Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	AVR
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	28
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.7V ~ 5.5V
Data Converters	A/D 28x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-TQFP
Supplier Device Package	32-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/attiny828r-aur

Note: 1. This is the initial value when CKDIV8 fuse has been unprogrammed.
2. This is the initial value when CKDIV8 fuse has been programmed. The device is shipped with the CKDIV8 Fuse programmed.

The initial value of clock prescaler bits is determined by the CKDIV8 fuse (see Table 91 on page 227). When CKDIV8 is unprogrammed, the system clock prescaler is set to one and, when programmed, to eight. Any value can be written to the CLKPS bits regardless of the CKDIV8 fuse bit setting.

When CKDIV8 is programmed the initial value of CLKPS bits give a clock division factor of eight at start up. This is useful when the selected clock source has a higher frequency than allowed under present operating conditions. See "Speed" on page 249.

To avoid unintentional changes to clock frequency, the following sequence must be followed:

1. Write the required signature to the CCP register. See page 14.
2. Within four instruction cycles, write the desired value to CLKPS bits.

6.6.2 OSCCAL0 – Oscillator Calibration Register

Bit	7	6	5	4	3	2	1	0	
(0x66)	CAL07	CAL06	CAL05	CAL04	CAL03	CAL02	CAL01	CAL00	OSCCAL0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	Device Specific Calibration Value								

- **Bits 7:0 – CAL0[7:0]: Oscillator Calibration Value**

The oscillator calibration register is used to trim the internal 8MHz oscillator and to remove process variations from the oscillator frequency. A pre-programmed calibration value is automatically written to this register during chip reset, giving the factory calibrated frequency specified in Table 104 on page 249.

The application software can write this register to change the oscillator frequency. The oscillator can be calibrated to frequencies specified in Table 104 on page 249. Calibration outside that range is not guaranteed.

The lowest oscillator frequency is reached by programming these bits to zero. Increasing the register value increases the oscillator frequency. A typical frequency response curve is shown in "Calibrated Oscillator Frequency vs. OSCCAL0 Value" on page 295.

Note that this oscillator is used to time EEPROM and Flash write accesses, and write times will be affected accordingly. Do not calibrate to more than 8.8MHz if EEPROM or Flash is to be written. Otherwise, the EEPROM or Flash write may fail.

To ensure stable operation of the MCU the calibration value should be changed in small steps. A step change in frequency of more than 2% from one cycle to the next can lead to unpredictable behavior. Also, the difference between two consecutive register values should not exceed 0x20. If these limits are exceeded the MCU must be kept in reset during changes to clock frequency.

the ACD bit in ACSRA. See “ACSRA – Analog Comparator Control and Status Register” on page 134. This will reduce power consumption in Idle mode.

If the ADC is enabled, a conversion starts automatically when this mode is entered.

7.1.2 ADC Noise Reduction Mode

This sleep mode halts $clk_{I/O}$, clk_{CPU}, and clk_{FLASH}, while allowing other clocks to run. In ADC Noise Reduction mode, the CPU is stopped but the following peripherals continue to operate:

- Watchdog (if enabled), and external interrupts
- ADC
- USART start frame detector, and TWI

This improves the noise environment for the ADC, enabling higher resolution measurements. If the ADC is enabled, a conversion starts automatically when this mode is entered.

The following events can wake up the MCU:

- Watchdog reset, external reset, and brown-out reset
- External level interrupt on INT0, and pin change interrupt
- ADC conversion complete interrupt, and SPM/EEPROM ready interrupt
- USART start frame detection, and TWI slave address match

7.1.3 Power-Down Mode

This sleep mode halts all generated clocks, allowing operation of asynchronous modules, only. In Power-down Mode the oscillator is stopped, while the following peripherals continue to operate:

- Watchdog (if enabled), external interrupts

The following events can wake up the MCU:

- Watchdog reset, external reset, and brown-out reset
- External level interrupt on INT0, and pin change interrupt
- USART start frame detection, and TWI slave address match

7.2 Power Reduction Register

The Power Reduction Register (PRR), see “PRR – Power Reduction Register” on page 37, provides a method to reduce power consumption by stopping the clock to individual peripherals. When the clock for a peripheral is stopped then:

- The current state of the peripheral is frozen.
- The associated registers can not be read or written.
- Resources used by the peripheral will remain occupied.

The peripheral should in most cases be disabled before stopping the clock. Clearing the PRR bit wakes up the peripheral and puts it in the same state as before shutdown.

Peripheral shutdown can be used in Idle mode and Active mode to significantly reduce the overall power consumption. See “Current Consumption of Peripheral Units” on page 266 for examples. In all other sleep modes, the clock is already stopped.

7.3 Minimizing Power Consumption

There are several issues to consider when trying to minimize the power consumption in an AVR controlled system. In general, sleep modes should be used as much as possible, and the sleep mode should be selected so that as few as

Table 12. Setting BOD Mode of Operation in Sleep Modes Other Than Idle

BODPD1	BODPD0	Mode of Operation
0	0	Reserved
0	1	Sampled
1	0	Enabled
1	1	Disabled

See “Fuse Bits” on page 226.

8.3 Internal Voltage Reference

ATtiny828 features an internal bandgap reference. This reference is used for Brown-out Detection, and it can be used as an input to the Analog Comparator or the ADC. The bandgap voltage varies with supply voltage and temperature.

8.3.1 Voltage Reference Enable Signals and Start-up Time

The voltage reference has a start-up time that may influence the way it should be used. The start-up time is given in “System and Reset Characteristics” on page 250. To save power, the reference is not always turned on. The reference is on during the following situations:

- When the BOD is enabled.
- When the internal reference is connected to the Analog Comparator.
- When the ADC is enabled.

Thus, when the BOD is not enabled, after setting the ACBG bit or enabling the ADC, the user must always allow the reference to start up before the output from the Analog Comparator or ADC is used. To reduce power consumption in Power-down mode, the user can avoid the three conditions above to ensure that the reference is turned off before entering Power-down mode.

8.4 Watchdog Timer

The Watchdog Timer is clocked from the internal 32kHz ultra low power oscillator (see page 29). By controlling the Watchdog Timer prescaler, the Watchdog Reset interval can be adjusted as shown in Table 15 on page 47. The WDR – Watchdog Reset – instruction resets the Watchdog Timer. The Watchdog Timer is also reset when it is disabled and when a Chip Reset occurs. Ten different clock cycle periods can be selected to determine the reset period. If the reset period expires without another Watchdog Reset, the ATtiny828 resets and executes from the Reset Vector.

