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	Active
Core Processor	AVR
Core Size	8-Bit
Speed	16MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	23
Program Memory Size	8KB (4K 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 (TA)
Mounting Type	Surface Mount
Package / Case	32-TQFP
Supplier Device Package	32-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega8a-au

2. Configuration Summary

Features	ATmega8A
Pin count	32
Flash (KB)	8
SRAM (KB)	1
EEPROM (Bytes)	512
General Purpose I/O pins	23
SPI	1
TWI (I^2C)	1
USART	1
ADC	10-bit 15ksps
ADC channels	6 (8 in TQFP and QFN/MLF packages)
AC propagation delay	Typ 400ns
8-bit Timer/Counters	2
16-bit Timer/Counters	1
PWM channels	3
RC Oscillator	+/-3%
Operating voltage	2.7 - 5.5V
Max operating frequency	16MHz
Temperature range	-40°C to +105°C

15. System Control and Reset

15.1. Resetting the AVR

During Reset, all I/O Registers are set to their initial values, and the program starts execution from the Reset Vector. If the program never enables an interrupt source, the Interrupt Vectors are not used, and regular program code can be placed at these locations. This is also the case if the Reset Vector is in the Application section while the Interrupt Vectors are in the boot section or vice versa. The circuit diagram in the following section shows the Reset Logic. The Table in *System and Reset Characteristics* defines the electrical parameters of the reset circuitry.

The I/O ports of the AVR are immediately reset to their initial state when a reset source goes active. This does not require any clock source to be running.

After all reset sources have gone inactive, a delay counter is invoked, stretching the internal reset. This allows the power to reach a stable level before normal operation starts. The time-out period of the delay counter is defined by the user through the CKSEL Fuses. The different selections for the delay period are presented in *Clock Sources*.

Related Links

System and Reset Characteristics on page 305

Clock Sources on page 45

15.2. Reset Sources

The ATmega8A has four sources of Reset:

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

When the BOOTRST Fuse is programmed, the boot section size set to 2K bytes, and the IVSEL bit in the GICR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses is:

```
;
.org $c00
$c00                rjmp        RESET               ; Reset handler
$c01                rjmp        EXT_INT0            ; IRQ0 Handler
$c02                rjmp        EXT_INT1            ; IRQ1 Handler
:.                  :.          :.
$c12                rjmp        SPM_RDY             ; Store Program
                                                    Memory Ready
                                                    Handler
$c13          RESET:    ldi       r16,high(RAMEND)    ; Main program
                                                    start
$c14                out         SPH,r16             ; Set Stack
                                                    Pointer to top of
                                                    RAM
$c15                ldi         r16,low(RAMEND)
$c16                out         SPL,r16
$c17                sei                             ; Enable
                                                    interrupts
$c18                <instr>     XXX
```

Related Links

Boot Loader Support – Read-While-Write Self-Programming on page 266
ATmega8A Boot Loader Parameters on page 278

16.1.1. Moving Interrupts Between Application and Boot Space

The General Interrupt Control Register controls the placement of the Interrupt Vector table.

16.2. Register Description

17.1.2. GICR – General Interrupt Control Register

When using the I/O specific commands IN and OUT, the I/O addresses 0x00 - 0x3F must be used. When addressing I/O Registers as data space using LD and ST instructions, 0x20 must be added to these offset addresses. The device is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

Name: GICR
Offset: 0x3B
Reset: 0
Property: When addressing I/O Registers as data space the offset address is 0x5B

Bit	7	6	5	4	3	2	1	0
	INT1	INT0						
Access	R/W	R/W						
Reset	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 enabled. 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 the INT1 Interrupt Vector.

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 enabled. 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 and/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 the INT0 Interrupt Vector.

17.1.3. GIFR – General Interrupt Flag Register

When using the I/O specific commands IN and OUT, the I/O addresses 0x00 - 0x3F must be used. When addressing I/O Registers as data space using LD and ST instructions, 0x20 must be added to these offset addresses. The device is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

Name: GIFR
Offset: 0x3A
Reset: 0
Property: When addressing I/O Registers as data space the offset address is 0x5A

Bit	7	6	5	4	3	2	1	0
	INTF1	INTF0						
Access	R/W	R/W						
Reset	0	0						

Bit 7 – INTF1: External Interrupt Flag 1

When an event on the INT1 pin triggers an interrupt request, INTF1 becomes set (one). If the I-bit in SREG and the INT1 bit in GICR are set (one), the MCU will jump to the corresponding 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 Flag 0

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

$\overline{SS}$: Slave Select input. When the SPI is enabled as a Slave, this pin is configured as an input regardless of the setting of DDB2. 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 DDB2. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB2 bit.

OC1B, Output Compare Match output: The PB2 pin can serve as an external output for the Timer/Counter1 Compare Match B. The PB2 pin has to be configured as an output (DDB2 set (one)) to serve this function. The OC1B pin is also the output pin for the PWM mode timer function.

