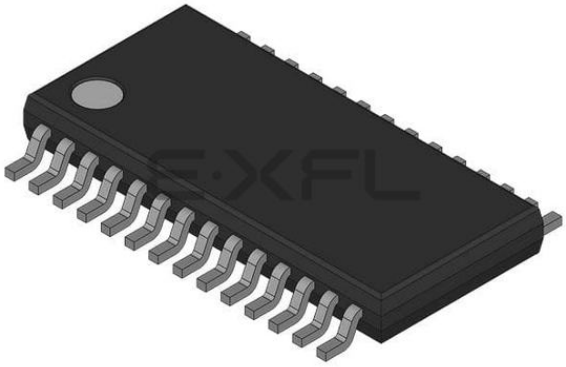


Welcome to E-XFL.COM

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	16KB (8K 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/atmel/atmega16hva-4tu

- **Bits 7:6 – Res: Reserved Bits**

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

- **Bits 5, 4 – EEPM1 and EEPM0: EEPROM Programming Mode Bits**

The EEPROM Programming mode bit setting defines which programming action that will be triggered when writing EEPE. It is possible to program data in one atomic operation (erase the old value and program the new value) or to split the Erase and Write operations in two different operations. The Programming times for the different modes are shown in Table 8-1. While EEPE is set, any write to EEPMn will be ignored. During reset, the EEPMn bits will be reset to 0b00 unless the EEPROM is busy programming.

Table 8-1. EEPROM Mode Bits

EEPM1	EEPM0	Typ Programming Time, f_{osc} = 4.0 MHz	Operation
0	0	3.4 ms	Erase and Write in one operation (Atomic Operation)
0	1	1.8 ms	Erase Only
1	0	1.8 ms	Write Only
1	1	–	Reserved for future use

- **Bit 3 – EERIE: EEPROM Ready Interrupt Enable**

Writing EERIE to one enables the EEPROM Ready Interrupt if the I bit in SREG is set. Writing EERIE to zero disables the interrupt. The EEPROM Ready interrupt generates a constant interrupt when EEPE is cleared.

- **Bit 2 – EEMPE: EEPROM Master Write Enable**

The EEMPE bit determines whether setting EEPE to one causes the EEPROM to be written. When EEMPE is set, setting EEPE within four clock cycles will write data to the EEPROM at the selected address If EEMPE is zero, setting EEPE will have no effect. When EEMPE has been written to one by software, hardware clears the bit to zero after four clock cycles. See the description of the EEPE bit for an EEPROM write procedure.

- **Bit 1 – EEPE: EEPROM Write Enable**

The EEPROM Write Enable Signal EEPE is the write strobe to the EEPROM. When address and data are correctly set up, the EEPE bit must be written to one to write the value into the EEPROM. The EEMPE bit must be written to one before a logical one is written to EEPE, otherwise no EEPROM write takes place. The following procedure should be followed when writing the EEPROM (the order of steps 2 and 3 is not essential):

1. Wait until EEPE becomes zero.
2. Write new EEPROM address to EEAR (optional).
3. Write new EEPROM data to EEDR (optional).
4. Write a logical one to the EEMPE bit while writing a zero to EEPE in EECR.
5. Within four clock cycles after setting EEMPE, write a logical one to EEPE.

Caution:

An interrupt between step 4 and step 5 will make the write cycle fail, since the EEPROM Master Write Enable will time-out. If an interrupt routine accessing the EEPROM is interrupting another EEPROM access, the EEAR or EEDR Register will be modified, causing the interrupted

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.