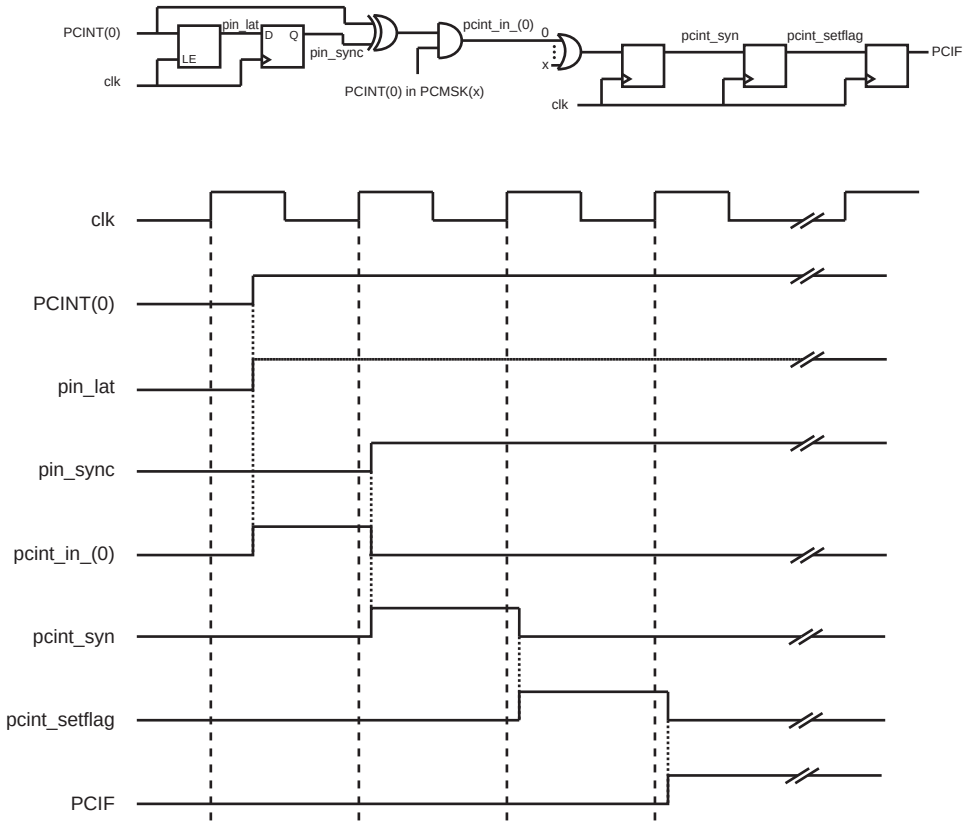
The Wathdog Timer can also be configured to generate an interrupt instead of a reset. This can be very helpful when using the Watchdog to wake-up from Power-down.

To prevent unintentional disabling of the Watchdog or unintentional change of time-out period, two different safety levels are selected by the fuse WDTON as shown in Table 13 See “Timed Sequences for Changing the Configuration of the Watchdog Timer” on page 44 for details.

Table 13. WDT Configuration as a Function of the Fuse Settings of WDTON

WDTON	Safety Level	WDT Initial State	How to Disable the WDT	How to Change Time-out
Unprogrammed	1	Disabled	Timed sequence	No limitations
Programmed	2	Enabled	Always enabled	Timed sequence

Figure 20. Timing of pin change interrupts



9.3 Register Description

9.3.1 MCUCR – MCU Control Register

Bit	7	6	5	4	3	2	1	0	
0x35 (0x55)	–	–	–	–	–	–	IVSEL	–	MCUCR
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, 0 – Res: Reserved Bits**

These bits are reserved and will always read zero.

- **Bit 1 – IVSEL: Interrupt Vector Select**

When this bit is cleared, interrupt vectors are placed at the start of Flash memory. When this bit is set, interrupt vectors are moved to the beginning of the boot loader section.

The start address of the boot section is determined by the BOOTSZ Fuses. See “Configuring the Boot Loader” on page 216 for details.

If interrupt vectors are placed in the boot loader section and boot lock bit BLB02 is programmed, interrupts will be disabled while executing from the application section.

If interrupt vectors are placed in the application section and boot lock bit BLB12 is programmed, interrupts will be disabled while executing from the boot loader section.

Pin	Function	Description of Alternative Function
PC1	PCINT17	Pin change interrupt source
	ADC17	Input channel for analog to digital converter (ADC)
	TOCC1	Timer/counter output compare, channel 1 (1)
	INT0	External interrupt request 0
	CLKO	System clock output (3)
PC2	PCINT18	Pin change interrupt source
	ADC18	Input channel for analog to digital converter (ADC)
	TOCC2	Timer/counter output compare, channel 2 (1)
	RXD	USART serial data input
	INT1	External interrupt request 1
PC3	PCINT19	Pin change interrupt source
	ADC19	Input channel for analog to digital converter (ADC)
	TOCC3	Timer/counter output compare, channel 3 (1)
	TXD	USART serial data output
PC4	PCINT20	Pin change interrupt source
	ADC20	Input channel for analog to digital converter (ADC)
	TOCC4	Timer/counter output compare, channel 4 (1)
PC5	PCINT21	Pin change interrupt source
	ADC21	Input channel for analog to digital converter (ADC)
	TOCC5	Timer/counter output compare, channel 5 (1)
	ICP1	Input capture pin
	T0	Timer/Counter0 Clock Source
PC6	PCINT22	Pin change interrupt source
	ADC22	Input channel for analog to digital converter (ADC)
	TOCC6	Timer/counter output compare, channel 6 (1)
	CLKI	Clock input from external source
PC7	PCINT23	Pin change interrupt source
	ADC23	Input channel for analog to digital converter (ADC)
	TOCC7	Timer/counter output compare, channel 7 (1)
	T1	Timer/Counter1 Clock Source

Note: 1. See “TOCPMSA1 and TOCPMSA0 – Timer/Counter Output Compare Pin Mux Selection Registers” on page 127.

Pin	Signal	Composition
PC2	PUOE	0
	PUOV	0
	DDOE	RXEN
	DDOV	0
	PVOE	TOCC2OE
	PVOV	TOCC2_OUT
	PTOE	0
	DIEOE	(PCINT18 • PCIE2) + ADC18D + (RXEN ● SFDE) + INT1
	DIEOV	PCINT18 • PCIE2 + (RXEN ● SFDE) + INT1
	DI	PCINT18 Input / INT1 Input / RXD_IN
	AIO	ADC18 Input

At the very start of period 2 in Figure 32 on page 95 OCnx has a transition from high to low even though there is no Compare Match. The point of this transition is to guaratee symmetry around BOTTOM. There are two cases that give a transition without Compare Match.

