E·XFL



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

Product Status	Obsolete
Core Processor	AVR
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	23
Program Memory Size	4KB (2K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 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-VFQFN Exposed Pad
Supplier Device Package	32-VQFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega48-20mi

Email: info@E-XFL.COM

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

In ATmega48, there is no Read-While-Write support and no separate Boot Loader Section. The SPM instruction can execute from the entire Flash.

2.3 Pin Descriptions

2.3.	1	VCC

- Digital supply voltage.
- 2.3.2 GND

Ground.

2.3.3 Port B (PB7..0) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB7..6 is used as TOSC2..1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

The various special features of Port B are elaborated in "Alternate Functions of Port B" on page 71 and "System Clock and Clock Options" on page 25.

2.3.4 Port C (PC5..0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5..0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

2.3.5 PC6/RESET

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in Table 8-1 on page 44. Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated in "Alternate Functions of Port C" on page 75.

2.3.6 Port D (PD7..0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up



When the write access time has elapsed, the EEPE bit is cleared by hardware. The user software can poll this bit and wait for a zero before writing the next byte. When EEPE has been set, the CPU is halted for two cycles before the next instruction is executed.

• Bit 0 – EERE: EEPROM Read Enable

The EEPROM Read Enable Signal EERE is the read strobe to the EEPROM. When the correct address is set up in the EEAR Register, the EERE bit must be written to a logic one to trigger the EEPROM read. The EEPROM read access takes one instruction, and the requested data is available immediately. When the EEPROM is read, the CPU is halted for four cycles before the next instruction is executed.

The user should poll the EEPE bit before starting the read operation. If a write operation is in progress, it is neither possible to read the EEPROM, nor to change the EEAR Register.

The calibrated Oscillator is used to time the EEPROM accesses. Table 5-2 lists the typical programming time for EEPROM access from the CPU.

 Table 5-2.
 EEPROM Programming Time

Symbol	Number of Calibrated RC Oscillator Cycles	Typ Programming Time
EEPROM write (from CPU)	26,368	3.3 ms

The following code examples show one assembly and one C function for writing to the EEPROM. The examples assume that interrupts are controlled (e.g. by disabling interrupts globally) so that no interrupts will occur during execution of these functions. The examples also assume that no Flash Boot Loader is present in the software. If such code is present, the EEPROM write function must also wait for any ongoing SPM command to finish.



6.5 Low Frequency Crystal Oscillator

The device can utilize a 32.768 kHz watch crystal as clock source by a dedicated Low Frequency Crystal Oscillator. The crystal should be connected as shown in Figure 6-2. When this Oscillator is selected, start-up times are determined by the SUT Fuses and CKSEL0 as shown in Table 6-7.

Power Conditions	Start-up Time from Power-down and Power-save	Additional Delay from Reset (V _{CC} = 5.0V)	CKSEL0	SUT10
BOD enabled	1K CK	14CK ⁽¹⁾	0	00
Fast rising power	1K CK	14CK + 4.1 ms ⁽¹⁾	0	01
Slowly rising power	1K CK	14CK + 65 ms ⁽¹⁾	0	10
	Reserved		0	11
BOD enabled	32K CK	14CK	1	00
Fast rising power	32K CK	14CK + 4.1 ms	1	01
Slowly rising power	32K CK	14CK + 65 ms	1	10
	Reserved		1	11

 Table 6-7.
 Start-up Times for the Low Frequency Crystal Oscillator Clock Selection

Note: 1. These options should only be used if frequency stability at start-up is not important for the application.

6.6 Calibrated Internal RC Oscillator

The calibrated internal RC Oscillator by default provides a 8.0 MHz clock. The frequency is nominal value at 3V and 25°C. The device is shipped with the CKDIV8 Fuse programmed. See "System Clock Prescaler" on page 34 for more details. This clock may be selected as the system clock by programming the CKSEL Fuses as shown in Table 6-8. If selected, it will operate with no external components. During reset, hardware loads the calibration byte into the OSCCAL Register and thereby automatically calibrates the RC Oscillator. At 3V and 25°C, this calibration gives a frequency of 8 MHz $\pm 10\%$. The oscillator can be calibrated to any frequency in the range 7.3 - 8.1 MHz within $\pm 2\%$ accuracy, by changing the OSCCAL register. When this Oscillator is used as the chip clock, the Watchdog Oscillator will still be used for the Watchdog Timer and for the Reset Time-out. For more information on the pre-programmed calibration value, see the section "Calibration Byte" on page 284.

