

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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

E·XFl

Product Status	Obsolete
Core Processor	AVR
Core Size	8-Bit
Speed	10MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	32
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Dual Rows, Exposed Pad
Supplier Device Package	44-QFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega324pv-10mcu

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

6. I/O Multiplexing

Each pin is by default controlled by the PORT as a general purpose I/O and alternatively it can be assigned to one of the peripheral functions.

The following table describes the peripheral signals multiplexed to the PORT I/O pins.

32-pin TQFP/ QFN/ MLF Pin #	40-pin PDIP Pin #	PAD	EXTINT	PCINT	ADC/AC	osc	T/C # 0	T/C # 1	USART
1	6	PB[5]		PCINT13					
2	7	PB[6]		PCINT14					
3	8	PB[7]		PCINT15					
4	9	RESET							
5	10	VCC							
6	11	GND							
7	12	XTAL2							
8	13	XTAL1							
9	14	PD[0]		PCINT24					RxD0
10	15	PD[1]		PCINT25					TxD0
11	16	PD[2]	INT0	PCINT26					RxD1
12	17	PD[3]	INT1	PCINT27					TXD1
13	18	PD[4]		PCINT28				OC1B	XCK1
14	19	PD[5]		PCINT29				OC1A	
15	20	PD[6]		PCINT30			OC2B	ICP1	
16	21	PD[7]		PCINT31			OC2A		
17	-	VCC							RxD2
18	-	GND							TxD2
19	22	PC[0]		PCINT16					
20	23	PC[1]		PCINT17					
21	24	PC[2]		PCINT18					
22	25	PC[3]		PCINT19					
23	26	PC[4]		PCINT20					
24	27	PC[5]		PCINT21					
25	28	PC[6]		PCINT22		TOSC1			
26	29	PC[7]		PCINT23		TOSC2			
27	30	AVCC							
28	31	GND							
29	32	AREF			AREF				

Table 6-1. PORT Function Multiplexing



30

31

32

33

34

33

34

35

36

37

PA[7]

PA[6]

PA[5]

PA[4]

PA[3]

PCINT7

PCINT6

PCINT5

PCINT4

PCINT3

ADC7

ADC6

ADC5

ADC4

ADC3

SPI

MOSI MISO SCK

MISO1 MOSI1

> TCK TMS TDO TDI

SCL SDA JTAG

I2C

Bit 0 – C: Carry Flag

The Carry Flag C indicates a carry in an arithmetic or logic operation. See the Instruction Set Description for detailed information.

General Purpose Register File 8.4.

The Register File is optimized for the AVR Enhanced RISC instruction set. In order to achieve the required performance and flexibility, the following input/output schemes are supported by the Register File:

- One 8-bit output operand and one 8-bit result input
- Two 8-bit output operands and one 8-bit result input
- Two 8-bit output operands and one 16-bit result input
- One 16-bit output operand and one 16-bit result input

Figure 8-2. AVR CPU General Purpose Working Registers

	7	0	Addr.	
	R0		0x00	
	R1		0x01	
	R2		0x02	
	R13		0x0D	
General	R14	ļ	0x0E	
Purpose	R15		0x0F	
Working	R16		0x10	
Registers	R17	,	0x11	
	R26	5	0x1A	X-register Low Byte
	R27	1	0x1B	X-register High Byte
	R28	;	0x1C	Y-register Low Byte
	R29)	0x1D	Y-register High Byte
	R30)	0x1E	Z-register Low Byte
	R31		0x1F	Z-register High Byte

Most of the instructions operating on the Register File have direct access to all registers, and most of them are single cycle instructions. As shown in the figure, each register is also assigned a data memory address, mapping them directly into the first 32 locations of the user Data Space. Although not being physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X-, Y-, and Z-pointer registers can be set to index any register in the file.

8.4.1. The X-register, Y-register, and Z-register

The registers R26...R31 have some added functions to their general purpose usage. These registers are 16-bit address pointers for indirect addressing of the data space. The three indirect address registers X, Y, and Z are defined as described in the figure.