- OCR0x changes its value from TOP, like in Figure 32 on page 95. When the OCR0x value is TOP the OCnx pin value is the same as the result of a down-counting Compare Match. To ensure symmetry around BOTTOM the OCnx value at TOP must correspond to the result of an up-counting Compare Match.
- The timer starts counting from a value higher than the one in OCR0x, and for that reason misses the Compare Match and hence the OCnx change that would have happened on the way up.

11.8 Timer/Counter Timing Diagrams

The Timer/Counter is a synchronous design and the timer clock (clk_{T0}) is therefore shown as a clock enable signal in the following figures. The figures include information on when Interrupt Flags are set. Figure 33 on page 96 contains timing data for basic Timer/Counter operation. The figure shows the count sequence close to the MAX value in all modes other than phase correct PWM mode.

Figure 33. Timer/Counter Timing Diagram, no Prescaling

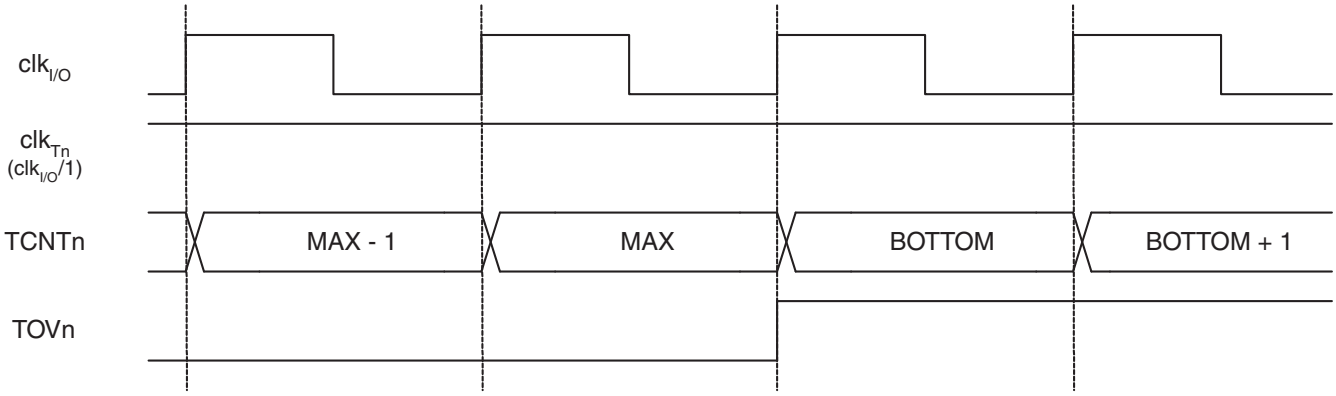


Figure 34 on page 96 shows the same timing data, but with the prescaler enabled.

Figure 34. Timer/Counter Timing Diagram, with Prescaler ($f_{clk\_I/O}$/8)

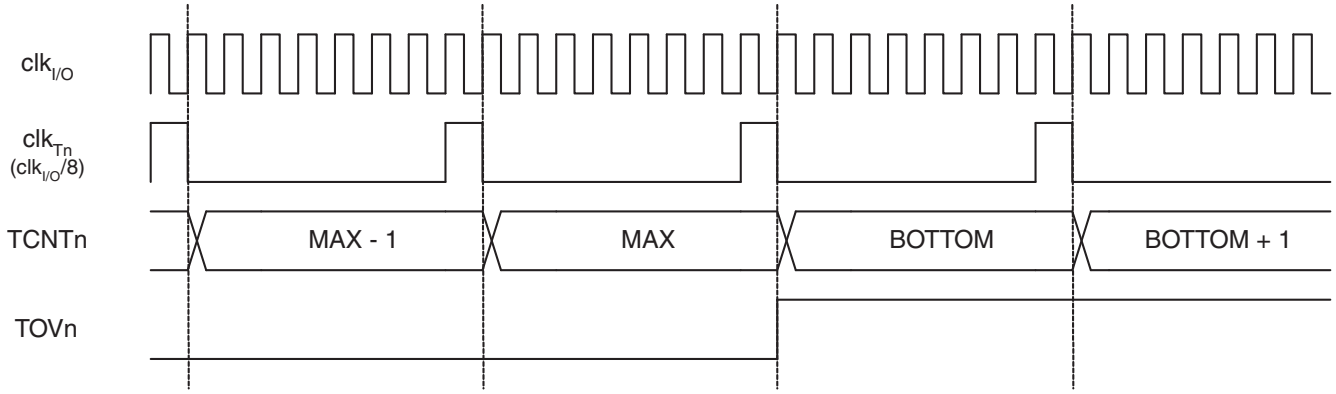
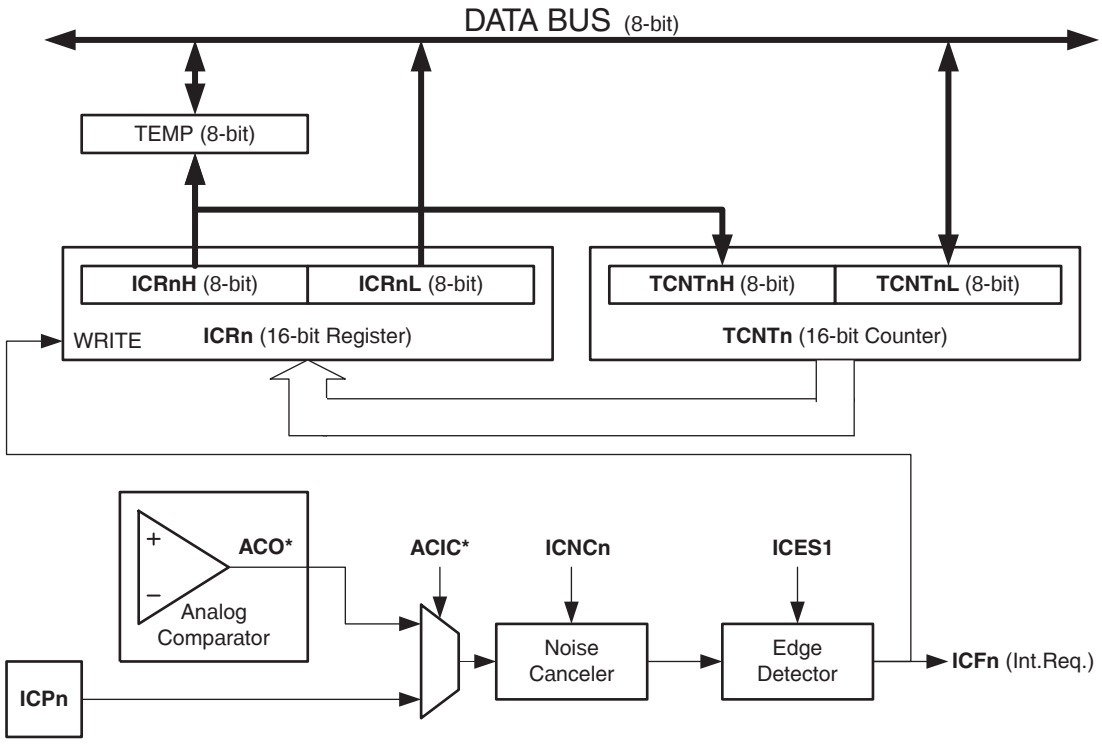


Figure 35 on page 97 shows the setting of OCF0B in all modes and OCF0A in all modes except CTC mode and PWM mode, where OCR0A is TOP.

or alternatively, via the analog-comparator unit. 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.