Table 6-8.	Internal Calibrated RC Oscillator Opera	ating Modes ⁽¹⁾⁽³⁾
------------	---	-------------------------------

Frequency Range ⁽²⁾ (MHz)	CKSEL30		
7.3 - 8.1	0010		

Notes: 1. The device is shipped with this option selected.

2. The frequency ranges are preliminary values. Actual values are TBD.

 If 8 MHz frequency exceeds the specification of the device (depends on V_{CC}), the CKDIV8 Fuse can be programmed in order to divide the internal frequency by 8.



If Timer/Counter2 is enabled, it will keep running during sleep. The device can wake up from either Timer Overflow or Output Compare event from Timer/Counter2 if the corresponding Timer/Counter2 interrupt enable bits are set in TIMSK2, and the Global Interrupt Enable bit in SREG is set.

If Timer/Counter2 is not running, Power-down mode is recommended instead of Power-save mode.

The Timer/Counter2 can be clocked both synchronously and asynchronously in Power-save mode. If Timer/Counter2 is not using the asynchronous clock, the Timer/Counter Oscillator is stopped during sleep. If Timer/Counter2 is not using the synchronous clock, the clock source is stopped during sleep. Note that even if the synchronous clock is running in Power-save, this clock is only available for Timer/Counter2.

7.5 Standby Mode

When the SM2..0 bits are 110 and an external crystal/resonator clock option is selected, the SLEEP instruction makes the MCU enter Standby mode. This mode is identical to Power-down with the exception that the Oscillator is kept running. From Standby mode, the device wakes up in six clock cycles.

	Active Clock Domains				Oscil	Oscillators Wake-up Source				es				
Sleep Mode	olk _{CPU}	olk _{FLASH}	clk _{IO}	clk _{ADC}	clk _{ASY}	Main Clock Source Enabled	Timer Oscillator Enabled	INT1, INT0 and Pin Change	TWI Address Match	Timer2	SPM/EEPROM Ready	ADC	WDT	Other/O
Idle			Х	Х	Х	Х	X ⁽²⁾	Х	Х	Х	Х	Х	Х	Х
ADC Noise Reduction				х	х	х	X ⁽²⁾	X ⁽³⁾	х	х	х	х	х	
Power-down								X ⁽³⁾	Х				Х	
Power-save					Х		Х	X ⁽³⁾	Х	Х			Х	
Standby ⁽¹⁾						Х		X ⁽³⁾	Х				Х	

 Table 7-2.
 Active Clock Domains and Wake-up Sources in the Different Sleep Modes.

Notes: 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.

7.6 Power Reduction Register

The Power Reduction Register, PRR, provides a method to stop the clock to individual peripherals to reduce power consumption. The current state of the peripheral is frozen and the I/O registers can not be read or written. Resources used by the peripheral when stopping the clock will remain occupied, hence the peripheral should in most cases be disabled before stopping the clock. Waking up a module, which is done by clearing the bit in PRR, puts the module in the same state as before shutdown.

Module shutdown can be used in Idle mode and Active mode to significantly reduce the overall power consumption. See "Power-Down Supply Current" on page 315 for examples. In all other sleep modes, the clock is already stopped.



Table 12-7 shows the COM0B1:0 bit functionality when the WGM02:0 bits are set to phase correct PWM mode.

COM0B1	COM0B0	Description
0	0	Normal port operation, OC0B disconnected.
0	1	Reserved
1	0	Clear OC0B on Compare Match when up-counting. Set OC0B on Compare Match when down-counting.
1	1	Set OC0B on Compare Match when up-counting. Clear OC0B on Compare Match when down-counting.

 Table 12-7.
 Compare Output Mode, Phase Correct PWM Mode⁽¹⁾

Note: 1. A special case occurs when OCR0B equals TOP and COM0B1 is set. In this case, the Compare Match is ignored, but the set or clear is done at TOP. See "Phase Correct PWM Mode" on page 96 for more details.

• Bits 3, 2 - Res: Reserved Bits

These bits are reserved bits in the ATmega48/88/168 and will always read as zero.

• Bits 1:0 - WGM01:0: Waveform Generation Mode

Combined with the WGM02 bit found in the TCCR0B Register, these bits control the counting sequence of the counter, the source for maximum (TOP) counter value, and what type of waveform generation to be used, see Table 12-8. Modes of operation supported by the Timer/Counter unit are: Normal mode (counter), Clear Timer on Compare Match (CTC) mode, and two types of Pulse Width Modulation (PWM) modes (see "Modes of Operation" on page 93).

