Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	AVR
Core Size	8-Bit
Speed	4MHz
Connectivity	SPI
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	7
Program Memory Size	8KB (4K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 9V
Data Converters	A/D 5x12b
Oscillator Type	Internal
Operating Temperature	-20°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-TSSOP (0.465", 11.80mm Width)
Supplier Device Package	28-TSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega8hva-4tu

2.1 Comparison Between ATmega8HVA and ATmega16HVA

The ATmega8HVA and ATmega16HVA differ only in memory size and interrupt vector size. Table 2-1 summarizes the different configuration for the two devices.

Table 2-1. Configuration summary

Device	Flash	Interrupt vector size
ATmega8HVA	8K	1 Word
ATmega16HVA	16K	2 Word

3. Disclaimer

All Min, Typ and Max values contained in this datasheet are preliminary estimates based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Final values will be available after the device is characterized.

4. Resources

A comprehensive set of development tools, application notes and datasheets are available for download on http://www.atmel.com/avr.

5. Data Retention

Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C.

6. About Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C compiler documentation for more details.

For I/O registers located in extended I/O map, “IN”, “OUT”, “SBIS”, “SBIC”, “CBI”, and “SBI” instructions must be replaced with instructions that allow access to extended I/O. Typically “LDS” and “STS” combined with “SBRS”, “SBRC”, “SBR”, and “CBR”.

ical ALU operation, two operands are output from the Register File, the operation is executed, and the result is stored back in the Register File – in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing – enabling efficient address calculations. One of the these address pointers can also be used as an address pointer for look up tables in Flash program memory. These added function registers are the 16-bit X-, Y-, and Z-register, described later in this section.

The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed in the ALU. After an arithmetic operation, the Status Register is updated to reflect information about the result of the operation.

Program flow is provided by conditional and unconditional jump and call instructions, able to directly address the whole address space. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the Stack. The Stack is effectively allocated in the general data SRAM, and consequently the Stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the Reset routine (before subroutines or interrupts are executed). The Stack Pointer (SP) is read/write accessible in the I/O space. The data SRAM can easily be accessed through the five different addressing modes supported in the AVR architecture.

The memory spaces in the AVR architecture are all linear and regular memory maps.

A flexible interrupt module has its control registers in the I/O space with an additional Global Interrupt Enable bit in the Status Register. All interrupts have a separate Interrupt Vector in the Interrupt Vector table. The interrupts have priority in accordance with their Interrupt Vector position. The lower the Interrupt Vector address, the higher the priority.

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, SPI, and other I/O functions. The I/O Memory can be accessed directly, or as the Data Space locations following those of the Register File, 0x20 - 0x5F. In addition, the ATmega8HVA/16HVA has Extended I/O space from 0x60 - 0xFF in SRAM where only the ST/STS/STD and LD/LDS/LDD instructions can be used.

7.2 ALU – Arithmetic Logic Unit

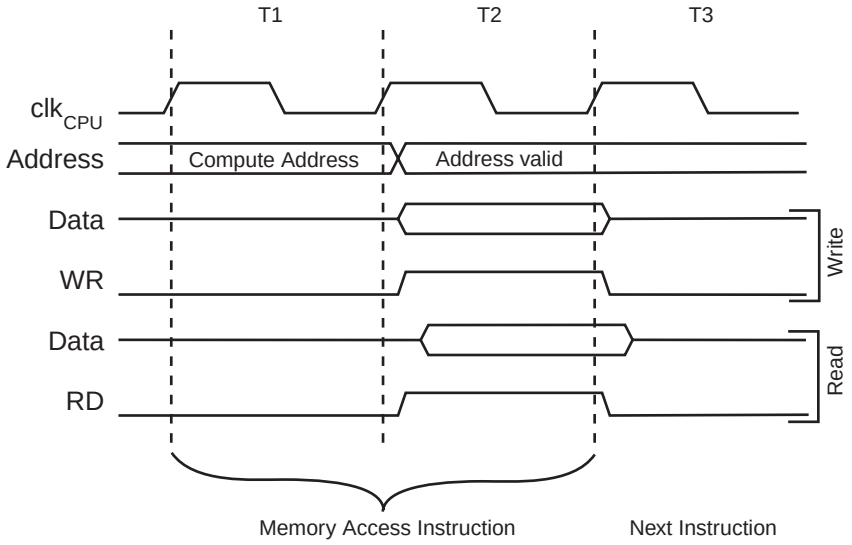
The high-performance AVR ALU operates in direct connection with all the 32 general purpose working registers. Within a single clock cycle, arithmetic operations between general purpose registers or between a register and an immediate are executed. The ALU operations are divided into three main categories – arithmetic, logical, and bit-functions. Some implementations of the architecture also provide a powerful multiplier supporting both signed/unsigned multiplication and fractional format. See the “Instruction Set” section for a detailed description.

7.3 Status Register

The Status Register contains information about the result of the most recently executed arithmetic instruction. This information can be used for altering program flow in order to perform conditional operations. Note that the Status Register is updated after all ALU operations, as specified in the Instruction Set Reference. This will in many cases remove the need for using the dedicated compare instructions, resulting in faster and more compact code.

The Status Register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt. This must be handled by software.

Figure 8-3. On-chip Data SRAM Access Cycles



8.4 EEPROM Data Memory

The ATmega8HVA/16HVA contains 256 bytes of data EEPROM memory. It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU is described in the following, specifying the EEPROM Address Registers, the EEPROM Data Register, and the EEPROM Control Register.

For a detailed description of EEPROM programming, see page 151 and page 156 respectively.

8.4.1 EEPROM Read/Write Access

The EEPROM Access Registers are accessible in the I/O space.