9.6.2. EEPROM Data Register

When addressing I/O Registers as data space using LD and ST instructions, the provided offset must be used. When using the I/O specific commands IN and OUT, the offset is reduced by 0x20, resulting in an I/O address offset within 0x00 - 0x3F.

The device is a complex microcontroller with more peripheral units than can be supported within the 64 locations 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:EEDROffset:0x40Reset:0x00Property:When addressing as I/O Register: address offset is 0x20

Bit	7	6	5	4	3	2	1	0
Γ				EEDI	R[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 – EEDR[7:0]: EEPROM Data

For the EEPROM write operation, the EEDR Register contains the data to be written to the EEPROM in the address given by the EEAR Register. For the EEPROM read operation, the EEDR contains the data read out from the EEPROM at the address given by EEAR.



11. PM - Power Management and Sleep Modes

11.1. Overview

Sleep modes enable the application to shut down unused modules in the MCU, thereby saving power. The device provides various sleep modes allowing the user to tailor the power consumption to the application requirements.

When enabled, the Brown-out Detector (BOD) actively monitors the power supply voltage during the sleep periods. To further save power, it is possible to disable the BOD in some sleep modes. See also BOD Disable.

11.2. Sleep Modes

The following Table shows the different sleep modes, BOD disable ability and their wake-up sources.

	Active Clock Domains			Oscillators		Wake-up Sources					Software				
Sleep Mode	clkCPU	CIKFLASH	clkIO	clkADC	clkasy	Main Clock Source Enabled	Timer Oscillator Enabled	INT and PCINT	TWI Address Match	Timer2	SPM/EEPROM Ready	ADC	WDT	Other I/O	BOD Disable
Idle			Yes	Yes	Yes	Yes	Yes ⁽²⁾	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
ADC Noise Reduction				Yes	Yes	Yes	Yes ⁽²⁾	Yes ⁽³⁾	Yes	Yes ⁽²⁾	Yes	Yes	Yes		
Power-down								Yes ⁽³⁾	Yes				Yes		Yes
Power-save					Yes		Yes ⁽²⁾	Yes ⁽³⁾	Yes	Yes			Yes		Yes
Standby ⁽¹⁾						Yes		Yes ⁽³⁾	Yes				Yes		Yes
Extended Standby					Yes ⁽²⁾	Yes	Yes ⁽²⁾	Yes ⁽³⁾	Yes	Yes			Yes		Yes

 Table 11-1. Active Clock Domains and Wake-up Sources in the Different Sleep Modes.

Note:

- 1. Only recommended with external crystal or resonator selected as clock source.
- 2. If Timer/Counter2 is running in asynchronous mode.
- 3. For INT1 and INT0, only level interrupt.

To enter any of the six sleep modes, the Sleep Enable bit in the Sleep Mode Control Register (SMCR.SE) must be written to '1' and a SLEEP instruction must be executed. Sleep Mode Select bits (SMCR.SM[2:0]) select which sleep mode (Idle, ADC Noise Reduction, Power-down, Power-save, Standby, or Extended Standby) will be activated by the SLEEP instruction.

Note: The block diagram in the section *System Clock and Clock Options* provides an overview over the different clock systems in the device, and their distribution. This figure is helpful in selecting an appropriate sleep mode.

If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU wakes up. The MCU is then halted for four cycles in addition to the start-up time, executes the interrupt routine, and resumes execution from the instruction following SLEEP. The contents of the Register File and SRAM are unaltered when the device wakes up from sleep. If a reset occurs during sleep mode, the MCU wakes up and executes from the Reset Vector.

Related Links

Clock Systems and Their Distribution on page 42



13. Interrupts

13.1. Overview

This section describes the specifics of the interrupt handling of the device. For a general explanation of the AVR interrupt handling, refer to the description of *Reset and Interrupt Handling*.