Figure 10-1. Sleep Mode State Diagram

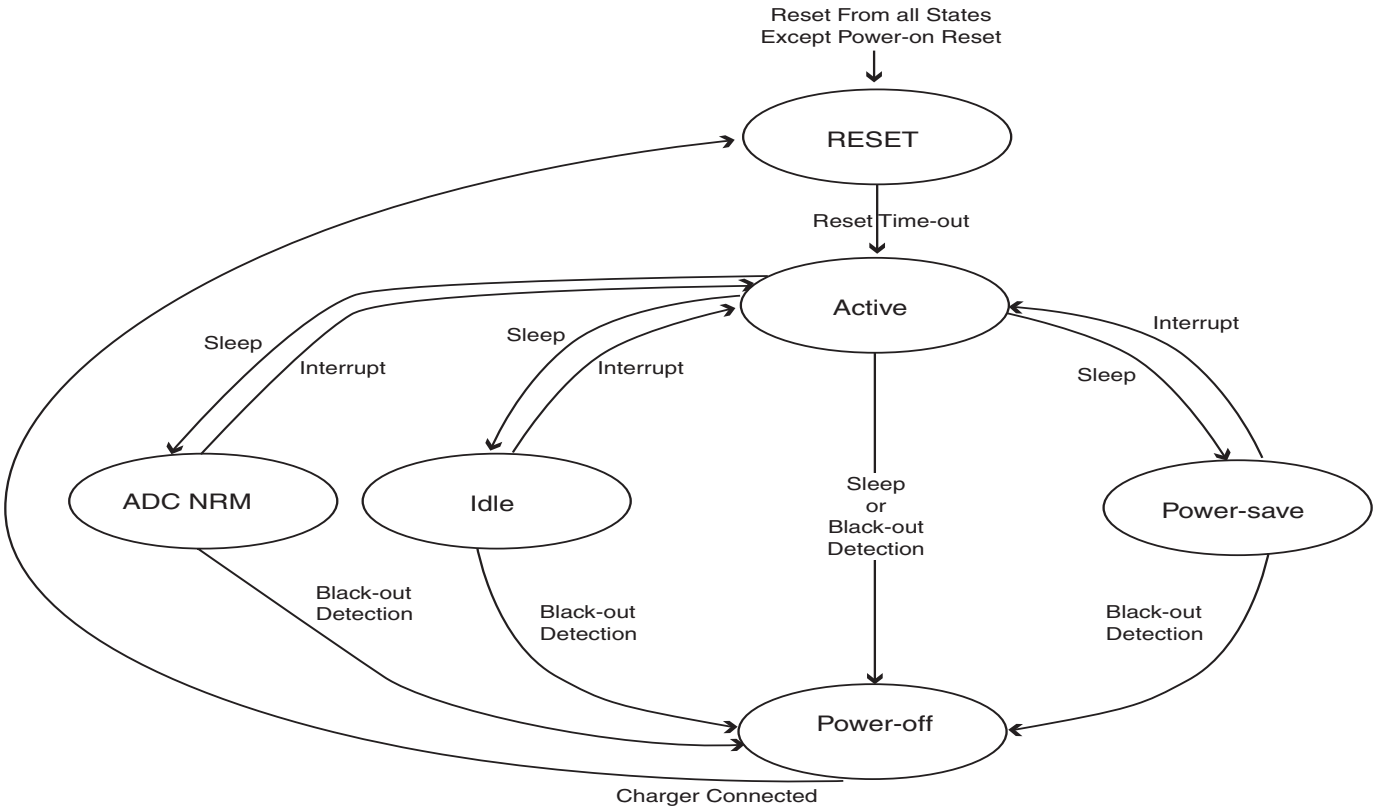


Table 10-2. Active modules in different Sleep Modes

Module	Mode				
	Active	Idle	ADC Noise Reduction	Power-save	Power-off
RCOSC_FAST	X	X	X	X(2)	
RCOSC_ULP	X	X	X	X	
RCOSC_SLOW	X(3)	X(3)	X(3)	X(3)	
CPU	X				
Flash	X				
Timer/Counter n	X	X			
SPI	X	X			
V-ADC	X	X	X		
CC-ADC	X	X(4)	X(4)	X(4)	
External Interrupts	X	X	X	X	
CBP	X	X	X	X	

- **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.

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

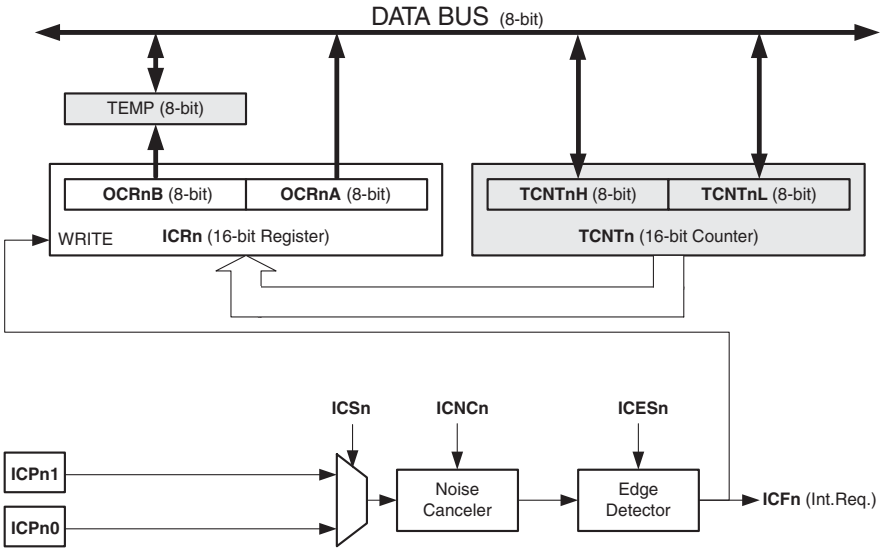
The Timer/Counter can also be used in a 16-bit Input Capture mode, see Table 17-2 on page 80 for bit settings. For full description, see "Input Capture Unit" on page 82.

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.

17.10.2 TCNTnL – Timer/Counter n Register Low Byte

Bit	7	6	5	4	3	2	1	0	
	TCNTnL[7:0]								TCNTnL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Timer/Counter Register TCNTnL gives direct access, both for read and write operations, to the Timer/Counter unit 8-bit counter. Writing to the TCNTnL Register blocks (disables) the Compare Match on the following timer clock. Modifying the counter (TCNTnL) while the counter is running, introduces a risk of missing a Compare Match between TCNTnL and the OCRnx Registers. In 16-bit mode the TCNTnL register contains the lower part of the 16-bit Timer/Counter n Register.

17.10.3 TCNTnH – Timer/Counter n Register High Byte

Bit	7	6	5	4	3	2	1	0	
	TCNTnH[7:0]								TCNTnH
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

When 16-bit mode is selected (the TCWn bit is set to one) the Timer/Counter Register TCNTnH combined to the Timer/Counter Register TCNTnL gives direct access, both for read and write operations, to the Timer/Counter unit 16-bit counter. To ensure that both the high and low bytes are read and written simultaneously when the CPU accesses these registers, the access is performed using an 8-bit temporary high byte register (TEMP). This temporary register is shared by all the other 16-bit registers. See ”Accessing Registers in 16-bit Mode” on page 86. In 8-bit mode, this register is accessable for both reading and writing, but will not be updated by the counter.

17.10.4 OCRnA – Timer/Counter n Output Compare Register A

Bit	7	6	5	4	3	2	1	0	
	OCRnA[7:0]								OCRnA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Output Compare Register A contains an 8-bit value that is continuously compared with the counter value (TCNTnL). A match can be used to generate an Output Compare interrupt.