• OC1A – Port B, Bit 1

OC1A, Output Compare Match output: The PB1 pin can serve as an external output for the Timer/Counter1 Compare Match A. The PB1 pin has to be configured as an output (DDB1 set (one)) to serve this function. The OC1A pin is also the output pin for the PWM mode timer function.

• ICP1 – Port B, Bit 0

ICP1 – Input Capture Pin: The PB0 pin can act as an Input Capture Pin for Timer/Counter1.

The tables below relate the alternate functions of Port B to the overriding signals shown in figure Figure 18-5 Alternate Port Functions(1) on page 82. SPI MSTR INPUT and SPI SLAVE OUTPUT constitute the MISO signal, while MOSI is divided into SPI MSTR OUTPUT and SPI SLAVE INPUT.

Table 18-4 Overriding Signals for Alternate Functions in PB7:PB4

Signal Name	PB7/XTAL2/ TOSC2[1][2]	PB6/XTAL1/ TOSC1[1]	PB5/SCK	PB4/MISO
PUOE	$\overline{EXT}$ • ($\overline{INTRC}$ + AS2)	$\overline{INTRC}$ + AS2	SPE • $\overline{MSTR}$	SPE • MSTR
PUO	0	0	PORTB5 • $\overline{PUD}$	PORTB4 • $\overline{PUD}$
DDOE	$\overline{EXT}$ • ($\overline{INTRC}$ + AS2)	$\overline{INTRC}$ + AS2	SPE • $\overline{MSTR}$	SPE • MSTR
DDOV	0	0	0	0
PVOE	0	0	SPE • MSTR	SPE • $\overline{MSTR}$
PVOV	0	0	SCK OUTPUT	SPI SLAVE OUTPUT
DIEOE	$\overline{EXT}$ • ($\overline{INTRC}$ + AS2)	$\overline{INTRC}$ + AS2	0	0
DIEOV	0	0	0	0
DI	–	–	SCK INPUT	SPI MSTR INPUT
AIO	Oscillator Output	Oscillator/Clock Input	–	–

Note:

1. INTRC means that the internal RC Oscillator is selected (by the CKSEL Fuse).
2. EXT means that the external RC Oscillator or an external clock is selected (by the CKSEL Fuse).

Table 18-5 Overriding Signals for Alternate Functions in PB3:PB0

Signal Name	PB3/MOSI/ OC2	PB2/$\overline{SS}$/ OC1B	PB1/OC1A	PB0/ICP1
PUOE	SPE • $\overline{MSTR}$	SPE • $\overline{MSTR}$	0	0
PUO	PORTB3 • $\overline{PUD}$	PORTB2 • $\overline{PUD}$	0	0
DDOE	SPE • $\overline{MSTR}$	SPE • $\overline{MSTR}$	0	0

Signal Name	PB3/MOSI/ OC2	PB2/$\overline{SS}$/ OC1B	PB1/OC1A	PB0/ICP1
DDOV	0	0	0	0
PVOE	SPE • MSTR + OC2 ENABLE	OC1B ENABLE	OC1A ENABLE	0
PVOV	SPI MSTR OUTPUT + OC2	OC1B	OC1A	0
DIEOE	0	0	0	0
DIEOV	0	0	0	0
DI	SPI SLAVE INPUT	SPI $\overline{SS}$	–	ICP1 INPUT
AIO	–	–	–	–

18.3.2. Alternate Functions of Port C

The Port C pins with alternate functions are shown in the table below:

Table 18-6 Port C Pins Alternate Functions

Port Pin	Alternate Function
PC6	$\overline{RESET}$ (Reset pin)
PC5	ADC5 (ADC Input Channel 5) SCL (Two-wire Serial Bus Clock Line)
PC4	ADC4 (ADC Input Channel 4) SDA (Two-wire Serial Bus Data Input/Output Line)
PC3	ADC3 (ADC Input Channel 3)
PC2	ADC2 (ADC Input Channel 2)
PC1	ADC1 (ADC Input Channel 1)
PC0	ADC0 (ADC Input Channel 0)

The alternate pin configuration is as follows:

• $\overline{RESET}$ – Port C, Bit 6

$\overline{RESET}$, Reset pin: When the RSTDISBL Fuse is programmed, this pin functions as a normal I/O pin, and the part will have to rely on Power-on Reset and Brown-out Reset as its reset sources. When the RSTDISBL Fuse is unprogrammed, the reset circuitry is connected to the pin, and the pin can not be used as an I/O pin.

If PC6 is used as a reset pin, DDC6, PORTC6 and PINC6 will all read 0.

• SCL/ADC5 – Port C, Bit 5

SCL, Two-wire Serial Interface Clock: When the TWEN bit in TWCR is set (one) to enable the Two-wire Serial Interface, pin PC5 is disconnected from the port and becomes the Serial Clock I/O pin for the Two-wire Serial Interface. In this mode, there is a spike filter on the pin to suppress spikes shorter than 50 ns on the input signal, and the pin is driven by an open drain driver with slew-rate limitation.

PC5 can also be used as ADC input Channel 5. Note that ADC input channel 5 uses digital power.