Mode	WGM02	WGM01	WGM00	Timer/Counter Mode of Operation	ТОР	Update of OCRx at	TOV Flag Set on ⁽¹⁾⁽²⁾
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	0	1	0	СТС	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	TOP	MAX
4	1	0	0	Reserved	_	_	_
5	1	0	1	PWM, Phase Correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	_	_	_
7	1	1	1	Fast PWM	OCRA	TOP	TOP

 Table 12-8.
 Waveform Generation Mode Bit Description

Notes: 1. MAX = 0xFF 2. BOTTOM = 0x00



cleared by software (writing a logical one to the I/O bit location). For measuring frequency only, the clearing of the ICF1 Flag is not required (if an interrupt handler is used).

13.6 Output Compare Units

The 16-bit comparator continuously compares TCNT1 with the *Output Compare Register* (OCR1x). If TCNT equals OCR1x the comparator signals a match. A match will set the *Output Compare Flag* (OCF1x) at the next timer clock cycle. If enabled (OCIE1x = 1), the Output Compare Flag generates an Output Compare interrupt. The OCF1x Flag is automatically cleared when the interrupt is executed. Alternatively the OCF1x Flag can be cleared by software by writing a logical one to its I/O bit location. The Waveform Generator uses the match signal to generate an output according to operating mode set by the *Waveform Generation mode* (WGM13:0) bits and *Compare Output mode* (COM1x1:0) bits. The TOP and BOTTOM signals are used by the Waveform Generator for handling the special cases of the extreme values in some modes of operation (See Section "13.8" on page 118.)

A special feature of Output Compare unit A allows it to define the Timer/Counter TOP value (i.e., counter resolution). In addition to the counter resolution, the TOP value defines the period time for waveforms generated by the Waveform Generator.

Figure 13-4 shows a block diagram of the Output Compare unit. The small "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 elements of the block diagram that are not directly a part of the Output Compare unit are gray shaded.



Figure 13-4. Output Compare Unit, Block Diagram

The OCR1x Register is double buffered when using any of the twelve *Pulse Width Modulation* (PWM) modes. For the Normal and *Clear Timer on Compare* (CTC) modes of operation, the double buffering is disabled. The double buffering synchronizes the update of the OCR1x Compare Register to either TOP or BOTTOM of the counting sequence. The synchronization





The PWM resolution for fast PWM can be fixed to 8-, 9-, or 10-bit, or 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 by using the following equation:

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

In fast PWM mode the counter is incremented until the counter value matches either one of the fixed values 0x00FF, 0x01FF, or 0x03FF (WGM13:0 = 5, 6, or 7), the value in ICR1 (WGM13:0 = 14), or the value in OCR1A (WGM13:0 = 15). The counter is then cleared at the following timer clock cycle. The timing diagram for the fast PWM mode is shown in Figure 13-7. The figure shows fast 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 single-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.

Figure 13-7. Fast PWM Mode, Timing Diagram



The Timer/Counter Overflow Flag (TOV1) is set each time the counter reaches TOP. In addition the OC1A or ICF1 Flag is set at the same timer clock cycle as TOV1 is set when either OCR1A or ICR1 is used for defining the TOP value. If one of the interrupts are enabled, the interrupt handler routine can be used for updating the TOP and compare values.

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 are written.

The procedure for updating ICR1 differs from updating OCR1A when used for defining the TOP value. The ICR1 Register is not double buffered. This means that if ICR1 is changed to a low value when the counter is running with none or a low prescaler value, there is a risk that the new ICR1 value written is lower than the current value of TCNT1. The result will then be that the counter will miss the compare match at the TOP value. The counter will then have to count to the MAX value (0xFFFF) and wrap around starting at 0x0000 before the compare match can occur. The OCR1A Register however, is double buffered. This feature allows the OCR1A I/O location

16. Serial Peripheral Interface – SPI

The Serial Peripheral Interface (SPI) allows high-speed synchronous data transfer between the ATmega48/88/168 and peripheral devices or between several AVR devices. The ATmega48/88/168 SPI includes the following features:

- Full-duplex, Three-wire Synchronous Data Transfer
- Master or Slave Operation
- LSB First or MSB First Data Transfer
- Seven Programmable Bit Rates
- End of Transmission Interrupt Flag
- Write Collision Flag Protection
- Wake-up from Idle Mode
- Double Speed (CK/2) Master SPI Mode

The USART can also be used in Master SPI mode, see "USART in SPI Mode" on page 196. The PRSPI bit in "Power Reduction Register - PRR" on page 40 must be written to zero to enable SPI module.