In 16-bit mode the OCRnA register contains the low byte of the 16-bit Output Compare Register. To ensure that both the high and the low bytes are written simultaneously when the CPU writes to these registers, the access is performed using an 8-bit temporary high byte register (TEMP). This temporary register is shared by all the other 16-bit registers. See ”Accessing Registers in 16-bit Mode” on page 86.

Note that the OCRnA is not writable in Input Capture mode.

Figure 18-3. SPI Transfer Format with CPHA = 0

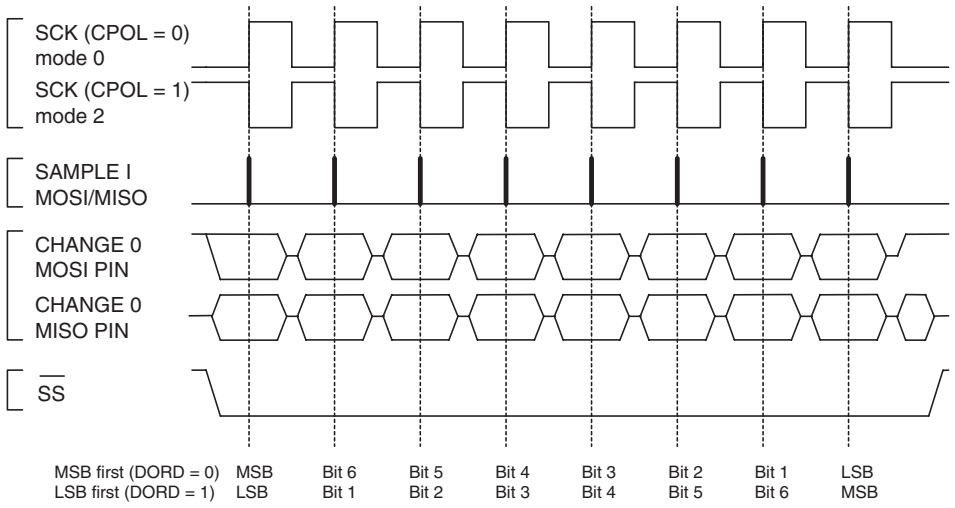
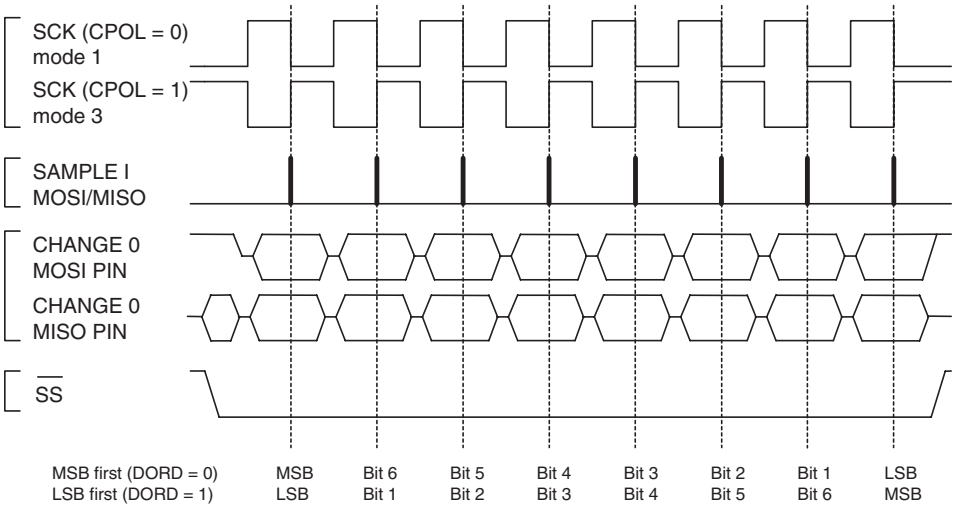


Figure 18-4. SPI Transfer Format with CPHA = 1



19.7.3 CADICH and CADICL - CC-ADC Instantaneous Current

Bit	15	14	13	12	11	10	9	8	
	7	6	5	4	3	2	1	0	
(0xE9)	CADIC[15:8]								CADICH
(0xE8)	CADIC[7:0]								CADICL
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

When a CC-ADC Instantaneous Current conversion is complete, the result is found in these two registers. CADIC[15:0] represents the converted result in 2's complement format. CADIC[12:0] are the 13-bit ADC result (including sign), while CADIC[15:13] are the sign extension bits.

When CADICL is read, the CC-ADC Instantaneous Current register is not updated until CADCH is read. Reading the registers in the sequence CADICL, CADICH will ensure that consistent values are read. When a conversion is completed, both registers must be read before the next conversion is completed, otherwise data will be lost.

19.7.4 CADAC3, CADADC2, CADAC1, CADAC0 - CC-ADC Accumulate Current

Bit	31	30	29	28	27	26	25	24	
	23	22	21	20	19	18	17	16	
	15	14	13	12	11	10	9	8	
	7	6	5	4	3	2	1	0	
(0xE3)	CADAC[31:24]								CADAC3
(0xE2)	CADAC[23:16]								CADAC2
(0xE1)	CADAC[15:8]								CADAC1
(0xE0)	CADAC[7:0]								CADAC0
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

The CADAC3, CADAC2, CADAC1 and CADAC0 Registers contain the Accumulate Current measurements in 2's complement format. CADAC[17:0] are the 18-bit ADC result (including sign), while CADAC[31:18] are the sign extension bits.

When CADAC0 is read, the CC-ADC Accumulate Current register is not updated until CADAC3 is read. Reading the registers in the sequence CADAC0, CADAC1, CADAC2, CADAC3 will ensure that consistent values are read. When a conversion is completed, all four registers must be read before the next conversion is completed, otherwise data will be lost.