Figure 39. Input Capture Unit Block Diagram



The Input Capture unit is illustrated by the block diagram shown in Figure 39. The elements of the block diagram that are not directly a part of the Input Capture unit are gray shaded. The small “n” in register and bit names indicates the Timer/Counter number.

When a change of the logic level (an event) occurs on the Input Capture pin (ICP1), alternatively on the Analog Comparator output (ACO), and this change confirms to the setting of the edge detector, a capture will be triggered. When a capture is triggered, the 16-bit value of the counter (TCNT1) is written to the Input Capture Register (ICR1). The Input Capture Flag (ICF1) is set at the same system clock as the TCNT1 value is copied into ICR1 Register. If enabled (ICIE1 = 1), the Input Capture Flag generates an Input Capture interrupt. The ICF1 flag is automatically cleared when the interrupt is executed. Alternatively the ICF1 flag can be cleared by software by writing a logical one to its I/O bit location.

Reading the 16-bit value in the Input Capture Register (ICR1) is done by first reading the low byte (ICR1L) and then the high byte (ICR1H). When the low byte is read the high byte is copied into the high byte temporary register (TEMP). When the CPU reads the ICR1H I/O location it will access the TEMP Register.

The ICR1 Register can only be written when using a Waveform Generation mode that utilizes the ICR1 Register for defining the counter’s TOP value. In these cases the Waveform Generation mode (WGM1[3:0]) bits must be set before the TOP value can be written to the ICR1 Register. When writing the ICR1 Register the high byte must be written to the ICR1H I/O location before the low byte is written to ICR1L.

For more information on how to access the 16-bit registers refer to “Accessing 16-bit Registers” on page 120.

12.5.1 Input Capture Trigger Source

The main trigger source for the Input Capture unit is the Input Capture pin (ICP1). Timer/Counter1 can alternatively use the Analog Comparator output as trigger source for the Input Capture unit. The Analog Comparator is selected as trigger source by setting the Analog Comparator Input Capture (ACIC) bit in the Analog Comparator Control and Status Register (ACSR). Be aware that changing trigger source can trigger a capture. The Input Capture Flag must therefore be cleared after the change.

C Code Example

```
void TIM16_WriteTCNT1( unsigned int i )
{
      unsigned char sreg;
      unsigned int i;

      /* Save global interrupt flag */
      sreg = SREG;

      /* Disable interrupts */
      _CLI();

      /* Set TCNT1 to i */
      TCNT1 = i;

      /* Restore global interrupt flag */
      SREG = sreg;
}
```

Note: See “Code Examples” on page 7.

The assembly code example requires that the r17:r16 register pair contains the value to be written to TCNT1.

12.10.1 Reusing the Temporary High Byte Register

If writing to more than one 16-bit register where the high byte is the same for all registers written, then the high byte only needs to be written once. However, note that the same rule of atomic operation described previously also applies in this case.

12.11 Register Description

12.11.1 TCCR1A – Timer/Counter1 Control Register A

Bit	7	6	5	4	3	2	1	0	
(0x80)	**COM1A1**	**COM1A0**	**COM1B1**	**COM1B0**	**–**	**–**	**WGM11**	**WGM10**	**TCCR1A**
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:6 – COM1A[1:0] : Compare Output Mode for Channel A**
- **Bits 5:4 – COM1B[1:0] : Compare Output Mode for Channel B**

The COM1A[1:0] and COM1B[1:0] control the Output Compare pins (OC1A and OC1B respectively) behavior. If one or both of the COM1A[1:0] bits are written to one, the OC1A output overrides the normal port functionality of the I/O pin it is connected to. If one or both of the COM1B[1:0] bit are written to one, the OC1B output overrides the normal port functionality of the I/O pin it is connected to. However, note that the Data Direction Register (DDR) bit corresponding to the OC1A or OC1B pin must be set in order to enable the output driver.

When the OC1A or OC1B is connected to the pin, the function of the COM1x[1:0] bits is dependent of the WGM1[3:0] bits setting.

Table 39 shows COM1x[1:0] bit functionality when WGM1[3:0] bits are set to a Normal or a CTC mode (non-PWM).

Table 47. Source Selection for Analog Comparator Negative Input

ACNMUX1	ACNMUX0	Analog Comparator Negative Input
0	0	AIN1 pin
0	1	Output of ADC multiplexer
1	0	Reserved
1	1	

- **Bits 1:0 – ACPMUX[1:0]: Analog Comparator Positive Input Multiplexer**

Together with ACPMUX2, these bits select the source for the positive input of the analog comparator, as shown in Table 48, below.

Table 48. Source Selection for Analog Comparator Positive Input

ACPMUX2	ACPMUX1	ACPMUX0	Analog Comparator Positive Input
0	0	0	AIN0 pin
0	0	1	Reserved
0	1	0	Reserved
0	1	1	Reserved
1	0	0	Internal bandgap reference voltage
1	0	1	Reserved
1	1	0	Reserved
1	1	1	Reserved

14.1.3 DIDR0 – Digital Input Disable Register 0

Bit	7	6	5	4	3	2	1	0	
(0x7E)	ADC7D	ADC6D	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D	DIDR0
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	

- **Bit 2 – ADC2D: ADC2/AIN1 Digital input buffer disable**

When used as an analog input but not required as a digital input the power consumption of the digital input buffer can be reduced by writing this bit to logic one. When this bit is set, the digital input buffer on the AIN1 pin is disabled and the corresponding pin register bit (PA2) will always read zero.

- **Bits 1 – ADC1D: ADC1/AIN0 Digital input buffer disable**

When used as an analog input but not required as a digital input the power consumption of the digital input buffer can be reduced by writing this bit to logic one. When this bit is set, the digital input buffer on the AIN0 pin is disabled and the corresponding pin register bit (PA1) will always read zero.