• SDA/ADC4 – Port C, Bit 4

SDA, Two-wire Serial Interface Data: When the TWEN bit in TWCR is set (one) to enable the Two-wire Serial Interface, pin PC4 is disconnected from the port and becomes the Serial Data I/O pin for the Two-wire Serial Interface. In this mode, there is a spike filter on the pin to suppress spikes shorter than 50 ns on the input signal, and the pin is driven by an open drain driver with slew-rate limitation.

PC4 can also be used as ADC input Channel 4. Note that ADC input channel 4 uses digital power.

• ADC3 – Port C, Bit 3

PC3 can also be used as ADC input Channel 3. Note that ADC input channel 3 uses analog power.

• ADC2 – Port C, Bit 2

PC2 can also be used as ADC input Channel 2. Note that ADC input channel 2 uses analog power.

• ADC1 – Port C, Bit 1

PC1 can also be used as ADC input Channel 1. Note that ADC input channel 1 uses analog power.

• ADC0 – Port C, Bit 0

PC0 can also be used as ADC input Channel 0. Note that ADC input channel 0 uses analog power.

The tables below relate the alternate functions of Port C to the overriding signals shown in figure Figure 18-5 Alternate Port Functions(1) on page 82.

Table 18-7 Overriding Signals for Alternate Functions in PC6:PC4

Signal Name	PC6/$\overline{\text{RESET}}$	PC5/SCL/ADC5	PC4/SDA/ADC4
PUOE	$\overline{\text{RSTDISBL}}$	TWEN	TWEN
PUOV	1	PORTC5 • $\overline{\text{PUD}}$	PORTC4 • $\overline{\text{PUD}}$
DDOE	$\overline{\text{RSTDISBL}}$	TWEN	TWEN
DDOV	0	SCL_OUT	SDA_OUT
PVOE	0	TWEN	TWEN
PVOV	0	0	0
DIEOE	$\overline{\text{RSTDISBL}}$	0	0
DIEOV	0	0	0
DI	–	–	–
AIO	RESET INPUT	ADC5 INPUT / SCL INPUT	ADC4 INPUT / SDA INPUT

Table 18-8 Overriding Signals for Alternate Functions in PC3:PC0(1)

Signal Name	PC3/A11	PC2/A10	PC1/A9	PC0/A8
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	0	0	0	0

21.11.4. TCNT1H – Timer/Counter1 High byte

Name: TCNT1H
Offset: 0x2D
Reset: 0x00
Property: When addressing I/O Registers as data space the offset address is 0x4D

Bit	7	6	5	4	3	2	1	0
	TCNT1H[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – TCNT1H[7:0]: Timer/Counter 1 High byte
Refer to TCNT1L.

24.7. Data Reception – The USART Receiver

The USART Receiver is enabled by writing the Receive Enable (RXEN) bit in the UCSRB Register to one. When the Receiver is enabled, the normal pin operation of the RxD pin is overridden by the USART and given the function as the Receiver's serial input. The baud rate, mode of operation and frame format must be set up once before any serial reception can be done. If synchronous operation is used, the clock on the XCK pin will be used as transfer clock.

24.7.1. Receiving Frames with 5 to 8 Data Bits

The Receiver starts data reception when it detects a valid start bit. Each bit that follows the start bit will be sampled at the baud rate or XCK clock, and shifted into the Receive Shift Register until the first stop bit of a frame is received. A second stop bit will be ignored by the Receiver. When the first stop bit is received (i.e., a complete serial frame is present in the Receive Shift Register), the contents of the Shift Register will be moved into the receive buffer. The receive buffer can then be read by reading the UDR I/O location.

The following code example shows a simple USART receive function based on polling of the Receive Complete (RXC) Flag. When using frames with less than eight bits the most significant bits of the data read from the UDR will be masked to zero. The USART has to be initialized before the function can be used.

Assembly Code Example[1]

```
USART_Receive:
   ; Wait for data to be received
   sbis  UCSRA, RXC
   rjmp  USART_Receive
   ; Get and return received data from buffer
   in    r16, UDR
   ret
```

C Code Example[1]

```
unsigned char USART_Receive( void )
{
   /* Wait for data to be received */
   while ( !(UCSRA & (1<<RXC)) )
      ;
   /* Get and return received data from buffer */
   return UDR;
}
```

Note: 1. See *About Code Examples*.

The function simply waits for data to be present in the receive buffer by checking the RXC Flag, before reading the buffer and returning the value.