20.4 Register Description

20.4.1 VADMUX – V-ADC Multiplexer Selection Register

Bit	7	6	5	4	3	2	1	0	
(0x7C)	–	–	–	–	VADMUX3	VADMUX2	VADMUX1	VADMUX0	VADMUX
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:4 – RES: Reserved Bits**

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

- **Bit 3:0 – VADMUX3:0: V-ADC Channel Selection Bits**

The VADMUX bits determine the V-ADC channel selection. See Table 20-1 on page 114.

Table 20-1. VADMUX channel selection

VADMUX3:0	Channel Selected	Scale
0000	RESERVED	–
0001	CELL 1	0.2
0010	CELL 2	0.2
0011	RESERVED	–
0100	VTEMP(1)	1.0
0101	ADC0	1.0
0110	ADC1	1.0
0111...1111	RESERVED	–

Note: 1. VTEMP must be measured in Active mode to get the highest accuracy when using calibration value stored in signature row.

20.4.2 VADCSR – V-ADC Control and Status Register

Bit	7	6	5	4	3	2	1	0	
(0x7A)	–	–	–	–	VADEN	VADSC	VADCCIF	VADCCIE	VADCSR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:4 – RES: Reserved Bits**

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

- **Bit 3 – VADEN: V-ADC Enable**

Writing this bit to one enables V-ADC conversion. By writing it to zero, the V-ADC is turned off. Turning the V-ADC off while a conversion is in progress will terminate this conversion. Note that the bandgap voltage reference must be enabled separately, see “BGCCR – Bandgap Calibration C Register” on page 118.

- **Bit 2 – VADSC: Voltage ADC Start Conversion**

Write this bit to one to start a new conversion of the selected channel.

Table 23-5. DL[7:0] with corresponding R_{SENSE} Current for all Current Detection Levels (R_{SENSE} = 10 mΩ, VREF = 1.100 ± 0.005V, T_A = -10°C to 70°C) (Continued)

Current Protection Detection Level			
0xFE		7.5A	
0xDD		8.0A	
0xDE		8.5A	
0xDF		9.0A	
0xBD		9.5A	
0xBE		10.0A	
0x9D		11.0A	
0x9E		12.0A	
0x7C		13.0A	
0x7D		14.0A	
0x7E		15.0A	
0x7F		16.0A	
0x5C		17.0A	
0x5D		18.0A	
0x5E		19.0A	
All other values	Reserved		

23.9.11 BPIMSK – Battery Protection Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
(0xF2)	-	-	-	SCIE	DOCIE	COCIE	DHCIE	CHCIE	BPIMSK
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 Bits**

These bits are reserved and will always read as zero.

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

The SCIE bit enables interrupt caused by the Short-circuit Protection Activated Interrupt.

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

The DOCIE bit enables interrupt caused by the Discharge Over-current Protection Activated Interrupt.

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

The COCIE bit enables interrupt caused by the Charge Over-current Protection Activated Interrupt.

- **Bit 1 - DHCIE : Discharger High-current Protection Activated Interrupt**

The DHCIE bit enables interrupt caused by the Discharge High-current Protection Activated Interrupt.

A Flash program corruption can be caused by two situations when the voltage is too low. First, a regular write sequence to the Flash requires a minimum voltage to operate correctly. Secondly, the CPU itself can execute instructions incorrectly, if the supply voltage for executing instructions is too low.

Flash corruption can easily be avoided by following these design recommendations (one is sufficient):

1. Keep the AVR RESET active (low) during periods of insufficient power supply voltage. This can be done by enabling the internal Brown-out Detector (BOD) if the operating voltage matches the detection level. If not, an external low V_{CC} reset protection circuit can be used. If a reset occurs while a write operation is in progress, the write operation will be completed provided that the power supply voltage is sufficient.
2. Keep the AVR core in Power-save sleep mode during periods of low V_{CC}. This will prevent the CPU from attempting to decode and execute instructions, effectively protecting the SPMCSR Register and thus the Flash from unintentional writes.

26.2.5 Reading the Signature Row from Software

To read the Signature Row from software, load the Z-pointer with the signature byte address given in Table 26-1 and set the SIGRD and SPMEN bits in SPMCSR. When an LPM instruction is executed within three CPU cycles after the SIGRD and SPMEN bits are set in SPMCSR, the signature byte value will be loaded in the destination register. The SIGRD and SPMEN bits will auto-clear 6 cycles after writing to SPMCSR, which is locked for further writing during these cycles. The LPM instruction must be executed within 3 CPU cycles after writing SPMCSR. When SIGRD and SPMEN are cleared, LPM will work as described in the Instruction set Manual.

Table 26-1. Signature Row Addressing.

Signature Byte Description	Z-Pointer Address
Device ID 0, Manufacture ID	00H
Device ID 1, Flash Size	02H
Device ID 2, Device	04H
FOSCCAL(1)	01H
FOSC SEGMENT(2)	03H
Reserved	05H
SLOW RC Period L	06H
SLOW RC Period H(3)	07H
SLOW RC Temp Prediction L	08H
SLOW RC Temp Prediction H(4)	09H
ULP RC FRQ(6)	0AH
SLOW RC FRQ(5)	0BH
Reserved	0CH:0EH
BGCCR Calibration Byte @ 25°C	0FH
Reserved	10H
BGCRR Calibration Byte @ 25°C	11H

- **Bit 2 – PGWRT: Page Write**

If this bit is written to one at the same time as SPMEN, the next SPM instruction within four clock cycles executes Page Write, with the data stored in the temporary buffer. The page address is taken from the high part of the Z-pointer. The data in R1 and R0 are ignored. The PGWRT bit will auto-clear upon completion of a Page Write, or if no SPM instruction is executed within four clock cycles. The CPU is halted during the entire Page Write operation.