C Code Example

```
void SPI_MasterInit(void)
{
      /* Set MOSI and SCK output, all others input */
      DDR_SPI = (1<<DD_MOSI)|(1<<DD_SCK);
      /* Enable SPI, Master, set clock rate fck/16 */
      SPCR = (1<<SPE)|(1<<MSTR)|(1<<SPR0);
}

void SPI_MasterTransmit(char cData)
{
      /* Start transmission */
      SPDR = cData;
      /* Wait for transmission complete */
      while(!(SPSR & (1<<SPIF)))
            ;
}
```

Note: See "Code Examples" on page 7.

The following code examples show how to initialize the SPI as a Slave and how to perform a simple reception.

Assembly Code Example

```
SPI_SlaveInit:
      ; Set MISO output, all others input
      ldi               r17,(1<<DD_MISO)
      out               DDR_SPI,r17
      ; Enable SPI
      ldi               r17,(1<<SPE)
      out               SPCR,r17
      ret

SPI_SlaveReceive:
      ; Wait for reception complete
      in                r16, SPSR
      sbrs              r16, SPIF
      rjmp              SPI_SlaveReceive
      ; Read received data and return
      in                r16,SPDR
      ret
```

- **Bit 1 – UCPHA: Clock Phase**

The UCPHA bit setting determine if data is sampled on the leasing edge (first) or tailing (last) edge of XCK. See “SPI Data Modes and Timing” on page 191 for details.

- **Bit 0 – UCPOL: Clock Polarity**

The UCPOL bit sets the polarity of the XCK clock. The combination of the UCPOL and UCPHA bit settings determine the timing of the data transfer. See “SPI Data Modes and Timing” on page 191 for details.

18.8.5 UBRRL and UBRRH – USART MSPIM Baud Rate Registers

The function and bit description of the baud rate registers in MSPI mode is identical to normal USART operation. See “UBRRL and UBRRH – USART Baud Rate Registers” on page 189.

When designing a system where debugWIRE will be used, the following must be observed:

- Pull-Up resistor on the dW/(RESET) line must be in the range of 10k to 20 kΩ. However, the pull-up resistor is optional.
- Connecting the RESET pin directly to V_{CC} will not work.
- Capacitors inserted on the RESET pin must be disconnected when using debugWire.
- All external reset sources must be disconnected.

20.4 Software Break Points

debugWIRE supports Program memory Break Points by the AVR Break instruction. Setting a Break Point in AVR Studio® will insert a BREAK instruction in the Program memory. The instruction replaced by the BREAK instruction will be stored. When program execution is continued, the stored instruction will be executed before continuing from the Program memory. A break can be inserted manually by putting the BREAK instruction in the program.

The Flash must be re-programmed each time a Break Point is changed. This is automatically handled by AVR Studio through the debugWIRE interface. The use of Break Points will therefore reduce the Falsh Data retention. Devices used for debugging purposes should not be shipped to end customers.

20.5 Limitations of debugWIRE

The debugWIRE communication pin (dW) is physically located on the same pin as External Reset (RESET). An External Reset source is therefore not supported when the debugWIRE is enabled.

The debugWIRE system accurately emulates all I/O functions when running at full speed, i.e., when the program in the CPU is running. When the CPU is stopped, care must be taken while accessing some of the I/O Registers via the debugger (AVR Studio). See the debugWIRE documentation for detailed description of the limitations.

The debugWIRE interface is asynchronous, which means that the debugger needs to synchronize to the system clock. If the system clock is changed by software (e.g. by writing CLKPS bits) communication via debugWIRE may fail. Also, clock frequencies below 100kHz may cause communication problems.

A programmed DWEN Fuse enables some parts of the clock system to be running in all sleep modes. This will increase the power consumption while in sleep. Thus, the DWEN Fuse should be disabled when debugWire is not used.

20.6 Register Description

The following section describes the registers used with the debugWire.

20.6.1 DWDR – debugWire Data Register

Bit	7	6	5	4	3	2	1	0	
0x31 (0x51)	DWDR[7:0]								DWDR
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 DWDR Register provides a communication channel from the running program in the MCU to the debugger. This register is only accessible by the debugWIRE and can therefore not be used as a general purpose register in the normal operations.

21.4.1 RWW – Read-While-Write Section

When the boot loader is programming a page inside the RWW section, it is possible to read code from the Flash, but only code that is located in the NRWW section.

During programming, the software must ensure that the RWW section is not read. If the software during programming tries to read code located inside the RWW section (i.e., by a call/jmp/lpm or an interrupt), it may end up in an unknown state. To avoid this, interrupts should either be disabled or moved to the BLS, which is always located in the NRWW section. The RWW Section Busy bit (RWWSB) in the Store Program Memory Control and Status Register (SPMCSR) is set as long as the RWW section is blocked for reading. After a programming is completed, the RWWSB must be cleared by software before reading code located in the RWW section.

21.4.2 NRWW – No Read-While-Write Section

Code located in the NRWW section can be read when the boot loader is updating a page in the RWW section. When the boot loader updates the NRWW section, the CPU is halted during the entire Page Erase or Page Write operation.

21.5 Entering the Boot Loader Program

Entering the boot loader takes place by a jump or call from the application program. This may be initiated by a trigger, such as a command received via USART, or SPI. Alternatively, the Boot Reset fuse (BOOTRST) can be programmed so that the reset vector is pointing to the boot loader start address, in which case the boot loader is started after a reset. See Table 80, below.

Table 80. Boot Reset Fuse

BOOTRST (1)	Reset Vector
1	Application reset address (0x0000)
0	Boot loader (see Table 82 on page 217)

Note: 1. “1” means unprogrammed, “0” means programmed

After the application code has been loaded, the boot program can start executing the application code.

Note that fuses cannot be changed by the MCU itself. This means that once the BOOTRST fuse is programmed, the reset vector will always point to the boot loader and the fuse can only be changed through the serial or parallel programming interface.

21.6 Configuring the Boot Loader

Read-While-Write (RWW) and No Read-While-Write(NRWW) sections of the Flash are constant, as shown in Table 81. For details on these two sections, see “Read-While-Write and No Read-While-Write Flash Sections” on page 214.

Table 81. Read-While-Write and No Read-While-Write Sections of the Flash

Section	Flash Pages	Address
Read-While-Write section (RWW)	96	0x000 - 0xBFF
No Read-While-Write section (NRWW)	32	0xC00 - 0xFFF

The size of application and boot loader sections can be changed, as shown in Table 82 on page 217.