In general:

- Each Interrupt Vector occupies two instruction words.
- The Reset Vector is affected by the BOOTRST fuse, and the Interrupt Vector start address is affected by the IVSEL bit in MCUCR (MCUCR.IVSEL)

Related Links

Reset and Interrupt Handling on page 27

13.2. Interrupt Vectors in ATmega324P

Table 13-1. Reset and Interrupt Vectors in ATmega324P

Vector No	Program Address ⁽²⁾	Source	Interrupts definition					
1	0x0000 ⁽¹⁾	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset					
2	0x0002	INT0	External Interrupt Request 0					
3	0x0004	INT1	External Interrupt Request 1					
4	0x0006	INT2	External Interrupt Request 2					
5	0x0008	PCINT0	Pin Change Interrupt Request 0					
6	0x000A	PCINT1	Pin Change Interrupt Request 1					
7	0x000C	PCINT2	Pin Change Interrupt Request 2					
8	0x000E	PCINT3	Pin Change Interrupt Request 3					
9	0x0010	WDT	Watchdog Time-out Interrupt					
10	0x0012	TIMER2_COMPA	Timer/Counter2 Compare Match A					
11	0x0014	TIMER2_COMPB	Timer/Coutner2 Compare Match B					
12	0x0016	TIMER2_OVF	Timer/Counter2 Overflow					
13	0x0018	TIMER1_CAPT	Timer/Counter1 Capture Event					
14	0x001A	TIMER1_COMPA	Timer/Counter1 Compare Match A					
15	0x001C	TIMER1_COMPB	Timer/Coutner1 Compare Match B					
16	0x001E	TIMER1_OVF	Timer/Counter1 Overflow					
17	0x0020	TIMER0_COMPA	Timer/Counter0 Compare Match A					
18	0x0022	TIMER0_COMPB	Timer/Coutner0 Compare Match B					
19	0x0024	TIMER0_OVF	Timer/Counter0 Overflow					
20	0x0026	SPI_STC	SPI Serial Transfer Complete					



0x1F004		jmp	EXT_INT1	;	IRQ1 Handler
0x1F036		jmp	SPM_RDY	; ;	SPM Ready Handler
0×1F03E 0×1F03F 0×1F040 0×1F041	RESET:	ldi out ldi out	r16, high (RAMEND) SPH, r16 r16, low (RAMEND) SPL, r16	; ;	Main program start Set Stack Pointer to top of RAM
0x1F042 0x1F043		<mark>sei</mark> <instr></instr>	xxx	;	Enable interrupts

13.3. Register Description

13.3.1. Moving Interrupts Between Application and Boot Space

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



15.4.13. Port D Input Pins Address

When addressing I/O Registers as data space using LD and ST instructions, the provided offset must be used. When using the I/O specific commands IN and OUT, the offset is reduced by 0x20, resulting in an I/O address offset within 0x00 - 0x3F.

The device is a complex microcontroller with more peripheral units than can be supported within the 64 locations 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:
 PIND

 Offset:
 0x29

 Reset:
 N/A

 Property:
 When addressing as I/O Register: address offset is 0x09

Bit	7	6	5	4	3	2	1	0
	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0
Access	R/W							
Reset	x	х	x	х	х	х	х	х

Bits 7:0 – PINDn: Port D Input Pins Address [n = 7:0]

Writing to the pin register provides toggle functionality for IO. Refer to Toggling the Pin.



diagram includes non-inverted and inverted PWM outputs. The small horizontal lines on the TCNT1 slopes mark compare matches between OCR1x and TCNT1. The OC1x Interrupt Flag will be set when a compare match occurs.





Note: The "n" in the register and bit names indicates the device number (n = 1 for Timer/Counter 1), and the "x" indicates Output Compare unit (A/B).

The Timer/Counter Overflow Flag (TOV1) is set each time the counter reaches BOTTOM. When either OCR1A or ICR1 is used for defining the TOP value, the OC1A or ICF1 Flag is set accordingly at the same timer clock cycle as the OCR1x Registers are updated with the double buffer value (at TOP). The Interrupt Flags can be used to generate an interrupt each time the counter reaches the TOP or BOTTOM value.