The write access time for the EEPROM is given in Table 8-1 on page 20. A self-timing function, however, lets the user software detect when the next byte can be written. If the user code contains instructions that write the EEPROM, some precautions must be taken.

In order to prevent unintentional EEPROM writes, a specific write procedure must be followed. Refer to the description of the EEPROM Control Register for details on this.

When the EEPROM is read, the CPU is halted for four clock cycles before the next instruction is executed. When the EEPROM is written, the CPU is halted for two clock cycles before the next instruction is executed.

8.5 I/O Memory

The I/O space definition of the ATmega8HVA/16HVA is shown in "Register Summary" on page 175.

All ATmega8HVA/16HVA I/Os and peripherals are placed in the I/O space. All I/O locations may be accessed by the LD/LDS/LDD and ST/STS/STD instructions, transferring data between the 32 general purpose working registers and the I/O space. I/O Registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions. Refer to the instruction set section for more details. When using the I/O specific commands IN and OUT, the

9.1.5 Coulomb Counter ADC Clock - clk_{CCADC}

The Coulomb Counter ADC is provided with a dedicated clock domain. This allows operating the Coulomb Counter ADC in low power modes like Power-save for continuous current measurements.

9.1.6 Watchdog Timer and Battery Protection Clock

The Watchdog Timer and Battery Protection are provided with a dedicated clock domain. This allows operation in all modes except Power-off. It also allows very low power operation by utilizing an Ultra Low Power RC Oscillator dedicated to this purpose.

9.2 Clock Sources

The following section describe the clock sources available in the device. The clocks are input to the AVR clock generator, and routed to the appropriate modules.

9.3 Calibrated Fast RC Oscillator

The calibrated Fast RC Oscillator by default provides a 8.0 MHz clock to the system clock prescaler. The frequency is nominal value at 70°C. This clock will operate with no external components. With an accurate time reference and by using runtime calibration, this oscillator can be calibrated to an accuracy of +/- 1. % over the entire temperature range. During reset, hardware loads the calibration byte into the FOSCCAL Register and thereby automatically calibrates the Fast RC Oscillator. At 70°C, this calibration gives a frequency of 8 MHz ± 4%. The oscillator can be calibrated to any frequency in the range 7.3- 8.1 MHz by changing the FOSCCAL register. For more information on the pre-programmed calibration value, see the section ”Reading the Signature Row from Software” on page 144. Note that the frequency of the system clock is given by the ”System Clock Prescaler” on page 27.

Start-up time of the chip is referred to as a number of Prescaled Fast RC Oscillator cycles (CK) and a number of ULP oscillator cycles. The start-up time is selected by SUT fuses as defined in Table 9-1 on page 25.

Table 9-1. Start-up times according to SUT fuse selection.

SUT3..0	Start-up Time from Power-save	Additional Delay from Reset, Typical Values[2]
000	6 CK	14 CK + 4 ms
001	6 CK	14 CK + 8 ms
010	6 CK	14 CK + 16 ms
011	6 CK	14 CK + 32 ms
100	6 CK	14 CK + 64 ms
101	6 CK	14 CK + 128 ms
110	6 CK	14 CK + 256 ms
111[1]	6 CK	14 CK + 512 ms

Notes: 1. The device is shipped with this option selected.
2. The actual value of the added, selectable 4- 512 ms delay depends on the actual frequency of the “Ultra Low Power RC Oscillator”. See Table 9-2 on page 27 and ”Electrical Characteristics” on page 165

10. Power Management and Sleep Modes

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

10.1 Sleep Modes

Figure 9-1 on page 24 presents the different clock systems in the ATmega8HVA/16HVA, and their distribution. The figure is helpful in selecting an appropriate sleep mode. The different sleep modes and their wake up sources is summarized in Table 10-1, and Figure 10-1 on page 35 shows a sleep mode state diagram.

Table 10-1. Wake-up Sources for Sleep Modes

Mode	Wake-up sources									
	Wake-up on Regular Current	Battery Protection Interrupts	External Interrupts	WDT	EEPROM Ready	VREGMON	CC-ADC	V-ADC	Other I/O	Charger Detect(1)
Idle	X	X	X	X	X	X	X(2)	X	X	
ADC Noise Reduction	X	X	X	X	X	X	X(2)	X		
Power-save	X	X	X	X			X(2)			
Power-off										X

Notes: 1. Discharge FET must be switched off for Charger Detect to be active.
2. Instantaneous and Accumulate Conversion Complete wake-up only.

To enter any of the sleep modes, the SE bit in SMCR, see "SMCR – Sleep Mode Control Register" on page 39, must be written to logic one and a SLEEP instruction must be executed. The SM2..0 bits in the SMCR Register select which sleep mode will be activated by the SLEEP instruction. See Table 10-3 on page 39 for a summary.

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 any sleep mode except Power-off. If a reset occurs during sleep mode, the MCU wakes up and executes from the Reset Vector.

Table 10-2. Active modules in different Sleep Modes (Continued)

Module	Mode				
	Active	Idle	ADC Noise Reduction	Power-save	Power-off
WDT	X	X	X	X	
VREG	X	X	X	X	
CHARGER_DETECT(1)	X	X	X	X	X
VREGMON	X	X	X		
OSI	X	X			

Notes:
1. Discharge FET must be switched off for Charger Detect to be enabled.
2. RCOSC_FAST runs in Power-save mode if DUVR mode is enabled. It also runs for approximately 125 ms after C-FET/D-FET has been enabled.
3. RCOSC_SLOW only runs if CC-ADC is enabled or when the oscillator is selected as input to the Oscillator sampling Interface and sampling is enabled.
4. Instantaneous and Accumulate Conversion Complete wake-up only.