Note: 1. Refer to Figure 1-1 on page 2, and Table 10-3 on page 71 for SPI pin placement.

The interconnection between Master and Slave CPUs with SPI is shown in Figure 16-2. The system consists of two shift Registers, and a Master clock generator. The SPI Master initiates the







When the clock recovery logic detects a high (idle) to low (start) transition on the RxDn line, the start bit detection sequence is initiated. Let sample 1 denote the first zero-sample as shown in the figure. The clock recovery logic then uses samples 8, 9, and 10 for Normal mode, and samples 4, 5, and 6 for Double Speed mode (indicated with sample numbers inside boxes on the figure), to decide if a valid start bit is received. If two or more of these three samples have logical high levels (the majority wins), the start bit is rejected as a noise spike and the Receiver starts looking for the next high to low-transition. If however, a valid start bit is detected, the clock recovery logic is synchronized and the data recovery can begin. The synchronization process is repeated for each start bit.

17.7.2 Asynchronous Data Recovery

When the receiver clock is synchronized to the start bit, the data recovery can begin. The data recovery unit uses a state machine that has 16 states for each bit in Normal mode and eight states for each bit in Double Speed mode. Figure 17-6 shows the sampling of the data bits and the parity bit. Each of the samples is given a number that is equal to the state of the recovery unit.





The decision of the logic level of the received bit is taken by doing a majority voting of the logic value to the three samples in the center of the received bit. The center samples are emphasized on the figure by having the sample number inside boxes. The majority voting process is done as follows: If two or all three samples have high levels, the received bit is registered to be a logic 1. If two or all three samples have low levels, the received bit is registered to be a logic 0. This majority voting process acts as a low pass filter for the incoming signal on the RxDn pin. The receivery process is then repeated until a complete frame is received. Including the first stop bit. Note that the Receiver only uses the first stop bit of a frame.

Figure 17-7 shows the sampling of the stop bit and the earliest possible beginning of the start bit of the next frame.



ATmega48/88/168

Status Code		Applica	tion Softv	vare Resp	onse		
(TWSR)	Status of the 2-wire Serial Bus			To	FWCR		
are 0	ware	To/from TWDR	STA	STO	TWIN T	TWE A	Next Action Taken by TWI Hardware
0x60	Own SLA+W has been received; ACK has been returned	No TWDR action or	х	0	1	0	Data byte will be received and NOT ACK will be returned
		No TWDR action	Х	0	1	1	Data byte will be received and ACK will be returned
0x68	Arbitration lost in SLA+R/W as Master; own SLA+W has been	No TWDR action or	X	0	1	0	Data byte will be received and NOT ACK will be returned
070	Received, ACK has been returned	No TWDR action	X	0	1	1	Data byte will be received and ACK will be returned
0x70	received; ACK has been returned	No TWDR action or	x	0	1	1	Data byte will be received and NOT ACK will be returned
0x78	Arbitration lost in SLA+R/W as	No TWDR action or	X	0	1	0	Data byte will be received and NOT ACK will be
	Master; General call address has been received; ACK has been returned	No TWDR action	х	0	1	1	returned Data byte will be received and ACK will be returned
0x80	Previously addressed with own SLA+W; data has been received;	Read data byte or	х	0	1	0	Data byte will be received and NOT ACK will be returned
	ACK has been returned	Read data byte	Х	0	1	1	Data byte will be received and ACK will be returned
0x88	Previously addressed with own SLA+W; data has been received;	Read data byte or	0	0	1	0	Switched to the not addressed Slave mode; no recognition of own SLA or GCA
	NOT ACK has been returned	Read data byte or	0	0	1	1	Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = "1"
		Read data byte or	1	0	1	0	Switched to the not addressed Slave mode; no recognition of own SLA or GCA; a START condition will be transmitted when the bus becomes free
		Read data byte	1	0	1	1	Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = "1"; a START condition will be transmitted when the bus becomes free
0x90	Previously addressed with	Read data byte or	Х	0	1	0	Data byte will be received and NOT ACK will be
	general call; data has been re- ceived; ACK has been returned	Read data byte	х	0	1	1	returned Data byte will be received and ACK will be returned
0x98	Previously addressed with	Read data byte or	0	0	1	0	Switched to the not addressed Slave mode;
	received; NOT ACK has been returned	Read data byte or	0	0	1	1	Switched to the not addressed Slave mode; own SLA will be recognized;
		Read data byte or	1	0	1	0	GCA will be recognized if TWGCE = "1" Switched to the not addressed Slave mode; no recognition of own SLA or GCA; a START condition will be transmitted when the bus
		Read data byte	1	0	1	1	Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = "1"; a START condition will be transmitted when the bus becomes free
0xA0	A STOP condition or repeated	No action	0	0	1	0	Switched to the not addressed Slave mode;
	received while still addressed as Slave		0	0	1	1	Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCF = "1"
			1	0	1	0	Switched to the not addressed Slave mode; no recognition of own SLA or GCA; a START condition will be transmitted when the bus becomes free
			1	0	1	1	Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = "1"; a START condition will be transmitted when the bus becomes free