Related Links

About Code Examples on page 23

24.7.2. Receiving Frames with 9 Data Bits

If 9-bit characters are used (UCSZ=7) the ninth bit must be read from the RXB8 bit in UCSRB before reading the low bits from the UDR. This rule applies to the FE, DOR and PE Status Flags as well. Read status from UCSRA, then data from UDR. Reading the UDR I/O location will change the state of the

25. TWI - Two-wire Serial Interface

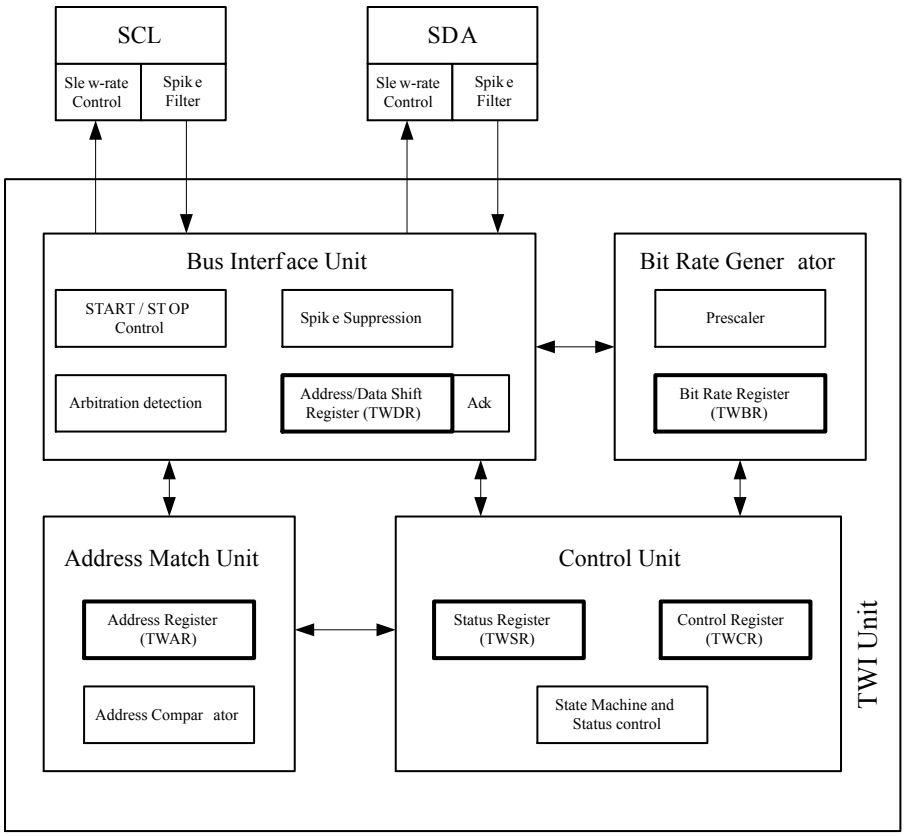
25.1. Features

- Simple, yet Powerful and Flexible Communication Interface, only two Bus Lines Needed
- Both Master and Slave Operation Supported
- Device can Operate as Transmitter or Receiver
- 7-bit Address Space Allows up to 128 Different Slave Addresses
- Multi-master Arbitration Support
- Up to 400kHz Data Transfer Speed
- Slew-rate Limited Output Drivers
- Noise Suppression Circuitry Rejects Spikes on Bus Lines
- Fully Programmable Slave Address with General Call Support
- Address Recognition Causes Wake-up When AVR is in Sleep Mode

25.2. Overview

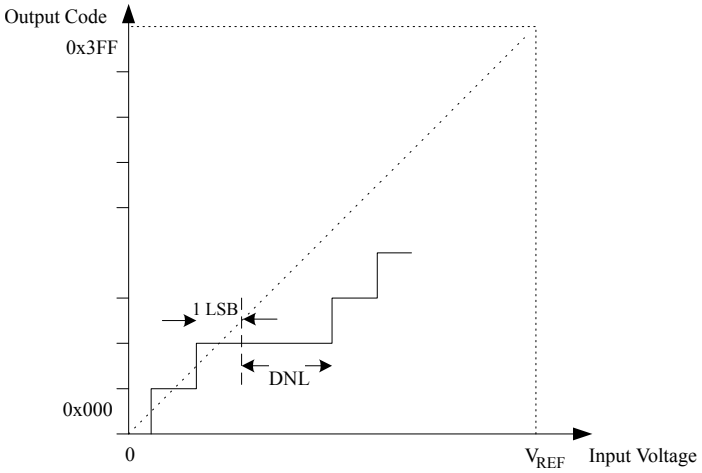
The TWI module is comprised of several submodules, as shown in the following figure. All registers drawn in a thick line are accessible through the AVR data bus.

Figure 25-1 Overview of the TWI Module



- Differential Non-linearity (DNL): The maximum deviation of the actual code width (the interval between two adjacent transitions) from the ideal code width (1 LSB). Ideal value: 0 LSB.

Figure 27-11 Differential Non-linearity (DNL)



- Quantization Error: Due to the quantization of the input voltage into a finite number of codes, a range of input voltages (1 LSB wide) will code to the same value. Always ±0.5 LSB.
- Absolute accuracy: The maximum deviation of an actual (unadjusted) transition compared to an ideal transition for any code. This is the compound effect of offset, gain error, differential error, non-linearity, and quantization error. Ideal value: ±0.5 LSB.

27.7. ADC Conversion Result

After the conversion is complete (ADCSRA.ADIF is set), the conversion result can be found in the ADC Result Registers (ADCL, ADCH).