When changing the TOP value the program must ensure that the new TOP value is higher or equal to the value of all of the Compare Registers. If the TOP value is lower than any of the Compare Registers, a compare match will never occur between the TCNT1 and the OCR1x. Note that when using fixed TOP values, the unused bits are masked to zero when any of the OCR1x registers is written. As illustrated by the third period in the timing diagram, changing the TOP actively while the Timer/Counter is running in the phase correct mode can result in an unsymmetrical output. The reason for this can be found in the time of update of the OCR1x Register. Since the OCR1x update occurs at TOP, the PWM period starts and ends at TOP. This implies that the length of the falling slope is determined by the previous TOP value, while the length of the rising slope is determined by the new TOP value. When these two values differ the two slopes of the period will differ in length. The difference in length gives the unsymmetrical result on the output.

It is recommended to use the phase and frequency correct mode instead of the phase correct mode when changing the TOP value while the Timer/Counter is running. When using a static TOP value, there are practically no differences between the two modes of operation.

In Phase Correct PWM mode, the compare units allow generation of PWM waveforms on the OC1x pins. Writing COM1x[1:0] bits to 0x2 will produce a non-inverted PWM. An inverted PWM output can be generated by writing the COM1x[1:0] to 0x3. The actual OC1x value will only be visible on the port pin if



the data direction for the port pin is set as output (DDR_OC1x). The PWM waveform is generated by setting (or clearing) the OC1x Register at the compare match between OCR1x and TCNT1 when the counter increments, and clearing (or setting) the OC1x Register at compare match between OCR1x and TCNT1 when the counter decrements. The PWM frequency for the output when using Phase Correct PWM can be calculated by the following equation:

 $f_{\text{OCnxPCPWM}} = \frac{f_{\text{clk_I/O}}}{2 \cdot N \cdot \text{TOP}}$

N represents the prescale divider (1, 8, 64, 256, or 1024).

The extreme values for the OCR1x Register represent special cases when generating a PWM waveform output in the Phase Correct PWM mode. If the OCR1x is set equal to BOTTOM the output will be continuously low and if set equal to TOP the output will be continuously high for non-inverted PWM mode. For inverted PWM the output will have the opposite logic values. If OCR1A is used to define the TOP value (WGM1[3:0]=0xB) and COM1A[1:0]=0x1, the OC1A output will toggle with a 50% duty cycle.

17.12.5. Phase and Frequency Correct PWM Mode

The phase and frequency correct Pulse Width Modulation, or phase and frequency correct PWM mode (WGM1[3:0] = 0x8 or 0x9) provides a high resolution phase and frequency correct PWM waveform generation option. The phase and frequency correct PWM mode is, like the phase correct PWM mode, based on a dual-slope operation. The counter counts repeatedly from BOTTOM (0x0000) to TOP and then from TOP to BOTTOM. In non-inverting Compare Output mode, the Output Compare (OC1x) is cleared on the compare match between TCNT1 and OCR1x while up-counting, and set on the compare match while down-counting. In inverting Compare Output mode, the operation is inverted. The dual-slope operation gives a lower maximum operation frequency compared to the single-slope operation. However, due to the symmetric feature of the dual-slope PWM modes, these modes are preferred for motor control applications.

The main difference between the phase correct, and the phase and frequency correct PWM mode is the time the OCR1x Register is updated by the OCR1x Buffer Register, (see Figure 17-8 and the Timing Diagram below).

The PWM resolution for the phase and frequency correct PWM mode can be defined by either ICR1 or OCR1A. The minimum resolution allowed is 2-bit (ICR1 or OCR1A set to 0x0003), and the maximum resolution is 16-bit (ICR1 or OCR1A set to MAX). The PWM resolution in bits can be calculated using the following equation:

 $R_{\rm PFCPWM} = \frac{\log(\rm TOP+1)}{\log(2)}$

In phase and frequency correct PWM mode the counter is incremented until the counter value matches either the value in ICR1 (WGM1[3:0]=0x8), or the value in OCR1A (WGM1[3:0]=0x9). The counter has then reached the TOP and changes the count direction. The TCNT1 value will be equal to TOP for one timer clock cycle. The timing diagram for the phase correct and frequency correct PWM mode is shown below. The figure shows phase and frequency correct PWM mode when OCR1A or ICR1 is used to define TOP. The TCNT1 value is in the timing diagram shown as a histogram for illustrating the dual-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT1 slopes represent compare matches between OCR1x and TCNT1. The OC1x Interrupt Flag will be set when a compare match occurs.