Table 19-5. Status Codes for Slave Receiver Mode



Analog Ground Plane

and ADC5) will only affect the conversion on ADC4 and ADC5 and not the other ADC channels.





ADC Accuracy Definitions 21.5.3

An n-bit single-ended ADC converts a voltage linearly between GND and V_{RFF} in 2ⁿ steps (LSBs). The lowest code is read as 0, and the highest code is read as 2ⁿ-1.

Several parameters describe the deviation from the ideal behavior:

 Offset: The deviation of the first transition (0x000 to 0x001) compared to the ideal transition (at 0.5 LSB). Ideal value: 0 LSB.





; return to RWW section ; verify that RWW section is safe to read Return: in temp1, SPMCSR sbrs temp1, RWWSB ; If RWWSB is set, the RWW section is not ready yet ret ; re-enable the RWW section ldi spmcrval, (1<<RWWSRE) | (1<<SELFPRGEN)</pre> call Do_spm rjmp Return Do_spm: ; check for previous SPM complete Wait_spm: in temp1, SPMCSR sbrc temp1, SELFPRGEN rjmp Wait_spm ; input: spmcrval determines SPM action ; disable interrupts if enabled, store status in temp2, SREG cli ; check that no EEPROM write access is present Wait_ee: sbic EECR, EEPE rjmp Wait_ee ; SPM timed sequence out SPMCSR, spmcrval spm ; restore SREG (to enable interrupts if originally enabled) out SREG, temp2 ret

24.7.13 ATmega88 Boot Loader Parameters

In Table 24-6 through Table 24-8, the parameters used in the description of the self programming are given.

BOOTSZ1	BOOTSZ0	Boot Size	Pages	Application Flash Section	Boot Loader Flash Section	End Application Section	Boot Reset Address (Start Boot Loader Section)
1	1	128 words	4	0x000 - 0xF7F	0xF80 - 0xFFF	0xF7F	0xF80
1	0	256 words	8	0x000 - 0xEFF	0xF00 - 0xFFF	0xEFF	0xF00
0	1	512 words	16	0x000 - 0xDFF	0xE00 - 0xFFF	0xDFF	0xE00
0	0	1024 words	32	0x000 - 0xBFF	0xC00 - 0xFFF	0xBFF	0xC00

 Table 24-6.
 Boot Size Configuration, ATmega88



25.4 Calibration Byte

The ATmega48/88/168 has a byte calibration value for the internal RC Oscillator. This byte resides in the high byte of address 0x000 in the signature address space. During reset, this byte is automatically written into the OSCCAL Register to ensure correct frequency of the calibrated RC Oscillator.

25.5 Page Size

Device	Flash Size	Page Size	PCWORD	No. of Pages	PCPAGE	PCMSB
ATmega48	2K words (4K bytes)	32 words	PC[4:0]	64	PC[10:5]	10
ATmega88	4K words (8K bytes)	32 words	PC[4:0]	128	PC[11:5]	11
ATmega168	8K words (16K bytes)	64 words	PC[5:0]	128	PC[12:6]	12

 Table 25-8.
 No. of Words in a Page and No. of Pages in the Flash

Device	EEPROM Size	Page Size	PCWORD	No. of Pages	PCPAGE	EEAMSB
ATmega48	256 bytes	4 bytes	EEA[1:0]	64	EEA[7:2]	7
ATmega88	512 bytes	4 bytes	EEA[1:0]	128	EEA[8:2]	8
ATmega168	512 bytes	4 bytes	EEA[1:0]	128	EEA[8:2]	8

25.6 Parallel Programming Parameters, Pin Mapping, and Commands

This section describes how to parallel program and verify Flash Program memory, EEPROM Data memory, Memory Lock bits, and Fuse bits in the ATmega48/88/168. Pulses are assumed to be at least 250 ns unless otherwise noted.