- **Bit 1 – PGERS: Page Erase**

If this bit is written to one at the same time as SPMEN, the next SPM instruction within four clock cycles executes Page Erase. The page address is taken from the high part of the Z-pointer. The data in R1 and R0 are ignored. The PGERS bit will auto-clear upon completion of a Page Erase, or if no SPM instruction is executed within four clock cycles. The CPU is halted during the entire Page Write operation.

- **Bit 0 – SPMEN: Store Program Memory Enable**

This bit enables the SPM instruction for the next four clock cycles. If written to one together with either CTPB, RFLB, PGWRT, or PGERS, the following SPM instruction will have a special meaning, see description above. If only SPMEN is written, the following SPM instruction will store the value in R1:R0 in the temporary page buffer addressed by the Z-pointer. The LSB of the Z-pointer is ignored. The SPMEN bit will auto-clear upon completion of an SPM instruction, or if no SPM instruction is executed within four clock cycles. During Page Erase and Page Write, the SPMEN bit remains high until the operation is completed.

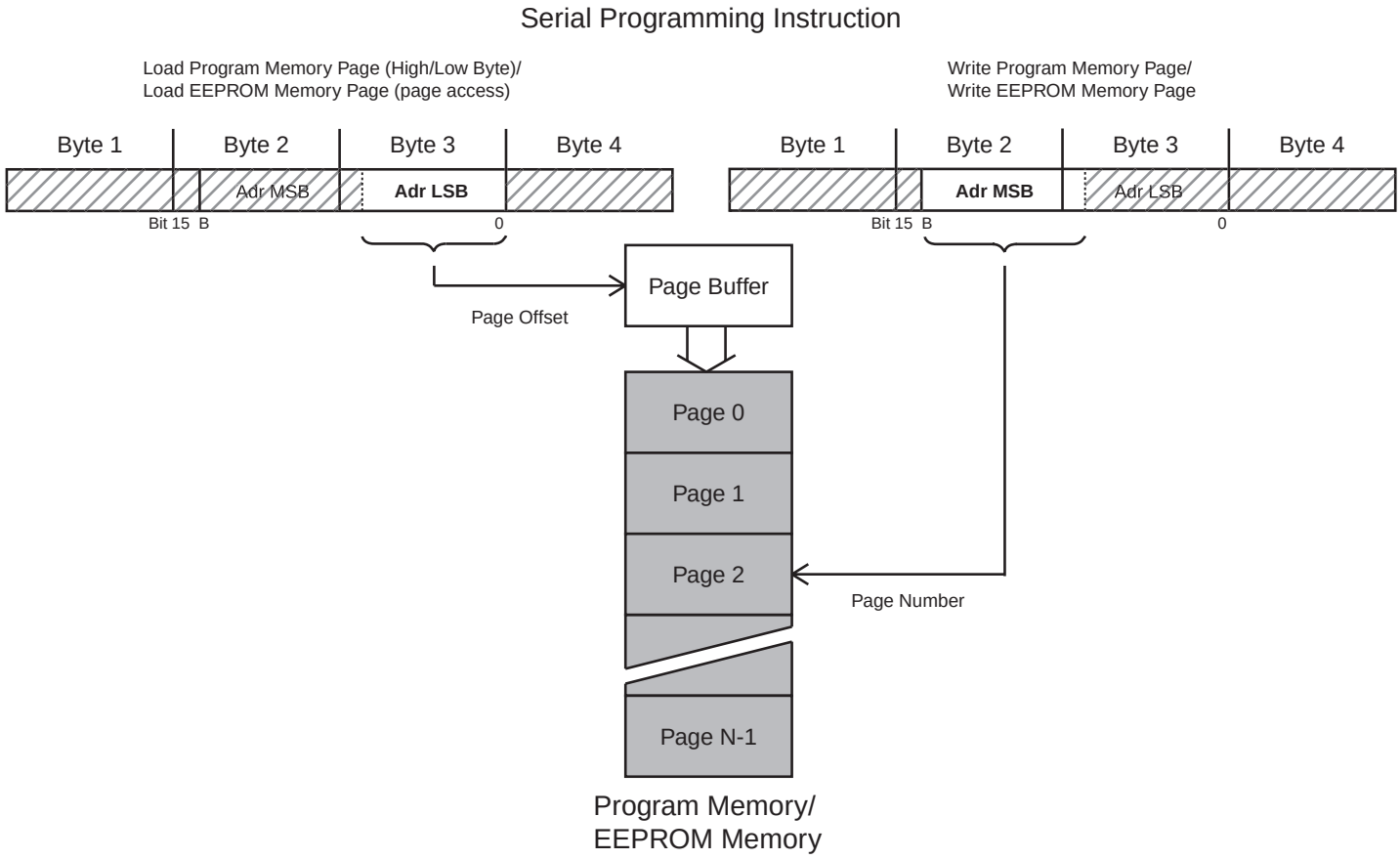
Writing any other combination than “100001”,“010001”, “001001”, “000101”, “000011” or “000001” in the lower six bits will have no effect.