10.2 Idle Mode

When the SM2..0 bits are written to 000, the SLEEP instruction makes the MCU enter Idle mode, stopping the CPU but allowing all peripheral functions to continue operating. This sleep mode basically halts clk_{CPU} and clk_{FLASH}, while allowing the other clocks to run. Idle mode enables the MCU to wake up from external triggered interrupts as well as internal ones like the Timer Overflow interrupt.

10.3 ADC Noise Reduction

When the SM2:0 bits are written to 001, the SLEEP instruction makes the MCU enter ADC Noise Reduction mode, stopping the CPU but allowing the Voltage ADC (V-ADC), Watchdog Timer (WDT), Coulomb Counter (CC), Current Battery Protection (CBP), and the Ultra Low Power RC Oscillator (RCOSC_ULP) to continue operating. This sleep mode basically halts $clk_{I/O}$, clk_{CPU}, and clk_{FLASH}, while allowing the other clocks to run.

This improves the noise environment for the Voltage ADC, enabling higher resolution measurements.

10.4 Power-save Mode

When the SM2..0 bits are written to 011, the SLEEP instruction makes the MCU enter Power-save mode. In this mode, the internal Fast RC Oscillator (RCOSC_FAST) is stopped, while Watchdog Timer (WDT), Coulomb Counter (CC), Current Battery Protection (CBP), the Ultra Low Power RC Oscillator (RCOSC_ULP), and the Slow RC oscillator (RCOSC_SLOW) continue operating.

This mode will be the default mode when application software does not require operation of CPU, Flash or any of the peripheral units running at the Fast internal Oscillator (RCOSC_FAST).

If the current through the sense resistor is so small that the Coulomb Counter cannot measure it accurately, Regular Current detection should be enabled to reduce power consumption. The WDT keeps accurately track of the time so that battery self discharge can be calculated.

- **Bit 3 - PRSPI: Power Reduction Serial Peripheral Interface**

Writing logic one to this bit shuts down the Serial Peripheral Interface by stopping the clock to the module. When waking up the SPI again, the SPI should be reinitialized to ensure proper operation.

- **Bit 2 - PRTIM1: Power Reduction Timer/Counter1**

Writing a logic one to this bit shuts down the Timer/Counter1 module. When the Timer/Counter1 is enabled, operation will continue like before the shutdown.

- **Bit 1 - PRTIM0: Power Reduction Timer/Counter0**

Writing a logic one to this bit shuts down the Timer/Counter0 module. When the Timer/Counter0 is enabled, operation will continue like before the shutdown.

- **Bit 0 - PRVADC: Power Reduction V-ADC**

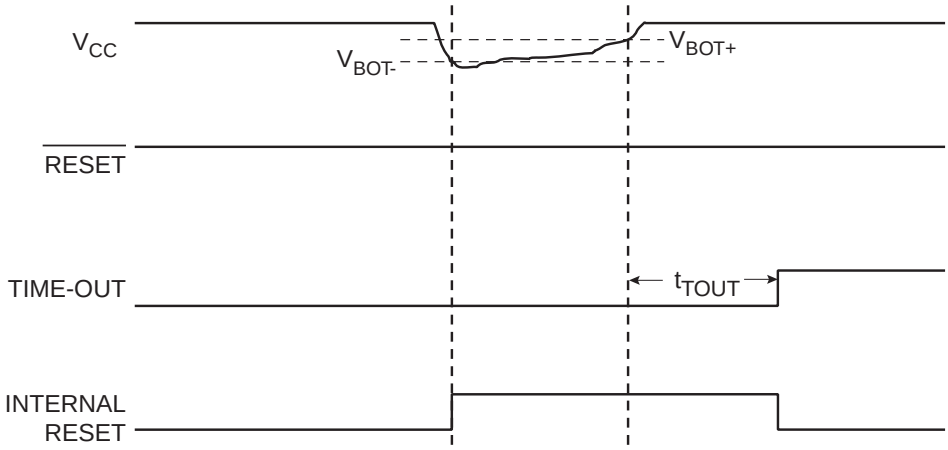
Writing a logic one to this bit shuts down the V-ADC. Before writing the PRVADC bit, make sure that the VADEN bit is cleared to minimize the power consumption.

Note: V-ADC control registers can be updated even if the PRVADC bit is set.

The BOD is automatically enabled in all modes of operation, except in Power-off mode.

When the BOD is enabled, and V_{REG} decreases to a value below the trigger level (V_{BOT-} in Figure 11-5), the Brown-out Reset is immediately activated. When V_{REG} increases above the trigger level (V_{BOT+} in Figure 11-5), the delay counter starts the MCU after the Time-out period t_{TOUT} has expired.

Figure 11-5. Brown-out Reset During Operation



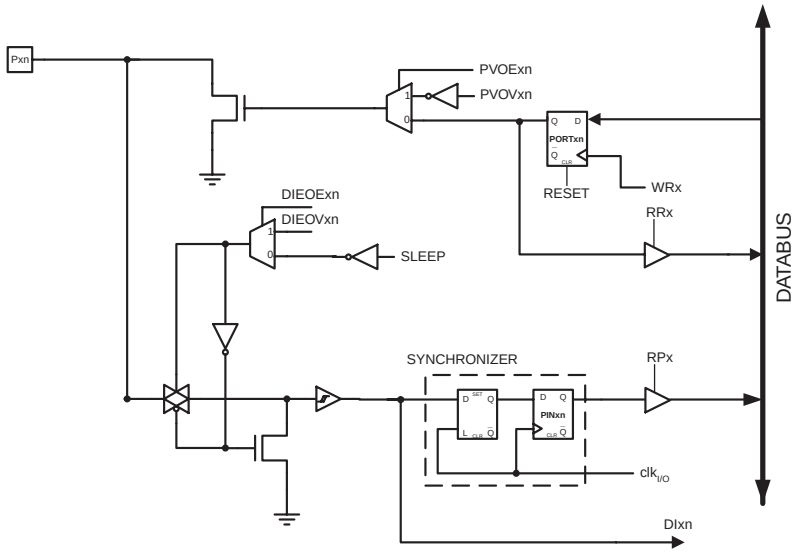
11.2.5 Black-out Detection

As an extra security feature, the chip will automatically enter Power-off if V_{REG} drops below V_{BLOT}. V_{BLOT} will always be well below the BOD level, V_{BOT}.