25.6.1 Signal Names

In this section, some pins of the ATmega48/88/168 are referenced by signal names describing their functionality during parallel programming, see Figure 25-1 and Table 25-10. Pins not described in the following table are referenced by pin names.

The XA1/XA0 pins determine the action executed when the XTAL1 pin is given a positive pulse. The bit coding is shown in Table 25-12.

When pulsing \overline{WR} or \overline{OE} , the command loaded determines the action executed. The different Commands are shown in Table 25-13.



Table 25-12. XA1 and XA0 Coding

XA1	XA0	Action when XTAL1 is Pulsed
0	0	Load Flash or EEPROM Address (High or low address byte determined by BS1).
0	1	Load Data (High or Low data byte for Flash determined by BS1).
1	0	Load Command
1	1	No Action, Idle

Table 25-13. Command Byte Bit Coding

Command Byte	Command Executed
1000 0000	Chip Erase
0100 0000	Write Fuse bits
0010 0000	Write Lock bits
0001 0000	Write Flash
0001 0001	Write EEPROM
0000 1000	Read Signature Bytes and Calibration byte
0000 0100	Read Fuse and Lock bits
0000 0010	Read Flash
0000 0011	Read EEPROM

25.7 Parallel Programming

25.7.1 Enter Programming Mode

The following algorithm puts the device in Parallel (High-voltage) Programming mode:

- 1. Set Prog_enable pins listed in Table 25-11 on page 285 to "0000", RESET pin to 0V and V_{CC} to 0V.
- 2. Apply 4.5 5.5V between V_{CC} and GND.

Ensure that V_{CC} reaches at least 1.8V within the next 20 μ s.

- 3. Wait 20 60 µs, and apply 11.5 12.5V to RESET.
- 4. Keep the Prog_enable pins unchanged for at least 10µs after the High-voltage has been applied to ensure the Prog_enable Signature has been latched.
- 5. Wait at least 300 µs before giving any parallel programming commands.
- 6. Exit Programming mode by power the device down or by bringing RESET pin to 0V.

If the rise time of the V_{CC} is unable to fulfill the requirements listed above, the following alternative algorithm can be used.

- 1. Set Prog_enable pins listed in Table 25-11 on page 285 to "0000", RESET pin to 0V and V_{CC} to 0V.
- 2. Apply 4.5 5.5V between V_{CC} and GND.
- 3. Monitor V_{CC} , and as soon as V_{CC} reaches 0.9 1.1V, apply 11.5 12.5V to RESET.

- 4. Keep the Prog_enable pins unchanged for at least 10µs after the High-voltage has been applied to ensure the Prog_enable Signature has been latched.
- 5. Wait until V_{CC} actually reaches 4.5 -5.5V before giving any parallel programming commands.
- 6. Exit Programming mode by power the device down or by bringing RESET pin to 0V.

25.7.2 Considerations for Efficient Programming

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

- The command needs only be loaded once when writing or reading multiple memory locations.
- Skip writing the data value 0xFF, that is the contents of the entire EEPROM (unless the EESAVE Fuse is programmed) and Flash after a Chip Erase.
- Address high byte needs only be loaded before programming or reading a new 256 word window in Flash or 256 byte EEPROM. This consideration also applies to Signature bytes reading.

25.7.3 Chip Erase

The Chip Erase will erase the Flash and EEPROM⁽¹⁾ memories plus Lock bits. The Lock bits are not reset until the program memory has been completely erased. The Fuse bits are not changed. A Chip Erase must be performed before the Flash and/or EEPROM are reprogrammed.

Note: 1. The EEPRPOM memory is preserved during Chip Erase if the EESAVE Fuse is programmed. Load Command "Chip Erase"

- 1. Set XA1, XA0 to "10". This enables command loading.
- 2. Set BS1 to "0".
- 3. Set DATA to "1000 0000". This is the command for Chip Erase.
- 4. Give XTAL1 a positive pulse. This loads the command.
- 5. Give WR a negative pulse. This starts the Chip Erase. RDY/BSY goes low.
- 6. Wait until RDY/BSY goes high before loading a new command.