For single ended conversion, the result is

$$\mathrm{ADC} = \frac{V_{\mathrm{IN}} \cdot 1024}{V_{\mathrm{REF}}}$$

where V_{IN} is the voltage on the selected input pin, and V_{REF} the selected voltage reference (see also descriptions of ADMUX.REFSn and ADMUX.MUX). 0x000 represents analog ground, and 0x3FF represents the selected reference voltage minus one LSB.

27.8. Register Description

4. Maximum conversion time is 1/50kHz*25 = 0.5ms.

31. Electrical Characteristics – TA = -40°C to 105°C

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C	*NOTICE: Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.
Storage Temperature	-65°C to +150°C	
Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground	-0.5V to V_{CC} +0.5V	
Voltage on $\overline{\text{RESET}}$ with respect to Ground	-0.5V to +13.0V	
Maximum Operating Voltage	6.0V	
DC Current per I/O Pin	40.0mA	
DC Current V_{CC} and GND Pins	200.0mA	

31.1. DC Characteristics

Table 31-1 T_A = -40°C to 105°C, V_{CC} = 2.7V to 5.5V (unless otherwise noted)

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
V_{IL}	Input Low Voltage, Except XTAL1 and $\overline{\text{RESET}}$ pin	V_{CC} = 2.7V - 5.5V	-0.5		$0.2V_{CC}$[1]	V
V_{IL1}	Input Low Voltage, XTAL1 pin	V_{CC} = 2.7V - 5.5V	-0.5		$0.1V_{CC}$[1]	V
V_{IL2}	Input Low Voltage, $\overline{\text{RESET}}$ pin	V_{CC} = 2.7V - 5.5V	-0.5		$0.1V_{CC}$[1]	V
V_{IH}	Input High Voltage, Except XTAL1 and $\overline{\text{RESET}}$ pins	V_{CC} = 2.7V - 5.5V	$0.6V_{CC}$[2]		V_{CC} + 0.5	V
V_{IH1}	Input High Voltage, XTAL1 pin	V_{CC} = 2.7V - 5.5V	$0.8V_{CC}$[2]		V_{CC} + 0.5	V
V_{IH2}	Input High Voltage, $\overline{\text{RESET}}$ pin	V_{CC} = 2.7V - 5.5V	$0.9V_{CC}$[2]		V_{CC} + 0.5	V
V_{OL}	Output Low Voltage[3], Port B (except $\overline{\text{RESET}}$)	I_{OL} =20mA, V_{CC} = 5V I_{OL} =10mA, V_{CC} = 3V			0.8 0.6	V
V_{OH}	Output High Voltage[4], Port B (except $\overline{\text{RESET}}$)	I_{OH} = -20mA, V_{CC} = 5V I_{OH} = -10mA, V_{CC} = 3V	4.0 2.2			V
I_{IL}	Input Leakage Current I/O Pin				3	µA

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
I_{IH}	Input Leakage Current I/O Pin				3	µA
R_{RST}	Reset Pull-up Resistor		30		80	kΩ
R_{PU}	I/O Pin Pull-up Resistor		20		50	kΩ
V_{ACIO}	Analog Comparator Input Offset Voltage	V_{CC} = 5V Vin = V_{CC}/2			20	mV
I_{ACLK}	Analog Comparator Input Leakage Current	V_{CC} = 5V V_{in} = V_{CC}/2	-50		50	nA

Note:

1. “Max” means the highest value where the pin is guaranteed to be read as low
2. “Min” means the lowest value where the pin is guaranteed to be read as high
3. Although each I/O port can sink more than the test conditions (20mA at Vcc = 5V, 10mA at Vcc = 3V) under steady state conditions (non-transient), the following must be observed:
   PDIP, TQFP, and QFN/MLF Package:
   3.1. The sum of all IOL, for all ports, should not exceed 300mA.
   3.2. The sum of all IOL, for ports C0 - C5 should not exceed 100mA.
   3.3. The sum of all IOL, for ports B0 - B7, C6, D0 - D7 and XTAL2, should not exceed 200mA.

   If IOL exceeds the test condition, VOL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test condition.
4. Although each I/O port can source more than the test conditions (20mA at Vcc = 5V, 10mA at Vcc = 3V) under steady state conditions (non-transient), the following must be observed:
   PDIP, TQFP, and QFN/MLF Package:
   4.1. The sum of all IOH, for all ports, should not exceed 300mA.
   4.2. The sum of all IOH, for port C0 - C5, should not exceed 100mA
   4.3. The sum of all IOH, for ports B0 - B7, C6, D0 - D7 and XTAL2, should not exceed 200mA.

   If IOH exceeds the test condition, VOH may exceed the related specification. Pins are not guaranteed to source current greater than the listed test condition.

Table 31-2 ATmega8A DC Characteristics

T_A = -40°C to 105°C, V_{CC} = 1.8V to 5.5V (unless otherwise noted)

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
I_{CC}	Power Supply Current	Active 4MHz, V_{CC} = 3V			6	mA
		Active 8MHz, V_{CC} = 5V			15	mA
		Idle 4MHz, V_{CC} = 3V			3	mA
		Idle 8MHz, V_{CC} = 5V			8	mA
	Power-down mode(1)	WDT enabled, V_{CC} = 3V			35	µA
		WDT disabled, V_{CC} = 3V			6	µA

Note: 1. The current consumption values include input leakage current.