14.4 Alternate Port Functions

The High Voltage I/O has alternate port functions in addition to being general digital I/O. Figure 14-3 shows how the port pin control signals from the simplified Figure 14-2 on page 59 can be overridden by alternate functions.

Figure 14-3. High Voltage Digital I/O[1]



PVOExn: Pxn PORT VALUE OVERRIDE ENABLE
PVOVxn: Pxn PORT VALUE OVERRIDE VALUE
DIEOExn: Pxn DIGITAL INPUT-ENABLE OVERRIDE ENABLE
DIEOVxn: Pxn DIGITAL INPUT-ENABLE OVERRIDE VALUE

RRx: READ PORTx REGISTER
WRx: WRITE PORTx REGISTER
RPx: READ PINx REGISTER
$clk_{I/O}$: I/O CLOCK
DIxn: DIGITAL INPUT PIN n ON PORTx
SLEEP: SLEEP CONTROL

Note: 1. WRx, RRx and RPx are common to all pins within the same port. $clk_{I/O}$ and SLEEP are common to all ports. All other signals are unique for each pin.

Table 14-1 on page 61 summarizes the function of the overriding signals. The pin and port indexes from Figure 14-3 are not shown in the succeeding tables. The overriding signals are generated internally in the modules having the alternate function.

17.2.1 Registers

The Timer/Counter Low Byte Register (TCNTnL) and Output Compare Registers (OCRnA and OCRnB) are 8-bit registers. Interrupt request (abbreviated to Int.Req. in Figure 17-1 on page 77) signals are all visible in the Timer Interrupt Flag Register (TIFRn). All interrupts are individually masked with the Timer Interrupt Mask Register (TIMSKn). TIFRn and TIMSKn are not shown in the figure.

In 16-bit mode the Timer/Counter consists one more 8-bit register, the Timer/Counter High Byte Register (TCNTnH). Furthermore, there is only one Output Compare Unit in 16-bit mode as the two Output Compare Registers, OCRnA and OCRnB, are combined to one 16-bit Output Compare Register. OCRnA contains the low byte of the word and OCRnB contains the higher byte of the word. When accessing 16-bit registers, special procedures described in section "Accessing Registers in 16-bit Mode" on page 86 must be followed.

The Timer/Counter can be clocked internally, via the prescaler, or by an external clock source on the Tn pin. The Clock Select logic block controls which clock source and edge the Timer/Counter uses to increment its value. The Timer/Counter is inactive when no clock source is selected. The output from the Clock Select logic is referred to as the timer clock (clk_{Tn}).

17.2.2 Definitions

Many register and bit references in this section are written in general form. A lower case "n" replaces the module number, e.g. Timer/Counter number. A lower case "x" replaces the unit, e.g. OCRnx and ICPnx describes OCRnB/A and ICP1/0x . However, when using the register or bit defines in a program, the precise form must be used, i.e., TCNT0L for accessing Timer/Counter0 counter value and so on.

The definitions in Table 17-1 are also used extensively throughout the document.

Table 17-1. Definitions

BOTTOM	The counter reaches the BOTTOM when it becomes 0.
MAX	The counter reaches its MAXimum when it becomes 0xFF (decimal 255) in 8-bit mode or 0xFFFF (decimal 65535) in 16-bit mode.
TOP	The counter reaches the TOP when it becomes equal to the highest value in the count sequence. The TOP value can be assigned to be the fixed value 0xFF/0xFFFF (MAX) or the value stored in the OCRnA Register.

17.3 Timer/Counter Clock Sources

The Timer/Counter can be clocked internally, via the prescaler, or by an external clock source. The Clock Select logic is controlled by the Clock Select (CSn2:0) bits located in the Timer/Counter Control Register n B (TCCRnB), and controls which clock source and edge the Timer/Counter uses to increment its value. The Timer/Counter is inactive when no clock source is selected. The output from the Clock Select logic is referred to as the timer clock (clk_{Tn}). For details on clock sources and prescaler, see "Timer/Counter0 and Timer/Counter1 Prescalers" on page 74

17.5.5 8-bit Input Capture Mode

The Timer/Counter can be used in a 8-bit Input Capture mode, see Table 17-2 on page 80 for bit settings. For full description, see "Input Capture Unit" on page 82.