25.7.4 Programming the Flash

The Flash is organized in pages, see Table 25-8 on page 284. When programming the Flash, the program data is latched into a page buffer. This allows one page of program data to be programmed simultaneously. The following procedure describes how to program the entire Flash memory:

A. Load Command "Write Flash"

- 1. Set XA1, XA0 to "10". This enables command loading.
- 2. Set BS1 to "0".
- 3. Set DATA to "0001 0000". This is the command for Write Flash.
- 4. Give XTAL1 a positive pulse. This loads the command.
- B. Load Address Low byte
- 1. Set XA1, XA0 to "00". This enables address loading.
- 2. Set BS1 to "0". This selects low address.
- 3. Set DATA = Address low byte (0x00 0xFF).



- 6. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO.
- 7. At the end of the programming session, **RESET** can be set high to commence normal operation.
- Power-off sequence (if needed): Set RESET to "1". Turn V_{CC} power off.

Table 25-16. Typical Wait Delay Before Writing the Next Flash or EEPROM Location

Symbol	Minimum Wait Delay
t _{WD_FLASH}	4.5 ms
t _{WD_EEPROM}	3.6 ms
t _{wd_erase}	9.0 ms

Figure 25-11. Serial Programming Waveforms



Table 25-17. Serial Programming Instruction Set

		Instruction Format			
Instruction	Byte 1	Byte 2	Byte 3	Byte4	Operation
Programming Enable	1010 1100	0101 0011	XXXX XXXX	XXXX XXXX	Enable Serial Programming after RESET goes low.
Chip Erase	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Chip Erase EEPROM and Flash.
Read Program Memory	0010 H 000	000 a aaaa	bbbb bbbb	0000 0000	Read H (high or low) data o from Program memory at word address a : b .
Load Program Memory Page	0100 H 000	000x xxxx	xx bb bbbb	1111 1111	Write H (high or low) data i to Program Memory page at word address b . Data low byte must be loaded before Data high byte is applied within the same address.
Write Program Memory Page	0100 1100	000 a aaaa	bbxx xxxx	xxxx xxxx	Write Program Memory Page at address a : b .
Read EEPROM Memory	1010 0000	000x xx aa	bbbb bbbb	0000 0000	Read data o from EEPROM memory at address a : b .
Write EEPROM Memory	1100 0000	000x xx aa	bbbb bbbb	1111 1111	Write data i to EEPROM memory at address a:b.





Figure 26-4. 2-wire Serial Bus Timing



26.7 SPI Timing Characteristics

See Figure 26-5 and Figure 26-6 for details.

Table 26-3.SPI Timing Parameters

	Description	Mode	Min	Тур	Мах	
1	SCK period	Master		See Table 16-4		
2	SCK high/low	Master		50% duty cycle		
3	Rise/Fall time	Master		3.6		
4	Setup	Master		10		
5	Hold	Master		10		
6	Out to SCK	Master		0.5 • t _{sck}		
7	SCK to out	Master		10		
8	SCK to out high	Master		10		
9	SS low to out	Slave		15		
10	SCK period	Slave	4 ∙ t _{ck}			ns
11	SCK high/low ⁽¹⁾	Slave	2 ∙ t _{ck}			
12	Rise/Fall time	Slave			1600	
13	Setup	Slave	10			
14	Hold	Slave	t _{ck}			
15	SCK to out	Slave		15		
16	SCK to SS high	Slave	20			
17	SS high to tri-state	Slave		10		
18	SS low to SCK	Slave	20			

Note: 1. In SPI Programming mode the minimum SCK high/low period is:

- 2 t_{CLCL} for f_{CK} < 12 MHz
- 3 t_{CLCL} for f_{CK} > 12 MHz
- 2. All DC Characteristics contained in this datasheet are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are preliminary values representing design targets, and will be updated after characterization of actual silicon.



Figure 27-4. Active Supply Current vs. V_{CC} (Internal RC Oscillator, 1 MHz)









27.12 Current Consumption of Peripheral Units





Figure 27-44. ADC Current vs. V_{CC} (ADC at 50 kHz)