Mode	WGM13	WGM12	WGM11 (DM(M11)(1)	WGM10	Timer/ Counter	ТОР	Update of	TOV1 Flag
					Mode of Operation			Set on
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCR1A	Immediate	MAX
5	0	1	0	1	Fast PWM, 8- bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9- bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10- bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICR1	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCR1A	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICR1	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCR1A	TOP	BOTTOM
12	1	1	0	0	CTC	ICR1	Immediate	MAX
13	1	1	0	1	Reserved	-	-	-
14	1	1	1	0	Fast PWM	ICR1	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCR1A	BOTTOM	TOP

Note:

1. The CTC1 and PWM1[1:0] bit definition names are obsolete. Use the WGM1[3:0] definitions. However, the functionality and location of these bits are compatible with previous versions of the timer.



unpredictable, as it depends on the wake-up time. The recommended procedure for reading TCNT2 is thus as follows:

- 1. Wait for the corresponding Update Busy Flag to be cleared.
- 2. Read TCNT2.
- During asynchronous operation, the synchronization of the Interrupt Flags for the asynchronous timer takes 3 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.

19.10. Timer/Counter Prescaler

Figure 19-12. Prescaler for TC2



clk₁₂

The clock source for TC2 is named clk_{T2S} . It is by default connected to the main system I/O clock $clk_{I/O}$. By writing a '1' to the Asynchronous TC2 bit in the Asynchronous Status Register (ASSR.AS2), TC2 is asynchronously clocked from the TOSC1 pin. This enables use of TC2 as a Real Time Counter (RTC). When AS2 is set, pins TOSC1 and TOSC2 are disconnected from Port B. A crystal can then be connected between the TOSC1 and TOSC2 pins to serve as an independent clock source for TC2. The Oscillator is optimized for use with a 32.768kHz crystal.

For TC2, the possible prescaled selections are: $clk_{T2S}/8$, $clk_{T2S}/32$, $clk_{T2S}/64$, $clk_{T2S}/128$, $clk_{T2S}/256$, and $clk_{T2S}/1024$. Additionally, clk_{T2S} as well as 0 (stop) may be selected. The prescaler is reset by writing a '1' to the Prescaler Reset TC2 bit in the General TC2 Control Register (GTCCR.PSRASY). This allows the user to operate with a defined prescaler.

19.11. Register Description

Atmel

21.12.4. USART Control and Status Register n C

```
        Name:
        UCSR0C, UCSR1C

        Offset:
        0xC2 + n*0x08 [n=0..1]

        Reset:
        0x06

        Property: -
```

Bit	7	6	5	4	3	2	1	0
	UMSEL[1:0]		UPM[1:0]		USBS	UCSZ1 /	UCSZ0 /	UCPOL
						UDORD	UCPHA	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	1	1	0

Bits 7:6 – UMSEL[1:0]: USART Mode Select

These bits select the mode of operation of the USARTn

	Table 21-8.	USART	Mode	Selection
--	-------------	-------	------	-----------

UMSEL[1:0]	Mode
00	Asynchronous USART
01	Synchronous USART
10	Reserved
11	Master SPI (MSPIM) ⁽¹⁾

Note:

1. The UDORD, UCPHA, and UCPOL can be set in the same write operation where the MSPIM is enabled.

Bits 5:4 – UPM[1:0]: USART Parity Mode

These bits enable and set type of parity generation and check. If enabled, the Transmitter will automatically generate and send the parity of the transmitted data bits within each frame. The Receiver will generate a parity value for the incoming data and compare it to the UPM setting. If a mismatch is detected, the UPE Flag in UCSRnA will be set.