17.5.6 16-bit Input Capture Mode

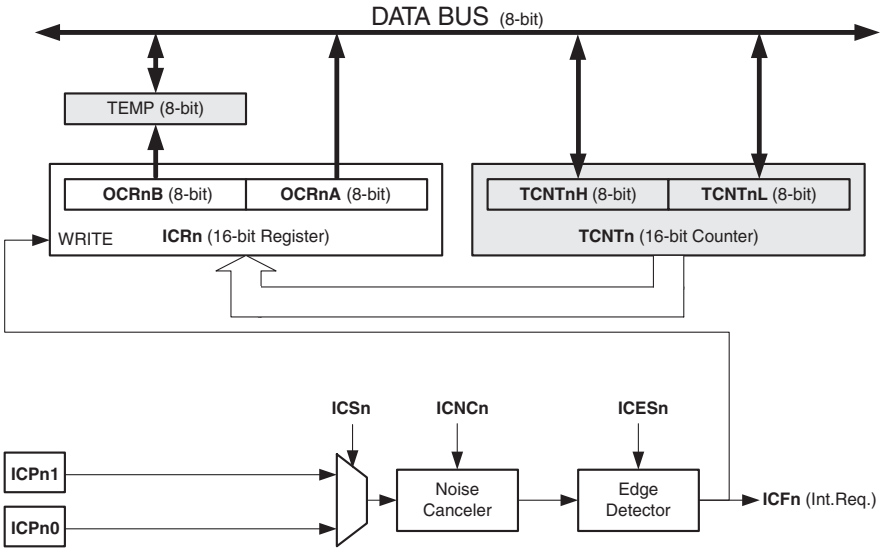
The Timer/Counter can also be used in a 16-bit Input Capture mode, see Table 17-2 on page 80 for bit settings. For full description, see "Input Capture Unit" on page 82.

17.6 Input Capture Unit

The Timer/Counter incorporates an Input Capture unit that can capture external events and give them a time-stamp indicating time of occurrence. The external signal indicates an event, or multiple events. For Timer/Counter0, the events can be applied via the PC0 pin (ICP01), or alternatively via the osi_posedge pin on the Oscillator Sampling Interface (ICP00). For Timer/Counter1, the events can be applied by the Battery Protection Interrupt (ICP10) or alternatively by the Voltage Regulator Interrupt (ICP11). The time-stamps can then be used to calculate frequency, duty-cycle, and other features of the signal applied. Alternatively the time-stamps can be used for creating a log of the events.

The Input Capture unit is illustrated by the block diagram shown in Figure 17-4 on page 82. The elements of the block diagram that are not directly a part of the Input Capture unit are gray shaded.

Figure 17-4. Input Capture Unit Block Diagram



The Output Compare Register OCRnA is a dual-purpose register that is also used as an 8-bit Input Capture Register ICRn. In 16-bit Input Capture mode the Output Compare Register OCRnB serves as the high byte of the Input Capture Register ICRn. In 8-bit Input Capture mode the Output Compare Register OCRnB is free to be used as a normal Output Compare Register, but in 16-bit Input Capture mode the Output Compare Unit cannot be used as there are no free Output Compare Register(s). Even though the Input Capture register is called ICRn in this section, it is referring to the Output Compare Register(s). For more information on how to access the 16-bit registers refer to "Accessing Registers in 16-bit Mode" on page 86.

When a change of the logic level (an event) occurs on the *Input Capture pin* (ICPx), and this change confirms to the setting of the edge detector, a capture will be triggered. When a capture is triggered, the value of the counter (TCNTn) is written to the *Input Capture Register* (ICRn). The *Input Capture Flag* (ICFn) is set at the same system clock as the TCNTn value is copied into Input Capture Register. If enabled (TICIEn=1), the Input Capture Flag generates an Input Capture interrupt. The ICFn flag is automatically cleared when the interrupt is executed. Alternatively the ICFn flag can be cleared by software by writing a logical one to its I/O bit location.

17.6.1 Input Capture Trigger Source

The default trigger source for the Input Capture unit is the I/O port PC0 in Timer/Counter0 and the Battery Protection Interrupt in Timer/Counter1. Alternatively can the osi_posedge pin on the Oscillator Sampling Interface in Timer/Counter0 and Voltage Regulator Interrupt in Timer/Counter1 be used as trigger sources. The osi_posedge pin in Timer/Counter0 Control Register A (TCCR0A) and the Voltage Regulator Interrupt bit in the Timer/Counter1 Control Register A (TCCR1A) is selected as trigger sources by setting the Input Capture Select bits respectively to 00 and 11. Be aware that changing trigger source can trigger a capture. The Input Capture Flag must therefore be cleared after the change.

Both Input Capture inputs are sampled using the same technique. The edge detector is also identical. However, when the noise canceler is enabled, additional logic is inserted before the edge detector, which increases the delay by four system clock cycles. An Input Capture on Timer/Counter0 can also be triggered by software by controlling the port of the PC0 pin.

17.6.2 Noise Canceler

The noise canceler improves noise immunity by using a simple digital filtering scheme. The noise canceler input is monitored over four samples, and all four must be equal for changing the output that in turn is used by the edge detector.

The noise canceler is enabled by setting the Input Capture Noise Canceler (ICNCn) bit in *Timer/Counter Control Register n B* (TCCRnB). When enabled the noise canceler introduces additional four system clock cycles of delay from a change applied to the input, to the update of the ICRn Register. The noise canceler uses the system clock and is therefore not affected by the prescaler.

17.6.3 Using the Input Capture Unit

The main challenge when using the Input Capture unit is to assign enough processor capacity for handling the incoming events. The time between two events is critical. If the processor has not read the captured value in the ICRn Register before the next event occurs, the ICRn will be overwritten with a new value. In this case the result of the capture will be incorrect.

When using the Input Capture interrupt, the ICRn Register should be read as early in the interrupt handler routine as possible. The maximum interrupt response time is dependent on the maximum number of clock cycles it takes to handle any of the other interrupt requests.

Measurement of an external signal duty cycle requires that the trigger edge is changed after each capture. Changing the edge sensing must be done as early as possible after the ICRn Register has been read. After a change of the edge, the Input Capture Flag (ICFn) must be

The following code examples show how to do an atomic read of the TCNTn register contents. Reading any of the OCRn register can be done by using the same principle.