	6.10Timer/Counter Oscillator	
	6.11System Clock Prescaler	
7	Power Management and Sleep Modes	
	7.1Idle Mode	
	7.2ADC Noise Reduction Mode	
	7.3Power-down Mode	
	7.4Power-save Mode	
	7.5Standby Mode	
	7.6Power Reduction Register	
	7.7Minimizing Power Consumption	41
8	System Control and Reset	43
	8.1 Internal Voltage Reference	48
	8.2Watchdog Timer	
9	Interrupts	54
	9.1Interrupt Vectors in ATmega48	54
	9.2Interrupt Vectors in ATmega88	
	9.3Interrupt Vectors in ATmega168	59
10	I/O-Ports	64
10	10.1Introduction	 64 64
10	10.1Introduction	
10	10.1Introduction	
10	I/O-Ports 10.1Introduction 10.2Ports as General Digital I/O 10.3Alternate Port Functions 10.4Register Description for I/O Ports	
10 11	I/O-Ports 10.1Introduction 10.2Ports as General Digital I/O 10.3Alternate Port Functions 10.4Register Description for I/O Ports External Interrupts	
10 11	I/O-Ports 10.1Introduction 10.2Ports as General Digital I/O 10.3Alternate Port Functions 10.4Register Description for I/O Ports External Interrupts 11.1Pin Change Interrupt Timing	
10 11 12	I/O-Ports 10.1Introduction 10.2Ports as General Digital I/O 10.3Alternate Port Functions 10.4Register Description for I/O Ports External Interrupts 11.1Pin Change Interrupt Timing 8-bit Timer/Counter0 with PWM	
10 11 12	I/O-Ports 10.1Introduction 10.2Ports as General Digital I/O 10.3Alternate Port Functions 10.4Register Description for I/O Ports External Interrupts 11.1Pin Change Interrupt Timing 8-bit Timer/Counter0 with PWM 12.1Overview	
10 11 12	I/O-Ports 10.1Introduction 10.2Ports as General Digital I/O 10.3Alternate Port Functions 10.4Register Description for I/O Ports External Interrupts 11.1Pin Change Interrupt Timing 8-bit Timer/Counter0 with PWM 12.1Overview 12.2Timer/Counter Clock Sources	
10 11 12	I/O-Ports 10.1Introduction 10.2Ports as General Digital I/O 10.3Alternate Port Functions 10.4Register Description for I/O Ports External Interrupts 11.1Pin Change Interrupt Timing 8-bit Timer/Counter0 with PWM 12.1Overview 12.2Timer/Counter Clock Sources 12.3Counter Unit	
10 11 12	 I/O-PORTS 10.1Introduction 10.2Ports as General Digital I/O 10.3Alternate Port Functions 10.4Register Description for I/O Ports I0.4Register Description for I/O Ports I1.1Pin Change Interrupts I1.1Pin Change Interrupt Timing 8-bit Timer/Counter0 with PWM 12.1Overview 12.2Timer/Counter Clock Sources 12.3Counter Unit 12.4Output Compare Unit 	
10 11 12	I/O-Ports 10.1Introduction 10.2Ports as General Digital I/O 10.3Alternate Port Functions 10.4Register Description for I/O Ports 10.4Register Description for I/O Ports 11.1Pin Change Interrupts 11.1Pin Change Interrupt Timing 8-bit Timer/Counter0 with PWM 12.1Overview 12.2Timer/Counter Clock Sources 12.3Counter Unit 12.4Output Compare Unit 12.5Compare Match Output Unit	
10 11 12	I/O-Ports 10.1Introduction 10.2Ports as General Digital I/O 10.3Alternate Port Functions 10.4Register Description for I/O Ports 10.4Register Description for I/O Ports 11.1Pin Change Interrupts 11.1Pin Change Interrupt Timing 8-bit Timer/Counter0 with PWM 12.1Overview 12.2Timer/Counter Clock Sources 12.3Counter Unit 12.4Output Compare Unit 12.5Compare Match Output Unit 12.6Modes of Operation	
10 11 12	I/O-Ports 10.1Introduction 10.2Ports as General Digital I/O 10.3Alternate Port Functions 10.4Register Description for I/O Ports 10.4Register Description for I/O Ports 11.1Pin Change Interrupts 11.1Pin Change Interrupt Timing 8-bit Timer/Counter0 with PWM 12.1Overview 12.2Timer/Counter Clock Sources 12.3Counter Unit 12.4Output Compare Unit 12.5Compare Match Output Unit 12.6Modes of Operation 12.7Timer/Counter Timing Diagrams	
10 11 12	I/O-Ports 10.1Introduction 10.2Ports as General Digital I/O 10.3Alternate Port Functions 10.4Register Description for I/O Ports 10.4Register Description for I/O Ports 11.1Pin Change Interrupts 11.1Pin Change Interrupt Timing 8-bit Timer/Counter0 with PWM 12.1Overview 12.2Timer/Counter Clock Sources 12.3Counter Unit 12.4Output Compare Unit 12.5Compare Match Output Unit 12.6Modes of Operation 12.7Timer/Counter Timing Diagrams 12.88-bit Timer/Counter Register Description	