22.3.1 Signature Bytes

All Atmel microcontrollers have a three-byte signature code which identifies the device. This code can be read in both serial and parallel mode, also when the device is locked.

Signature bytes can also be read by the device firmware. See section “Reading Lock, Fuse and Signature Data from Software” on page 229.

The three signature bytes reside in a separate address space called the device signature imprint table. The signature data for ATtiny828 is given in Table 93.

Table 93. Device Signature Bytes

Part	Signature Byte 0	Signature Byte 1	Signature Byte 0
ATtiny828	0x1E	0x93	0x14

22.3.2 Calibration Bytes

The device signature imprint table of ATtiny828 contains calibration data for the internal oscillators, as shown in Table 92 on page 228. During reset, calibration data is automatically copied to the calibration registers (OSCCAL0, OSCCAL1) to ensure correct frequency of the calibrated oscillators. See “OSCCAL0 – Oscillator Calibration Register” on page 32, and “OSCCAL1 – Oscillator Calibration Register” on page 33.

Calibration bytes can also be read by the device firmware. See section “Reading Lock, Fuse and Signature Data from Software” on page 229.

22.4 Reading Lock, Fuse and Signature Data from Software

Fuse and lock bits can be read by device firmware. Programmed fuse and lock bits read zero. unprogrammed as one. See “Lock Bits” on page 225 and “Fuse Bits” on page 226.

In addition, firmware can also read data from the device signature imprint table. See “Device Signature Imprint Table” on page 228.

22.4.1 Lock Bit Read

Lock bit values are returned in the destination register after an LPM instruction has been issued within three CPU cycles after RWFLB and SPMEN bits have been set in SPMCSR (see page 223). The RWFLB and SPMEN bits automatically clear upon completion of reading the lock bits, or if no LPM instruction is executed within three CPU cycles, or if no SPM instruction is executed within four CPU cycles. When RWFLB and SPMEN are cleared LPM functions normally.

To read the lock bits, follow the below procedure:

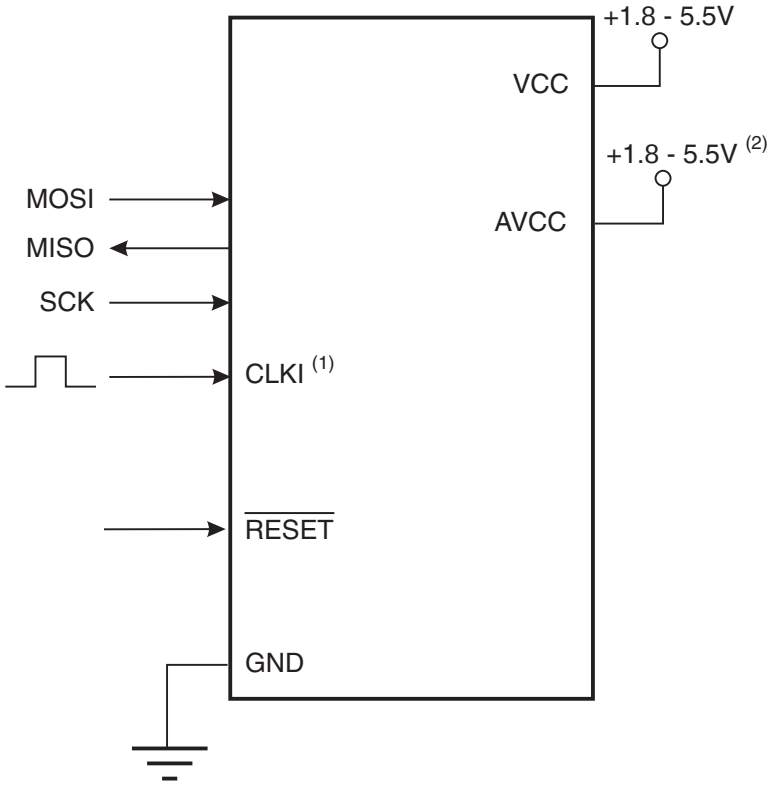
1. Load the Z-pointer with 0x0001.
2. Set RWFLB and SPMEN bits in SPMCSR.
3. Issue an LPM instruction within three clock cycles.
4. Read the lock bits from the LPM destination register.

If successful, the contents of the destination register are as follows.

Bit	7	6	5	4	3	2	1	0
Rd	–	–	BLB12	BLB11	BLB02	BLB01	LB2	LB1

See section “Lock Bits” on page 225 for more information.

Figure 99. Serial Programming Signals



Notes: 1. If the device is clocked by the internal oscillator there is no need to connect a clock source to the CLKI pin.
2. V_{CC} - 0.3V < AV_{CC} < V_{CC} + 0.3V, however, AV_{CC} should always be within 1.7 – 5.5V.

When programming the EEPROM, an auto-erase cycle is built into the self-timed programming operation and there is no need to first execute the Chip Erase instruction. This applies for serial programming mode, only.

The Chip Erase operation turns the content of every memory location in Flash and EEPROM arrays into 0xFF.

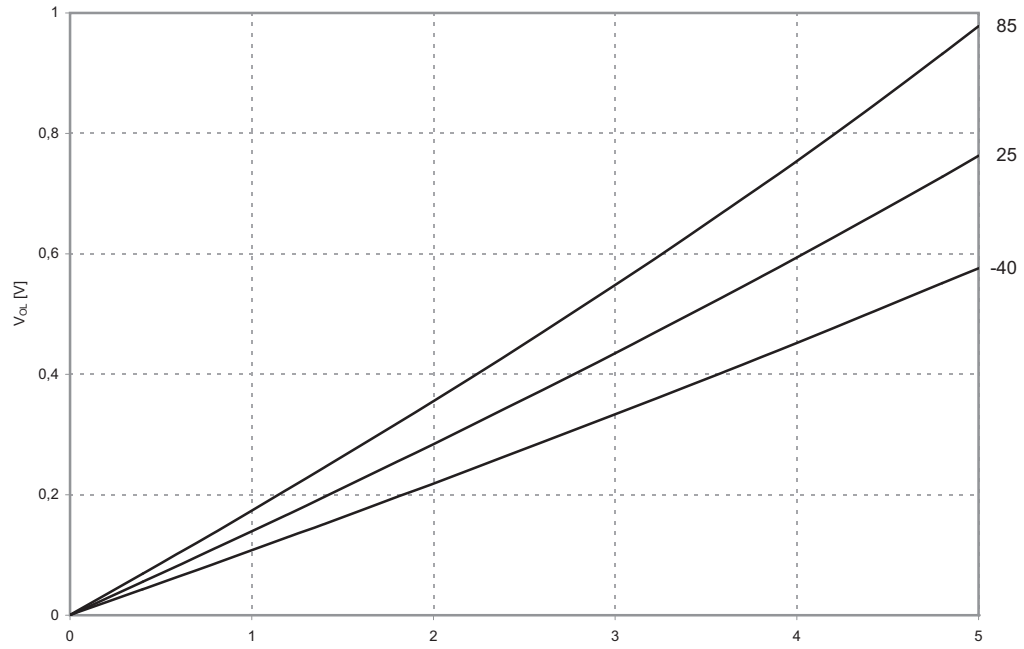
Depending on CKSEL fuses, a valid clock must be present. The minimum low and high periods for the serial clock (SCK) input are defined as follows:

- Minimum low period of serial clock:
  - When f_{ck} < 12MHz: > 2 CPU clock cycles
  - When f_{ck} >= 12MHz: 3 CPU clock cycles
- Minimum high period of serial clock:
  - When f_{ck} < 12MHz: > 2 CPU clock cycles
  - When f_{ck} >= 12MHz: 3 CPU clock cycles

23.3.1 Pin Mapping

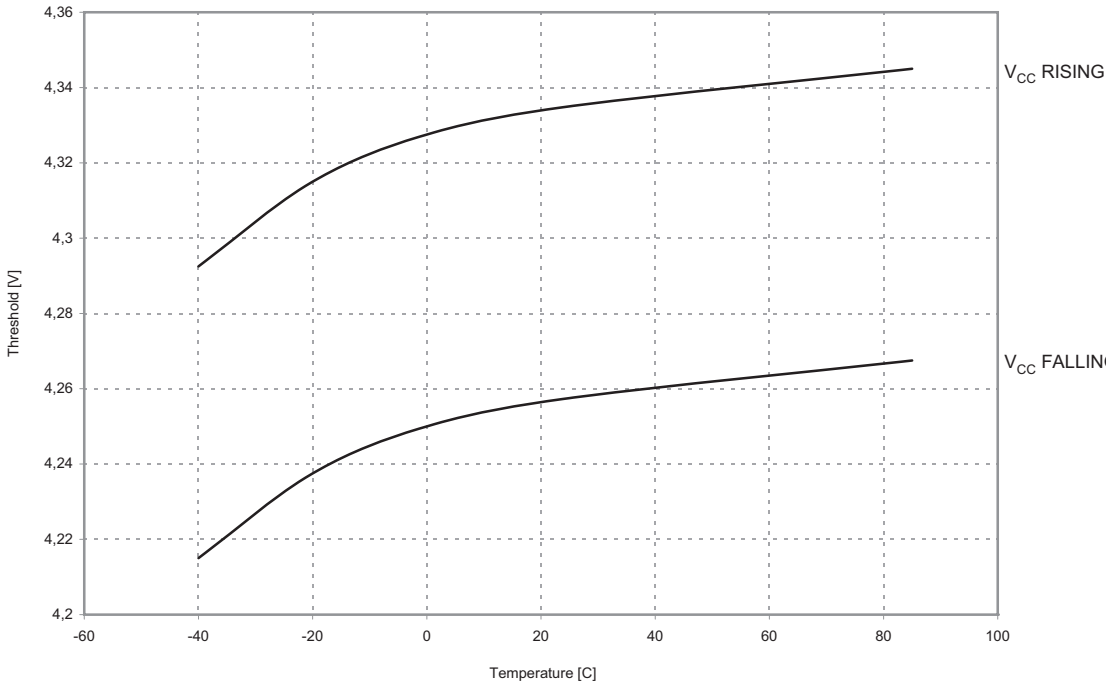
The pin mapping is listed in Table 100 on page 243. Note that not all parts use the SPI pins dedicated for the internal SPI interface.

Figure 164. V_{OL}: Output Voltage vs. Sink Current (Reset Pin as I/O, V_{CC} = 5V)



25.10 BOD

Figure 165. BOD Threshold vs Temperature (BODLEVEL = 4.3V)