Assembly Code Example

```
TIMn_ReadTCNTn:
  ; Save global interrupt flag
  in  r18,SREG
  ; Disable interrupts
  cli
  ; Read TCNTn into r17:r16
  in  r16,TCNTnL
  in  r17,TCNTnH
  ; Restore global interrupt flag
  out SREG,r18
  ret
```

C Code Example

```
unsigned int TIMn_ReadTCNTn( void )
{
  unsigned char sreg;
  unsigned int i;
  /* Save global interrupt flag */
  sreg = SREG;
  /* Disable interrupts */
  _CLI();
  /* Read TCNTn into i */
  i = TCNTn;
  /* Restore global interrupt flag */
  SREG = sreg;
  return i;
}
```

Note: 1. See “About Code Examples” on page 7.

The assembly code example returns the TCNTnH/L value in the r17:r16 register pair.

- **Bits 1, 0 – SPR1, SPR0: SPI Clock Rate Select 1 and 0**

These two bits control the SCK rate of the device configured as a Master. SPR1 and SPR0 have no effect on the Slave. The relationship between SCK and the Oscillator Clock frequency f_{osc} is shown in the following table:

Table 18-5. Relationship Between SCK and the Oscillator Frequency

SPI2X	SPR1	SPR0	SCK Frequency
0	0	0	$f_{osc}/4$
0	0	1	$f_{osc}/16$
0	1	0	$f_{osc}/64$
0	1	1	$f_{osc}/128$
1	0	0	$f_{osc}/2$
1	0	1	$f_{osc}/8$
1	1	0	$f_{osc}/32$
1	1	1	$f_{osc}/64$

18.5.2 SPSR – SPI Status Register

Bit	7	6	5	4	3	2	1	0	
0x2D (0x4D)	SPIF	WCOL	–	–	–	–	–	SPI2X	SPSR
Read/Write	R	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – SPIF: SPI Interrupt Flag**

When a serial transfer is complete, the SPIF Flag is set. An interrupt is generated if SPIE in SPCR is set and global interrupts are enabled. If $\overline{SS}$ is an input and is driven low when the SPI is in Master mode, this will also set the SPIF Flag. SPIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, the SPIF bit is cleared by first reading the SPI Status Register with SPIF set, then accessing the SPI Data Register (SPDR).

- **Bit 6 – WCOL: Write Collision Flag**

The WCOL bit is set if the SPI Data Register (SPDR) is written during a data transfer. The WCOL bit (and the SPIF bit) are cleared by first reading the SPI Status Register with WCOL set, and then accessing the SPI Data Register.

- **Bit 5:1 – Res: Reserved Bits**

These bits are reserved bits in the ATmega8HVA/16HVA and will always read as zero.

- **Bit 0 – SPI2X: Double SPI Speed Bit**

When this bit is written logic one the SPI speed (SCK Frequency) will be doubled when the SPI is in Master mode (see Table 18-5 on page 102). This means that the minimum SCK period will be two CPU clock periods. When the SPI is configured as Slave, the SPI is only guaranteed to work at $f_{osc}/4$ or lower.

The SPI interface on the ATmega8HVA/16HVA is also used for program memory and EEPROM downloading or uploading. See Table 27.6 on page 151 for serial programming and verification.

$$T(K) = \frac{VADCH/L \cdot VPTAT\ CAL}{16384}$$

20.4.4 DIDR0 – Digital Input Disable Register 0

Bit	7	6	5	4	3	2	1	0	
(0x7E)	–	–	–	–	–	–	PA1DID	PA0DID	DIDR0
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:2 – Res: Reserved Bits**

These bits are reserved for future use. To ensure compatibility with future devices, these bits must be written to zero when DIDR0 is written.

- **Bit 1:0 – PA1DID:PA0DID: Digital Input Disable**

When this bit is written logic one, the digital input buffer on the corresponding Port A pin is disabled. The corresponding PIN Register bit will always read as zero when this bit is set. When an analog signal is applied to the PA1:0 pin and the digital input from this pin is not needed, this bit should be written logic one to reduce power consumption in the digital input buffer.

23. Battery Protection

23.1 Features

- **Short-circuit Protection**
- **Discharge Over-current Protection**
- **Charge Over-current Protection**
- **Discharge High-current Protection**
- **Charge High-current Protection**
- **Programmable and Lockable Detection Levels and Reaction Times**
- **Autonomous Operation Independent of CPU**

23.2 Overview

The Current Battery Protection circuitry (CBP) monitors the charge and discharge current and disables C-FET and D-FET if a Short-circuit, Over-current or High-current condition is detected. There are five different programmable detection levels: Short-circuit Detection Level, Discharge Over-current Detection Level, Charge Over-current Detection Level, Discharge High-current Detection Level, Charge High-current Detection Level. There are three different programmable delays for activating Current Battery Protection: Short-circuit Reaction Time, Over-current Reaction Time and High-current Reaction Time. After Current Battery Protection has been activated, the application software must re-enable the FETs. The Battery Protection hardware provides a hold-off time of 1 second before software can re-enable the discharge FET. This provides safety in case the application software should unintentionally re-enable the discharge FET too early.

The activation of a protection also issues an interrupt to the CPU. The battery protection interrupts can be individually enabled and disabled by the CPU.

The effect of the various battery protection types is given in Table 23-1.

Table 23-1. Effect of Battery Protection Types

Battery Protection Type	Interrupt Requests	C-FET	D-FET	MCU
Short-circuit Protection	Entry	Disabled	Disabled	Operational
Discharge Over-current Protection	Entry	Disabled	Disabled	Operational
Charge Over-current Protection	Entry	Disabled	Disabled	Operational
Discharge High-current Protection	Entry	Disabled	Disabled	Operational
Charge High-current Protection	Entry	Disabled	Disabled	Operational

In order to reduce power consumption, Short-circuit, Discharge High-current and Discharge Over-current Protection are automatically deactivated when the D-FET is disabled. The Charge Over-current and Charge High-current Protection are disabled when the C-FET is disabled. Note however that Charge Over-current Protection and Charge High-current Protection are never automatically disabled when the chip is operated in DUVR mode.