Figure 32-71 Reset Supply Current vs. V_{CC} (1 - 16MHz, Excluding Current Through The Reset Pull-up)

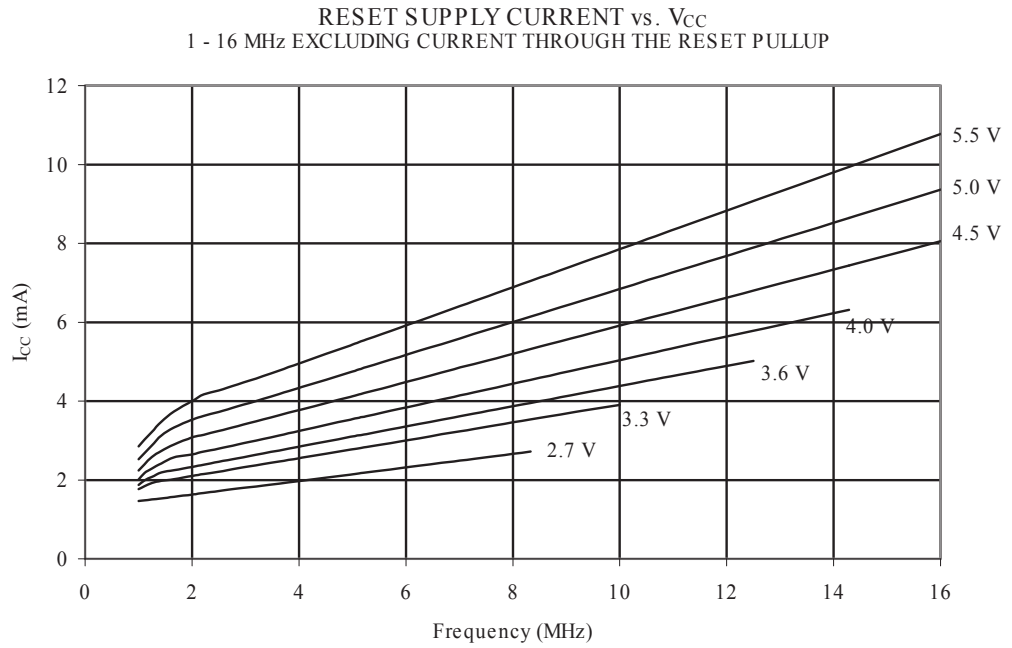


Figure 32-72 Reset Pulse Width vs. V_{CC}

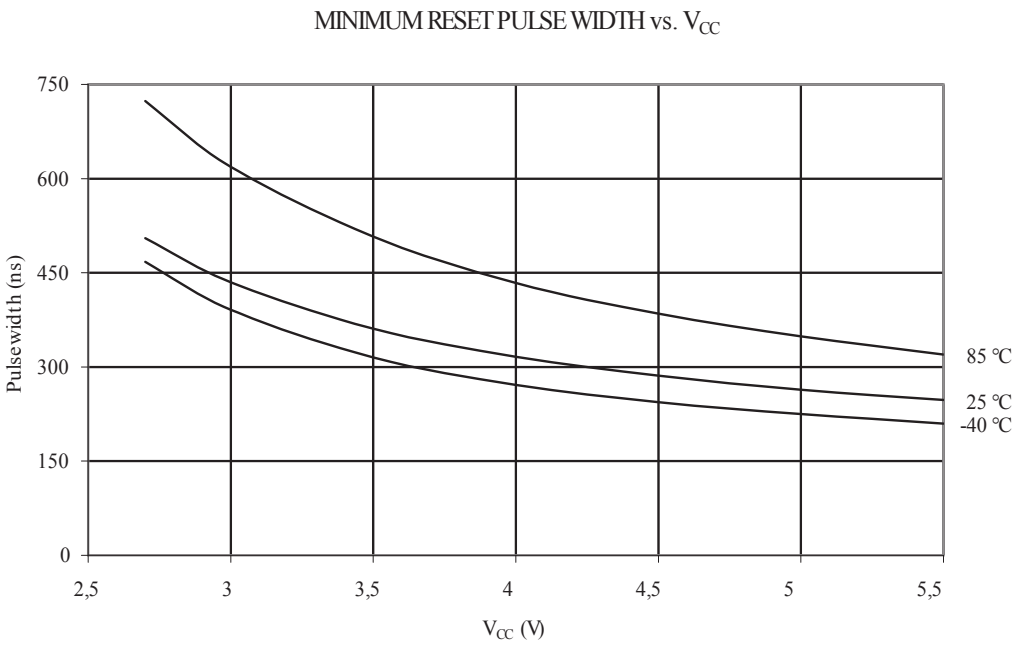
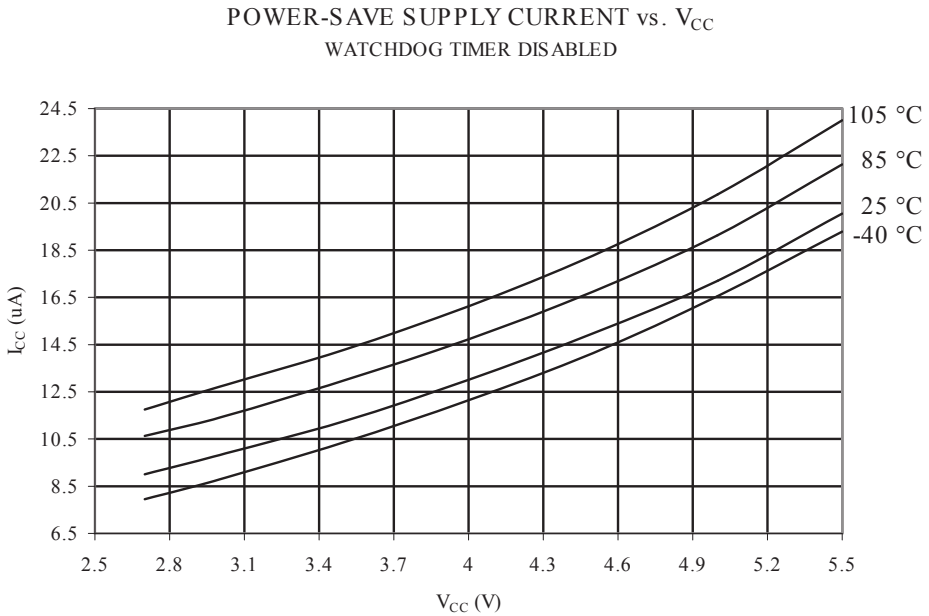