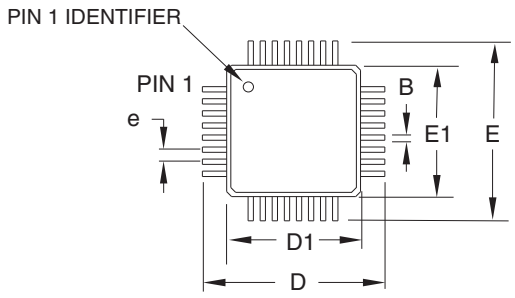


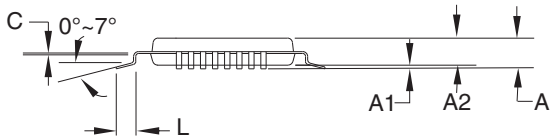
26. Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page(s)
(0xFF)	Reserved	–	–	–	–	–	–	–	–	
(0xFE)	Reserved	–	–	–	–	–	–	–	–	
(0xFD)	Reserved	–	–	–	–	–	–	–	–	
(0xFC)	Reserved	–	–	–	–	–	–	–	–	
(0xFB)	Reserved	–	–	–	–	–	–	–	–	
(0xFA)	Reserved	–	–	–	–	–	–	–	–	
(0xF9)	Reserved	–	–	–	–	–	–	–	–	
(0xF8)	Reserved	–	–	–	–	–	–	–	–	
(0xF7)	Reserved	–	–	–	–	–	–	–	–	
(0xF6)	Reserved	–	–	–	–	–	–	–	–	
(0xF5)	Reserved	–	–	–	–	–	–	–	–	
(0xF4)	Reserved	–	–	–	–	–	–	–	–	
(0xF3)	Reserved	–	–	–	–	–	–	–	–	
(0xF2)	Reserved	–	–	–	–	–	–	–	–	
(0xF1)	OSCTCAL0B	Oscillator Temperature Compensation Register B								Page 33
(0xF0)	OSCTCAL0A	Oscillator Temperature Compensation Register A								Page 33
(0xEF)	Reserved	–	–	–	–	–	–	–	–	
(0xEE)	Reserved	–	–	–	–	–	–	–	–	
(0xED)	Reserved	–	–	–	–	–	–	–	–	
(0xEC)	Reserved	–	–	–	–	–	–	–	–	
(0xEB)	Reserved	–	–	–	–	–	–	–	–	
(0xEA)	Reserved	–	–	–	–	–	–	–	–	
(0xE9)	TOCPMSA1	TOCC7S1	TOCC7S0	TOCC6S1	TOCC6S0	TOCC5S1	TOCC5S0	TOCC4S1	TOCC4S0	Page 127
(0xE8)	TOCPMSA0	TOCC3S1	TOCC3S0	TOCC2S1	TOCC2S0	TOCC1S1	TOCC1S0	TOCC0S1	TOCC0S0	Page 127
(0xE7)	Reserved	–	–	–	–	–	–	–	–	
(0xE6)	Reserved	–	–	–	–	–	–	–	–	
(0xE5)	Reserved	–	–	–	–	–	–	–	–	
(0xE4)	Reserved	–	–	–	–	–	–	–	–	
(0xE3)	Reserved	–	–	–	–	–	–	–	–	
(0xE2)	TOCPMCOE	TOCC7OE	TOCC6OE	TOCC5OE	TOCC4OE	TOCC3OE	TOCC2OE	TOCC1OE	TOCC0OE	Page 128
(0xE1)	Reserved	–	–	–	–	–	–	–	–	
(0xE0)	Reserved	–	–	–	–	–	–	–	–	
(0xDF)	DIDR3	–	–	–	–	ADC27D	ADC26D	ADC25D	ADC24D	Page 154
(0xDE)	DIDR2	ADC23D	ADC22D	ADC21D	ADC20D	ADC19D	ADC18D	ADC17D	ADC16D	Page 154
(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)	Reserved	–	–	–	–	–	–	–	–	
(0xD0)	Reserved	–	–	–	–	–	–	–	–	
(0xCF)	Reserved	–	–	–	–	–	–	–	–	
(0xCE)	Reserved	–	–	–	–	–	–	–	–	
(0xCD)	Reserved	–	–	–	–	–	–	–	–	
(0xCC)	Reserved	–	–	–	–	–	–	–	–	
(0xCB)	Reserved	–	–	–	–	–	–	–	–	
(0xCA)	Reserved	–	–	–	–	–	–	–	–	
(0xC9)	Reserved	–	–	–	–	–	–	–	–	
(0xC8)	Reserved	–	–	–	–	–	–	–	–	
(0xC7)	Reserved	–	–	–	–	–	–	–	–	
(0xC6)	UDR	USART Data Register								Pages 184, 195
(0xC5)	UBRRH	–	–	–	–	USART Baud Register High				Page 189, 198
(0xC4)	UBRRL	USART Baud Rate Register Low								Page 189, 198
(0xC3)	UCSRD	RXSIE	RXS	SFDE	–	–	–	–	–	Page 188
(0xC2)	UCSRC	UMSEL1	UMSEL0	UPM1	UPM0	USBS	UCSZ1/UDO	UCSZ0/UCP	UCPOL	Page 186, 197
(0xC1)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	Page 185, 196
(0xC0)	UCSRA	RXC	TXC	UDRE	FE	DOR	UPE	U2X	MPCM	Page 184, 196
(0xBF)	Reserved	–	–	–	–	–	–	–	–	
(0xBE)	Reserved	–	–	–	–	–	–	–	–	

29. Packaging Information

29.1 32A





COMMON DIMENSIONS
(Unit of measure = mm)

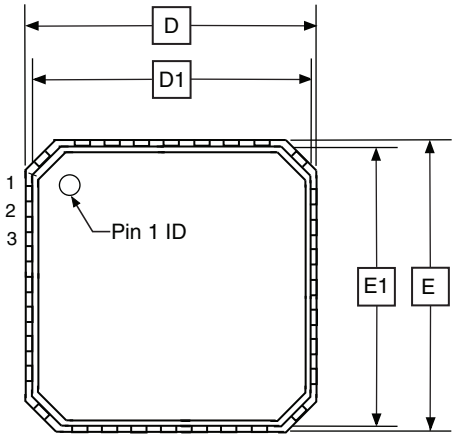
SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.20	
A1	0.05	–	0.15	
A2	0.95	1.00	1.05	
D	8.75	9.00	9.25	
D1	6.90	7.00	7.10	Note 2
E	8.75	9.00	9.25	
E1	6.90	7.00	7.10	Note 2
B	0.30	–	0.45	
C	0.09	–	0.20	
L	0.45	–	0.75	
e	0.80 TYP			

Notes:
1. This package conforms to JEDEC reference MS-026, Variation ABA.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10mm maximum.

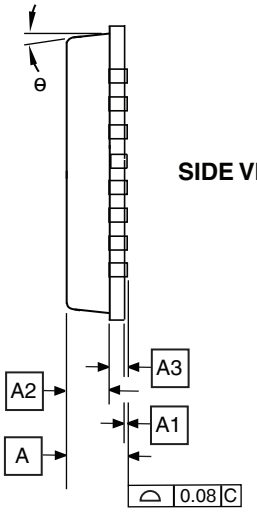
2010-10-20

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**32A,** 32-lead, 7 x 7mm body size, 1.0mm body thickness, 0.8mm lead pitch, thin profile plastic quad flat package (TQFP)	32A	C

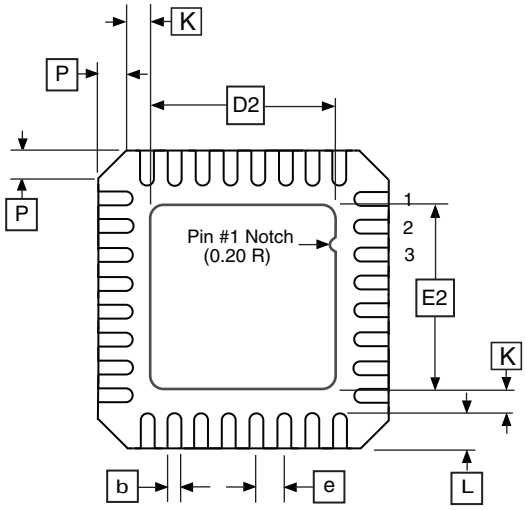
29.2 32M1-A



TOP VIEW



SIDE VIEW



BOTTOM VIEW

Note: JEDEC Standard MO-220, Fig. 2 (Anvil Singulation), VHHD-2.

COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	0.80	0.90	1.00	
A1	–	0.02	0.05	
A2	–	0.65	1.00	
A3		0.20 REF		
b	0.18	0.23	0.30	
D	4.90	5.00	5.10	
D1	4.70	4.75	4.80	
D2	2.95	3.10	3.25	
E	4.90	5.00	5.10	
E1	4.70	4.75	4.80	
E2	2.95	3.10	3.25	
e		0.50 BSC		
L	0.30	0.40	0.50	
P	–	–	0.60	
θ	–	–	12^{o}	
K	0.20	–	–	

5/25/06

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**32M1-A**, 32-pad, 5 x 5 x 1.0mm Body, Lead Pitch 0.50mm, 3.10mm Exposed Pad, Micro Lead Frame Package (MLF)	32M1-A	E