supply Current vs. Frequency