Table 21-9.	USART	Mode	Selection
-------------	-------	------	-----------

UPM[1:0]	ParityMode
00	Disabled
01	Reserved
10	Enabled, Even Parity
11	Enabled, Odd Parity

These bits are reserved in Master SPI Mode (MSPIM).

Bit 3 – USBS: USART Stop Bit Select

This bit selects the number of stop bits to be inserted by the Transmitter n. The Receiver ignores this setting.



writing to the REFSn bits in the ADMUX Register. The internal voltage reference must be decoupled by an external capacitor at the AREF pin to improve noise immunity.





The analog input channel is selected by writing to the MUX bits in the ADC Multiplexer Selection register ADMUX.MUX[3:0]. Any of the ADC input pins, as well as GND and a fixed bandgap voltage reference, can be selected as single ended inputs to the ADC. The ADC is enabled by writing a '1' to the ADC Enable bit in the ADC Control and Status Register A (ADCSRA.ADEN). Voltage reference and input channel selections will not take effect until ADEN is set. The ADC does not consume power when ADEN is cleared, so it is recommended to switch off the ADC before entering power saving sleep modes.

If differential channels are selected, the differential gain stage amplifies the voltage difference between the selected input channel pair by the selected gain factor. This amplified value then becomes the analog input to the ADC. If single ended channels are used, the gain amplifier is bypassed altogether.



The ADC generates a 10-bit result which is presented in the ADC Data Registers, ADCH and ADCL. By default, the result is presented right adjusted, but can optionally be presented left adjusted by setting the ADC Left Adjust Result bit ADMUX.ADLAR.

If the result is left adjusted and no more than 8-bit precision is required, it is sufficient to read ADCH. Otherwise, ADCL must be read first, then ADCH, to ensure that the content of the Data Registers belongs to the same conversion: Once ADCL is read, ADC access to Data Registers is blocked. This means that if ADCL has been read, and a second conversion completes before ADCH is read, neither register is updated and the result from the second conversion is lost. When ADCH is read, ADC access to the ADCH and ADCL Registers is re-enabled.

The ADC has its own interrupt which can be triggered when a conversion completes. When ADC access to the Data Registers is prohibited between reading of ADCH and ADCL, the interrupt will trigger even if the result is lost.

Related Links

Power Management and Sleep Modes on page 56 Power Reduction Register on page 59

25.3. Starting a Conversion

A single conversion is started by writing a '0' to the Power Reduction ADC bit in the Power Reduction Register (PRR.PRADC), and writing a '1' to the ADC Start Conversion bit in the ADC Control and Status Register A (ADCSRA.ADSC). ADCS will stay high as long as the conversion is in progress, and will be cleared by hardware when the conversion is completed. If a different data channel is selected while a conversion is in progress, the ADC will finish the current conversion before performing the channel change.

Alternatively, a conversion can be triggered automatically by various sources. Auto Triggering is enabled by setting the ADC Auto Trigger Enable bit (ADCSRA.ADATE). The trigger source is selected by setting the ADC Trigger Select bits in the ADC Control and Status Register B (ADCSRB.ADTS). See the description of the ADCSRB.ADTS for a list of available trigger sources.

When a positive edge occurs on the selected trigger signal, the ADC prescaler is reset and a conversion is started. This provides a method of starting conversions at fixed intervals. If the trigger signal still is set when the conversion completes, a new conversion will not be started. If another positive edge occurs on the trigger signal during conversion, the edge will be ignored. Note that an interrupt flag will be set even if the specific interrupt is disabled or the Global Interrupt Enable bit in the AVR Status REgister (SREG.I) is cleared. A conversion can thus be triggered without causing an interrupt. However, the Interrupt Flag must be cleared in order to trigger a new conversion at the next interrupt event.