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

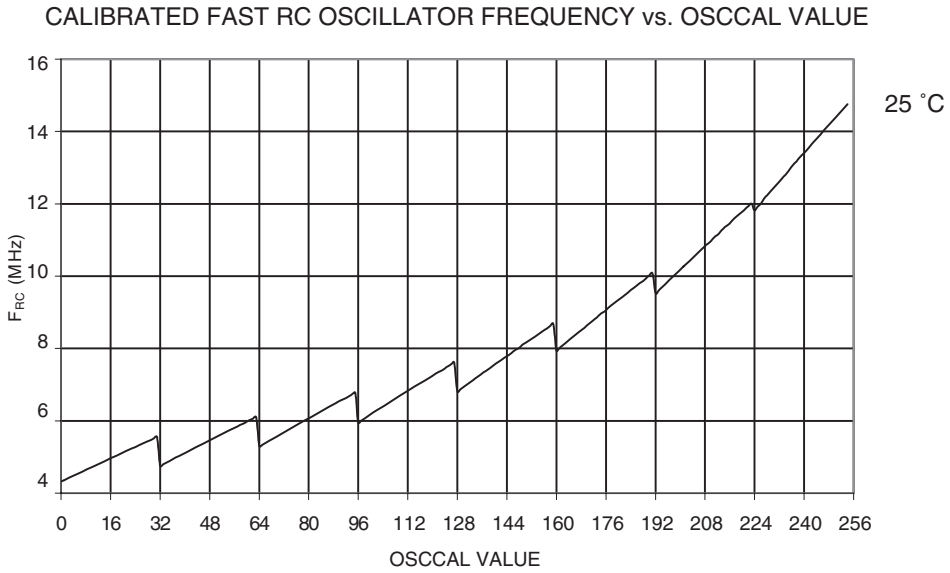


30. Typical Characteristics – Preliminary Data

All Typical Characteristics contained in this data sheet are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These figures are preliminary and will be updated after characterization of actual silicon.

These figures are not tested during manufacturing, and are added for illustration purpose only.

Figure 30-1. Fast RC Oscillator frequency vs. OSCCAL value.



31. Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(0xFF)	Reserved	–	–	–	–	–	–	–	–	
(0xFE)	BPPLR	–	–	–	–	–	–	BPPLE	BPPL	127
(0xFD)	BPCR	–	–	–	SCD	DOCD	COCD	DHCD	CHCD	127
(0xFC)	BPHCTR	–	–	HCPT[5:0]						130
(0xFB)	BPOCTR	–	–	OCPT[5:0]						129
(0xFA)	BPSCTR	–	SCPT[6:0]							128
(0xF9)	BPCHCD	CHCDL[7:0]								132
(0xF8)	BPDHCD	DHCDL[7:0]								132
(0xF7)	BPCOCD	COCDL[7:0]								131
(0xF6)	BPDOCD	DOCDL[7:0]								131
(0xF5)	BPSCD	SCDL[7:0]								131
(0xF4)	Reserved	–	–	–	–	–	–	–	–	
(0xF3)	BPIFR	–	–	–	SCIF	DOCIF	COCIF	DHCIF	CHCIF	134
(0xF2)	BPIMSK	–	–	–	SCIE	DOCIE	COCIE	DHCIE	CHCIE	133
(0xF1)	Reserved	–	–	–	–	–	–	–	–	
(0xF0)	FCSR	–	–	–	–	DUVRD	CPS	DFE	CFE	138
(0xEF)	Reserved	–	–	–	–	–	–	–	–	
(0xEE)	Reserved	–	–	–	–	–	–	–	–	
(0xED)	Reserved	–	–	–	–	–	–	–	–	
(0xEC)	Reserved	–	–	–	–	–	–	–	–	
(0xEB)	Reserved	–	–	–	–	–	–	–	–	
(0xEA)	Reserved	–	–	–	–	–	–	–	–	
(0xE9)	CADICH	CADIC[15:8]								110
(0xE8)	CADICL	CADIC[7:0]								110
(0xE7)	Reserved	–	–	–	–	–	–	–	–	
(0xE6)	CADRC	CADRC[7:0]								111
(0xE5)	CADCSRB	–	CADACIE	–	CADICIE	–	CADACIF	CADRCIF	CADICIF	109
(0xE4)	CADCSRA	CADEN	CADPOL	CADUB	CADAS[1:0]		CADSI[1:0]		CADSE	107
(0xE3)	CADAC3	CADAC[31:24]								110
(0xE2)	CADAC2	CADAC[23:16]								110
(0xE1)	CADAC1	CADAC[15:8]								110
(0xE0)	CADAC0	CADAC[7:0]								110
(0xDF)	Reserved	–	–	–	–	–	–	–	–	
(0xDE)	Reserved	–	–	–	–	–	–	–	–	
(0xDD)	Reserved	–	–	–	–	–	–	–	–	
(0xDC)	Reserved	–	–	–	–	–	–	–	–	
(0xDB)	Reserved	–	–	–	–	–	–	–	–	
(0xDA)	Reserved	–	–	–	–	–	–	–	–	
(0xD9)	Reserved	–	–	–	–	–	–	–	–	
(0xD8)	Reserved	–	–	–	–	–	–	–	–	
(0xD7)	Reserved	–	–	–	–	–	–	–	–	
(0xD6)	Reserved	–	–	–	–	–	–	–	–	
(0xD5)	Reserved	–	–	–	–	–	–	–	–	
(0xD4)	Reserved	–	–	–	–	–	–	–	–	
(0xD3)	Reserved	–	–	–	–	–	–	–	–	
(0xD2)	Reserved	–	–	–	–	–	–	–	–	
(0xD1)	BGCRR	BGCR[7:0]								119
(0xD0)	BGCCR	BGD	–	BGCC[5:0]						118
(0xCF)	Reserved	–	–	–	–	–	–	–	–	
(0xCE)	Reserved	–	–	–	–	–	–	–	–	
(0xCD)	Reserved	–	–	–	–	–	–	–	–	
(0xCC)	Reserved	–	–	–	–	–	–	–	–	
(0xCB)	Reserved	–	–	–	–	–	–	–	–	
(0xCA)	Reserved	–	–	–	–	–	–	–	–	
(0xC9)	Reserved	–	–	–	–	–	–	–	–	
(0xC8)	ROCR	ROCS	–	–	–	–	–	ROCWIF	ROCWIE	123
(0xC7)	Reserved	–	–	–	–	–	–	–	–	
(0xC6)	Reserved	–	–	–	–	–	–	–	–	
(0xC5)	Reserved	–	–	–	–	–	–	–	–	
(0xC4)	Reserved	–	–	–	–	–	–	–	–	
(0xC3)	Reserved	–	–	–	–	–	–	–	–	
(0xC2)	Reserved	–	–	–	–	–	–	–	–	
(0xC1)	Reserved	–	–	–	–	–	–	–	–	
(0xC0)	Reserved	–	–	–	–	–	–	–	–	