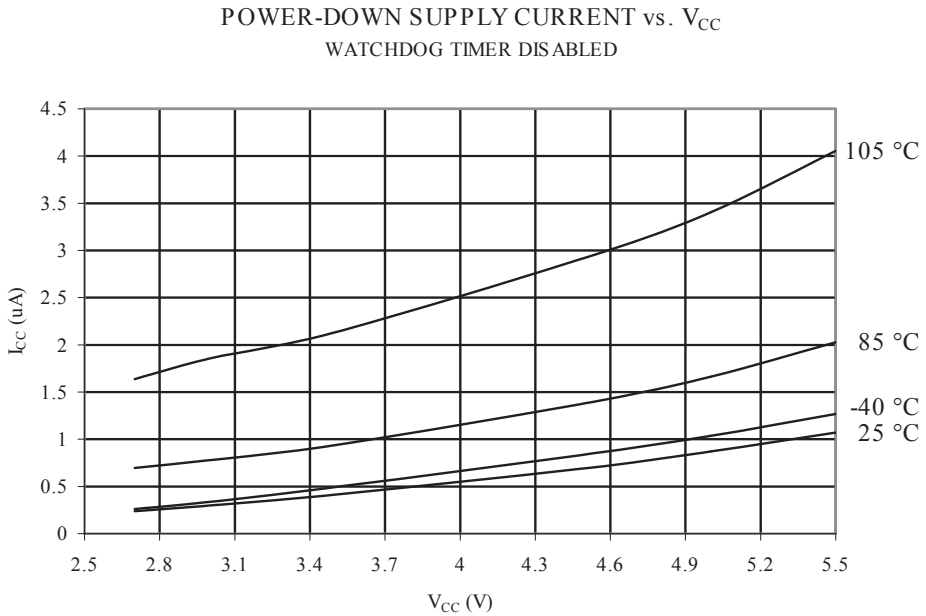


Figure 33-10 Idle Supply Current vs. V_{CC} (32kHz External RC Oscillator)



33.1.3. Power-down Supply Current

Figure 33-11 Power-down Supply Current vs. V_{CC} (Watchdog Timer Disabled)