23.9.6 BPSCD – Battery Protection Short-circuit Detection Level Register

Bit	7	6	5	4	3	2	1	0	
(0xF5)	SCDL[7:0]								BPSCD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	1	1	1	0	0	1	1	

- **Bits 7:0 – SCDL7:0: Short-circuit Detection Level**

These bits sets the R_{SENSE} voltage level for detection of Short-circuit in the discharge direction, as defined in Table 23-5 on page 132.

Note: Due to synchronization of parameters between clock domains, a guard time of 3 ULP oscillator cycles + 3 CPU clock cycles is required between each time the BPSCD register is written. Any writing to the BPSCD register during this period will be ignored.

23.9.7 BPDOCD – Battery Protection Discharge-Over-current Detection Level Register

Bit	7	6	5	4	3	2	1	0	
(0xF6)	DOCDL[7:0]								BPDOCD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	1	1	1	0	0	1	1	

- **Bits 7:0 – DOCDL7:0: Discharge Over-current Detection Level**

These bits sets the R_{SENSE} voltage level for detection of Discharge Over-current, as defined in Table 23-5 on page 132.

Note: Due to synchronization of parameters between clock domains, a guard time of 3 ULP oscillator cycles + 3 CPU clock cycles is required between each time the BPDOCD register is written. Any writing to the BPDOCD register during this period will be ignored.

23.9.8 BPCOCD – Battery Protection Charge-Over-current Detection Level Register

Bit	7	6	5	4	3	2	1	0	
(0xF7)	COCDL[7:0]								BPCOCD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	1	1	1	0	0	1	1	

- **Bits 7:0 –COCDL7:0: Charge Over-current Detection Level**

These bits sets the R_{SENSE} voltage level for detection of Charge Over-current, as defined in Table 23-5 on page 132.

Note: Due to synchronization of parameters between clock domains, a guard time of 3 ULP oscillator cycles + 3 CPU clock cycles is required between each time the BPCOCD register is written. Any writing to the BPCOCD register during this period will be ignored.

- **Bit 0 - CHCIE : Charger High-current Protection Activated Interrupt**

The CHCIE bit enables interrupt caused by the Charge High-current Protection Activated Interrupt.

23.9.12 BPIFR – Battery Protection Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
(0xF3)	-	-	-	SCIF	DOCIF	COCIF	DHCIF	CHCIF	BPIFR
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:5 – Res: Reserved Bit**

These bits are reserved and will always read as zero.

- **Bit 4 – SCIF: Short-circuit Protection Activated Interrupt**

Once Short-circuit violation is detected, SCIF becomes set. The flag is cleared by writing a logic one to it.

- **Bit 3 – DOCIF: Discharge Over-current Protection Activated Interrupt**

Once Disharge Over-current violation is detected, DOCIF becomes set. The flag is cleared by writing a logic one to it.

- **Bit 2 – COCIF: Charge Over-current Protection Activated Interrupt**

Once Charge Over-current violation is detected, COCIF becomes set. The flag is cleared by writing a logic one to it.

- **Bit 1 – DHCIF: Disharge High-current Protection Activated Interrupt**

Once Discharge High-current violation is detected, DHCIF becomes set. The flag is cleared by writing a logic one to it.

- **Bit 0 – CHCIF: Charge High-current Protection Activated Interrupt**

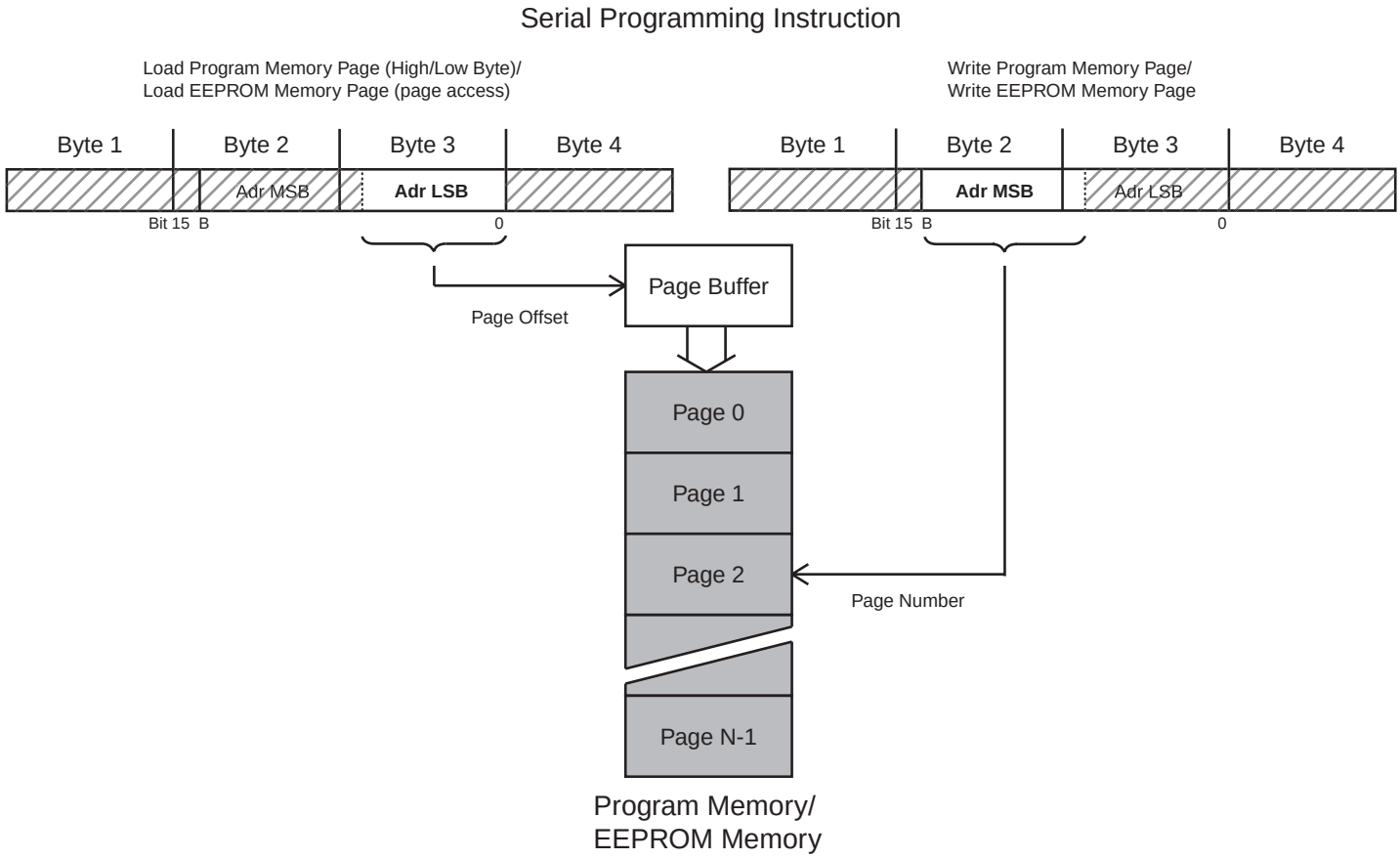
Once Charge High-current violation is detected, CHCIF becomes set. The flag is cleared by writing a logic one to it.