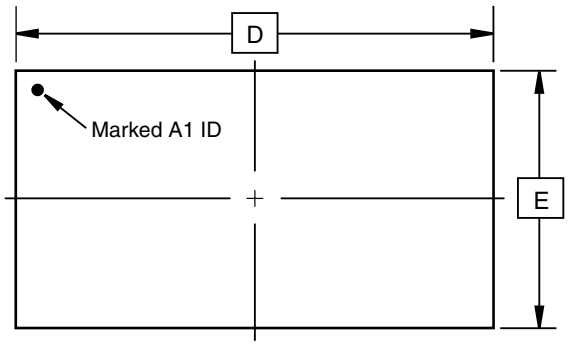
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(0x7D)	Reserved	–	–	–	–	–	–	–	–	
(0x7C)	VADMUX	–	–	–	–	VADMUX[3:0]				114
(0x7B)	Reserved	–	–	–	–	–	–	–	–	
(0x7A)	VADCSR	–	–	–	–	VADEN	VADSC	VADCCIF	VADCCIE	114
(0x79)	VADCH	–	–	–	–	VADC Data Register High byte				115
(0x78)	VADCL	VADC Data Register Low byte								115
(0x77)	Reserved	–	–	–	–	–	–	–	–	
(0x76)	Reserved	–	–	–	–	–	–	–	–	
(0x75)	Reserved	–	–	–	–	–	–	–	–	
(0x74)	Reserved	–	–	–	–	–	–	–	–	
(0x73)	Reserved	–	–	–	–	–	–	–	–	
(0x72)	Reserved	–	–	–	–	–	–	–	–	
(0x71)	Reserved	–	–	–	–	–	–	–	–	
(0x70)	Reserved	–	–	–	–	–	–	–	–	
(0x6F)	TIMSK1	–	–	–	–	ICIE1	OCIE1B	OCIE1A	TOIE1	92
(0x6E)	TIMSK0	–	–	–	–	ICIE0	OCIE0B	OCIE0A	TOIE0	92
(0x6D)	Reserved	–	–	–	–	–	–	–	–	
(0x6C)	Reserved	–	–	–	–	–	–	–	–	
(0x6B)	Reserved	–	–	–	–	–	–	–	–	
(0x6A)	Reserved	–	–	–	–	–	–	–	–	
(0x69)	EICRA	–	–	ISC21	ISC20	ISC11	ISC10	ISC01	ISC00	56
(0x68)	Reserved	–	–	–	–	–	–	–	–	
(0x67)	Reserved	–	–	–	–	–	–	–	–	
(0x66)	FOSCCAL	Fast Oscillator Calibration Register								30
(0x65)	Reserved	–	–	–	–	–	–	–	–	
(0x64)	PRR0	–	–	PRVRM	–	PRSPI	PRTIM1	PRTIM0	PRVADC	39
(0x63)	Reserved	–	–	–	–	–	–	–	–	
(0x62)	Reserved	–	–	–	–	–	–	–	–	
(0x61)	CLKPR	CLKPCE	–	–	–	–	–	CLKPS1	CLKPS0	31
(0x60)	WDTCSR	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	49
0x3F (0x5F)	SREG	I	T	H	S	V	N	Z	C	9
0x3E (0x5E)	SPH	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	12
0x3D (0x5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	12
0x3C (0x5C)	Reserved	–	–	–	–	–	–	–	–	
0x3B (0x5B)	Reserved	–	–	–	–	–	–	–	–	
0x3A (0x5A)	Reserved	–	–	–	–	–	–	–	–	
0x39 (0x59)	Reserved	–	–	–	–	–	–	–	–	
0x38 (0x58)	Reserved	–	–	–	–	–	–	–	–	
0x37 (0x57)	SPMCSR	–	–	SIGRD	CTPB	RFLB	PGWRT	PGERS	SPMEN	147
0x36 (0x56)	Reserved	–	–	–	–	–	–	–	–	
0x35 (0x55)	MCUCR	–	–	CKOE	PUD	–	–	–	–	73/31
0x34 (0x54)	MCUSR	–	–	–	OCDRF	WDRF	BODRF	EXTRF	PORF	49
0x33 (0x53)	SMCR	–	–	–	–	SM[2:0]			SE	39
0x32 (0x52)	Reserved	–	–	–	–	–	–	–	–	
0x31 (0x51)	DWDR	debugWIRE Data Register								140
0x30 (0x50)	Reserved	–	–	–	–	–	–	–	–	
0x2F (0x4F)	Reserved	–	–	–	–	–	–	–	–	
0x2E (0x4E)	SPDR	SPI Data Register								103
0x2D (0x4D)	SPSR	SPIF	WCOL	–	–	–	–	–	SPI2X	102
0x2C (0x4C)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	101
0x2B (0x4B)	GPIOR2	General Purpose I/O Register 2								23
0x2A (0x4A)	GPIOR1	General Purpose I/O Register 1								23
0x29 (0x49)	OCR0B	Timer/Counter0 Output Compare Register B								92
0x28 (0x48)	OCR0A	Timer/Counter0 Output Compare Register A								91
0x27 (0x47)	TCNT0H	Timer/Counter0 (8 Bit) High Byte								91
0x26 (0x46)	TCNT0L	Timer/Counter0 (8 Bit) Low Byte								91
0x25 (0x45)	TCCR0B	–	–	–	–	–	CS02	CS01	CS00	76
0x24 (0x44)	TCCR0A	TCW0	ICEN0	ICNC0	ICES0	ICS0	–	–	WGM00	90
0x23 (0x43)	GTCCR	TSM	–	–	–	–	–	–	PSRSYNC	
0x22 (0x42)	Reserved	–	–	–	–	–	–	–	–	
0x21 (0x41)	EEAR	EEPROM Address Register Low Byte								19
0x20 (0x40)	EEDR	EEPROM Data Register								19
0x1F (0x3F)	EECR	–	–	EEPM1	EEPM0	EERIE	EEMPE	EEPE	EERE	19
0x1E (0x3E)	GPIOR0	General Purpose I/O Register 0								23
0x1D (0x3D)	EIMSK	–	–	–	–	–	INT2	INT1	INT0	57
0x1C (0x3C)	EIFR	–	–	–	–	–	INTF2	INTF1	INTF0	57