Table 25-4. Input Channel and Gain Selections

MUX[4:0]	Single Ended Input	Positive Differential Input	Negative Differential Input	Gain
00000	ADC0	N/A		
00001	ADC1			
00010	ADC2			
00011	ADC3			
00100	ADC4			
00101	ADC5			
00110	ADC6			
00111	ADC7			
01000	N/A	ADC0	ADC0	10x
01001		ADC1	ADC0	10x
01010	-	ADC0	ADC0	200x
01011		ADC1	ADC0	200x
01100	-	ADC2	ADC2	10x
01101		ADC3	ADC2	10x
01110		ADC2	ADC2	200x
01111		ADC3	ADC2	200x
10000	N/A	ADC0	ADC1	1x
10001		ADC1	ADC1	1x
10010		ADC2	ADC1	1x
10011		ADC3	ADC1	1x
10100		ADC4	ADC1	1x
10101		ADC5	ADC1	1x
10110		ADC6	ADC1	1x
11111		ADC7	ADC1	1x
11000		ADC0	ADC2	1x
11000		ADC1	ADC2	1x
11001		ADC2	ADC2	1x
11010		ADC3	ADC2	1x
11011		ADC4	ADC2	1x
11100		ADC5	ADC2	1x



26.14. ATmega324P Boundary-scan Order

The table below shows the Scan order between TDI and TDO when the Boundary-scan Chain is selected as data path. Bit 0 is the LSB; the first bit scanned in, and the first bit scanned out. The scan order follows the pin-out order as far as possible. Therefore, the bits of Port A are scanned in the opposite bit order of the other ports.

Exceptions from the rules are the scan chains for the analog circuits, which constitute the most significant bits of the scan chain regardless of which physical pin they are connected to. In Figure 26-5, PXn. Data corresponds to FF0, PXn. Control corresponds to FF1, and PXn. Pullup_enable corresponds to FF2. Bit 2, 3, 4, and 5 of Port C is not in the scan chain, since these pins constitute the TAP pins when the JTAG is enabled.

Bit Number	Signal Name	Module
56	PB0.Data	Port B
55	PB0.Control	
54	PB1.Data	
53	PB1.Control	
52	PB2.Data	
51	PB2.Control	
50	PB3.Data	
49	PB3.Control	
48	PB4.Data	
47	PB4.Control	
46	PB5.Data	
45	PB5.Control	
44	PB6.Data	
43	PB6.Control	
42	PB7.Data	
41	PB7.Control	
40	RSTT	Reset Logic (Observe Only)

Table 26-3. ATmega324P Boundary-scan Order (TBD)



BLB1 Mode	BLB12	BLB11	Protection
3	0	0	SPM is not allowed to write to the Boot Loader section, and LPM executing from the Application section is not allowed to read from the Boot Loader section. If Interrupt Vectors are placed in the Application section, interrupts are disabled while executing from the Boot Loader section.
4	0	1	LPM executing from the Application section is not allowed to read from the Boot Loader section. If Interrupt Vectors are placed in the Application section, interrupts are disabled while executing from the Boot Loader section.

Note: "1" means unprogrammed, "0" means programmed.

27.7. Addressing the Flash During Self-Programming

The Z-pointer is used to address the SPM commands. The Z pointer consists of the Z-registers ZL and ZH in the register file. The number of bits actually used is implementation dependent.

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

Since the Flash is organized in pages, the Program Counter can be treated as having two different sections. One section, consisting of the least significant bits, is addressing the words within a page, while the most significant bits are addressing the pages. This is shown in the following figure. The Page Erase and Page Write operations are 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. Once a programming operation is initiated, the address is latched and the Z-pointer can be used for other operations.

The only SPM 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. The LPM instruction does also use 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.





Figure 28-4. Programming the FUSES Waveforms

28.8.11. Programming the Lock Bits

The algorithm for programming the Lock bits is as follows (Please refer to Programming the Flash for details on Command and Data loading):

- 1. Step A: Load Command "0010 0000".
- Step C: Load Data Low Byte. Bit n = "0" programs the Lock bit. If LB mode 3 is programmed (LB1 and LB2 is programmed), it is not possible to program the Boot Lock bits by any External Programming mode.
- 3. Give \overline{WR} a negative pulse and wait for RDY/BSY to go high.