33.1.10. Internal Oscillator Speed

Figure 33-35 Watchdog Oscillator Frequency vs. V_{CC}

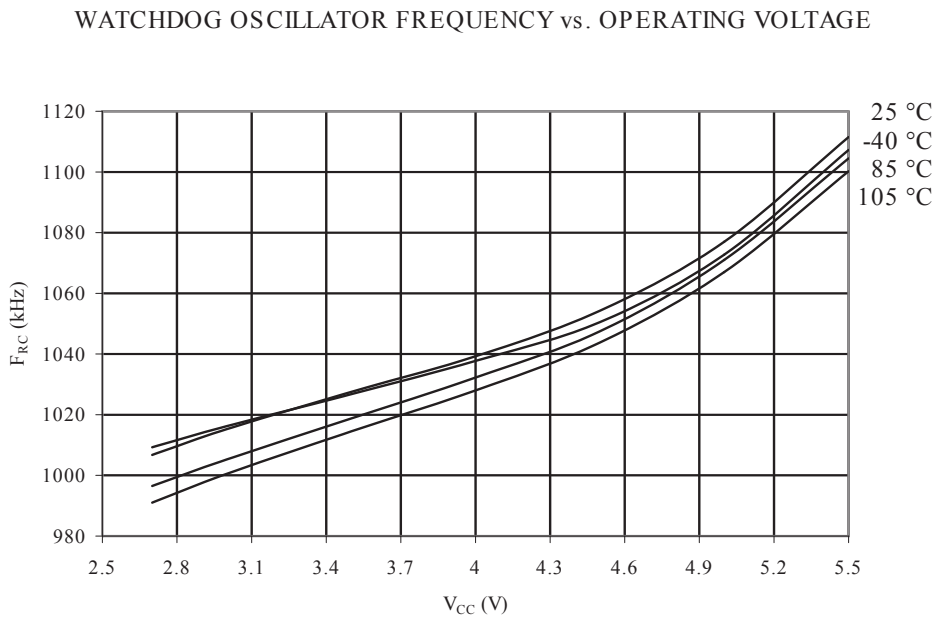
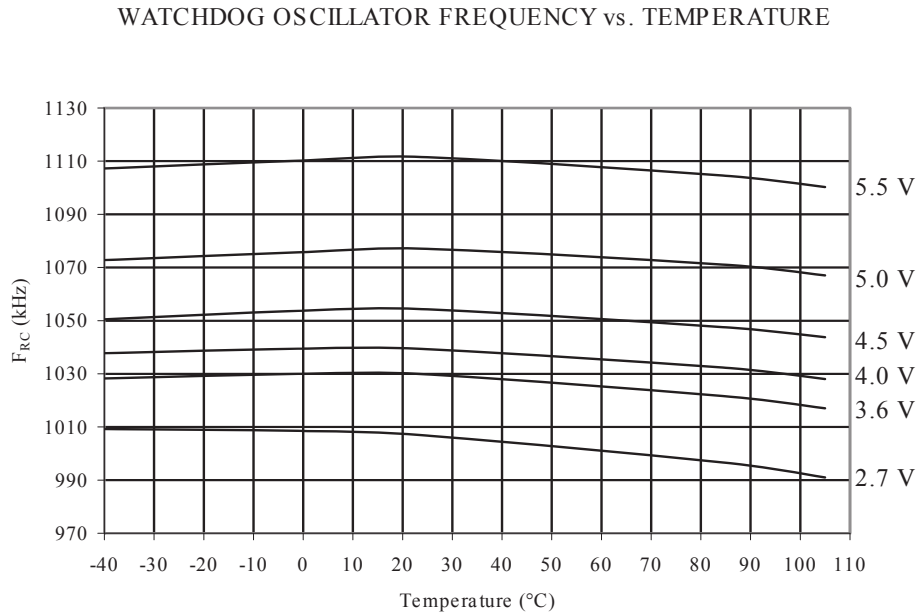


Figure 33-36 Watchdog Oscillator Frequency vs. Temperature



Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x14 (0x34)	DDRC	–	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0
0x13 (0x33)	PINC	–	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0
0x12 (0x32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0
0x11 (0x31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0
0x10 (0x30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0
0x0F (0x2F)	SPDR	SPI Data Register							
0x0E (0x2E)	SPSR	SPIF	WCOL	–	–	–	–	–	SPI2X
0x0D (0x2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
0x0C (0x2C)	UDR	USART I/O Data Register							
0x0B (0x2B)	UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM
0x0A (0x2A)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8
0x09 (0x29)	UBRRL	USART Baud Rate Register Low byte							
0x08 (0x28)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0
0x07 (0x27)	ADMUX	REFS1	REFS0	ADLAR	–	MUX3	MUX2	MUX1	MUX0
0x06 (0x26)	ADCSRA	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0
0x05 (0x25)	ADCH	ADC Data Register High byte							
0x04 (0x24)	ADCL	ADC Data Register Low byte							
0x03 (0x23)	TWDR	Two-wire Serial Interface Data Register							
0x02 (0x22)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE
0x01 (0x21)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0
0x00 (0x20)	TWBR	Two-wire Serial Interface Bit Rate Register							

Note:

1. Refer to the USART description for details on how to access UBRRH and UCSRC.
2. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
3. Some of the Status Flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O Register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers 0x00 to 0x1F only.

DATA TRANSFER INSTRUCTIONS					
Mnemonics	**Operands**	**Description**	**Operation**	**Flags**	**#Clocks**
LD	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, - Y	Load Indirect and Pre-Dec.	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd,Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z), Z ← Z+1	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
LDS	Rd, k	Load Direct from SRAM	Rd ← (k)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr, X ← X + 1	None	2
ST	#NAME?	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ¬ Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr, Y ← Y + 1	None	2
ST	#NAME?	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q,Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr, Z ← Z + 1	None	2
ST	#NAME?	Store Indirect and Pre-Dec.	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q,Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
STS	k, Rr	Store Direct to SRAM	(k) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
LPM	Rd, Z	Load Program Memory	Rd ← (Z)	None	3
LPM	Rd, Z+	Load Program Memory and Post-Inc	Rd ← (Z), Z ← Z+1	None	3
SPM		Store Program Memory	(Z) ← R1:R0	None	-
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2