32. Instruction Set Summary (Continued)

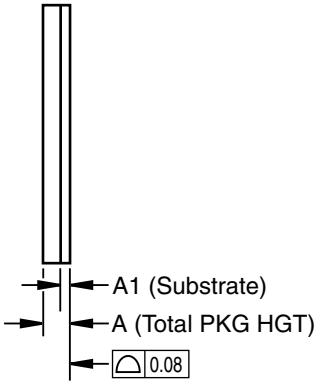
Mnemonics	Operands	Description	Operation	Flags	#Clocks
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	None	1/2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	None	1/2
BIT AND BIT-TEST INSTRUCTIONS					
SBI	P,b	Set Bit in I/O Register	I/O(P,b) ← 1	None	2
CBI	P,b	Clear Bit in I/O Register	I/O(P,b) ← 0	None	2
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0)←C,Rd(n+1)← Rd(n),C←Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry	Rd(7)←C,Rd(n)← Rd(n+1),C←Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0)←Rd(7..4),Rd(7..4)←Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Twos Complement Overflow.	V ← 1	V	1
CLV		Clear Twos Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half Carry Flag in SREG	H ← 1	H	1
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1
DATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
MOVW	Rd, Rr	Copy Register Word	Rd+1:Rd ← Rr+1:Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LD	Rd, X	Load Indirect	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	Rd ← (X), X ← X + 1	None	2
LD	Rd, - X	Load Indirect and Pre-Dec.	X ← X - 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, - Y	Load Indirect and Pre-Dec.	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd,Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z), Z ← Z+1	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
LDS	Rd, k	Load Direct from SRAM	Rd ← (k)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr, X ← X + 1	None	2
ST	- X, Rr	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr, Y ← Y + 1	None	2
ST	- Y, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q,Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr, Z ← Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q,Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
STS	k, Rr	Store Direct to SRAM	(k) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
LPM	Rd, Z	Load Program Memory	Rd ← (Z)	None	3
LPM	Rd, Z+	Load Program Memory and Post-Inc	Rd ← (Z), Z ← Z+1	None	3
SPM		Store Program Memory	(Z) ← R1:R0	None	-
IN	Rd, P	In Port	Rd ← P	None	1

34. Packaging Information

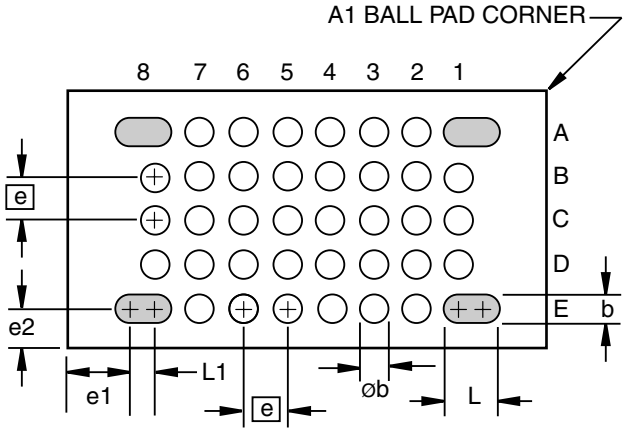
34.1 36CK1



Top View



Side View



Bottom View

COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
D	6.40	6.50	6.60	
E	3.40	3.50	3.60	
A	0.59	0.66	0.73	
A1	0.17	0.21	0.25	
L		0.70 REF		2
L1		0.35 REF		
b		0.35 REF		2
øb	0.32	0.35	0.38	2
e		0.60 TYP		
e1		0.80 REF		
e2		0.55 REF		

Notes: 1. This drawing is for general information only.

2. Metal pad dimensions.

3. ⬭ = > Dummy pad.

3/15/07

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**36CK1**, 36-Pad, 6.50 x 3.50 x 0.73 mm Body, 0.60 mm Pitch, Land Grid Array (LGA) Package	36CK1	D

35. Errata

35.1 ATmega8HVA

35.1.1 Rev. A

No known errata.

35.2 ATmega16HVA

35.2.1 Rev. A

No known errata.