If the LSB in RDY/BSY data byte out is ‘1’, a programming operation is still pending. Wait until this bit returns ‘0’ before the next instruction is carried out.

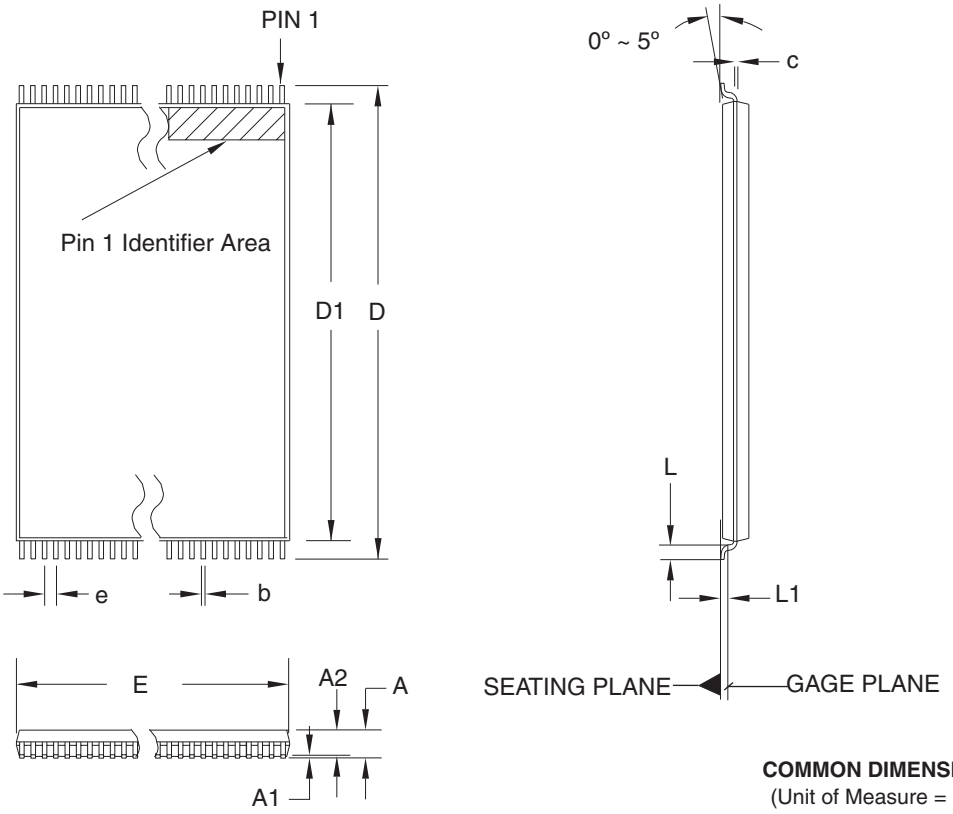
Within the same page, the low data byte must be loaded prior to the high data byte.

After data is loaded to the page buffer, program the EEPROM page, see Figure 27-2 on page 155.

Figure 27-2. Serial Programming Instruction example



34.2 28T



COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.20	
A1	0.05	–	0.15	
A2	0.90	1.00	1.05	
D	13.20	13.40	13.60	
D1	11.70	11.80	11.90	Note 2
E	7.90	8.00	8.10	Note 2
L	0.50	0.60	0.70	
L1	0.25 BASIC			
b	0.17	0.22	0.27	
c	0.10	–	0.21	
e	0.55 BASIC			

Notes:
1. This package conforms to JEDEC reference MO-183.
2. Dimensions D1 and E do not include mold protrusion. Allowable protrusion on E is 0.15 mm per side and on D1 is 0.25 mm per side.
3. Lead coplanarity is 0.10 mm maximum.

12/06/02

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**28T**, 28-lead (8 x 13.4 mm) Plastic Thin Small Outline Package, Type I (TSOP)	28T	C