The Lock bits can only be cleared by executing Chip Erase.

28.8.12. Reading the Fuse and Lock Bits

The algorithm for reading the Fuse and Lock bits is as follows (Please refer to Programming the Flash for details on Command loading):

- 1. Step A: Load Command "0000 0100".
- 2. Set \overline{OE} to "0", BS2 to "0" and BS1 to "0". The status of the Fuse Low bits can now be read at DATA ("0" means programmed).
- 3. Set \overline{OE} to "0", BS2 to "1" and BS1 to "1". The status of the Fuse High bits can now be read at DATA ("0" means programmed).
- 4. Set \overline{OE} to "0", BS2 to "1", and BS1 to "0". The status of the Extended Fuse bits can now be read at DATA ("0" means programmed).
- 5. Set \overline{OE} to "0", BS2 to "0" and BS1 to "1". The status of the Lock bits can now be read at DATA ("0" means programmed).
- 6. Set OE to "1".



29. Electrical Characteristics

29.1. Absolute Maximum Ratings

Table 29-1. Absolute Maximum Ratings

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin except RESET with respect to Ground	-0.5V to V_{CC} +0.5V
Voltage on 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

Note: 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.

29.2. DC Characteristics

Table 29-2. Common DC characteristics T_A = -40°C to 105°C, V_{CC} = 1.8V to 5.5V (unless otherwise noted)

Symbol	Parameter	Condition	Min.	Тур.	Max.	Units
V _{IL}	Input Low Voltage, except XTAL1	V _{CC} = 1.8V - 2.4V	-0.5		0.2V _{CC} ⁽¹⁾	V
and RESET pin		V _{CC} = 2.4V - 5.5V	-0.5		0.3V _{CC} ⁽¹⁾	
V _{IL1}	Input Low Voltage, XTAL1 pin	V _{CC} = 1.8V - 5.5V	-0.5		0.1V _{CC} ⁽¹⁾	V
V _{IL2}	Input Low Voltage, RESET pin	V _{CC} = 1.8V - 5.5V	-0.5		0.1V _{CC} ⁽¹⁾	V
V _{IH}	Input High Voltage, except	V _{CC} = 1.8V - 2.4V	0.7V _{CC} ⁽²⁾		V _{CC} + 0.5	V
	XIAL1 and RESET pins	V _{CC} = 2.4V - 5.5V	$0.6V_{CC}^{(2)}$		V _{CC} + 0.5	
V _{IH1}	Input High Voltage, XTAL1 pin	V _{CC} = 1.8V - 2.4V	0.8V _{CC} ⁽²⁾		V _{CC} + 0.5	V
		V _{CC} = 2.4V - 5.5V	$0.7 V_{CC}^{(2)}$		V _{CC} + 0.5	
V _{IH2}	Input High Voltage, RESET pin	V _{CC} = 1.8V - 5.5V	0.9V _{CC} ⁽²⁾		V _{CC} + 0.5	V



29.3. Speed Grades

Maximum frequency is depending on V_{CC}. The Maximum Frequency vs. V_{CC} curve is linear between 1.8V < V_{CC} < 2.7V and between 2.7V < V_{CC} < 4.5V.





29.4. Clock Characteristics

Related Links

Calibrated Internal RC Oscillator on page 48

29.4.1. Calibration Accuracy of Internal RC Oscillator

Table 29-5. Calibration Accuracy of Internal RC Oscillator.

	Frequency	V _{cc}	Temperature	Calibration accuracy
Factory calibration	8.0MHz	3V	25°C	±10%
User calibration	7.3 - 8.1MHz	1.8 - 5.5V ⁽¹⁾	-40°C - 85°C	±1%
		2.7 - 5.5 ⁽²⁾		

- 1. Voltage range for ATmega324PV.
- 2. Voltage range for ATmega324P.

