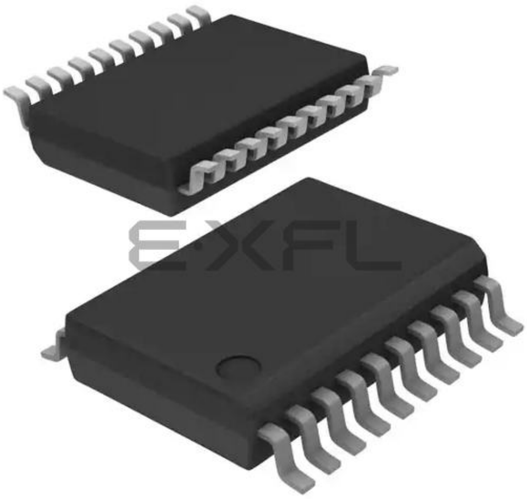
Microchip Technology - ATTINY167-XUR Datasheet



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"

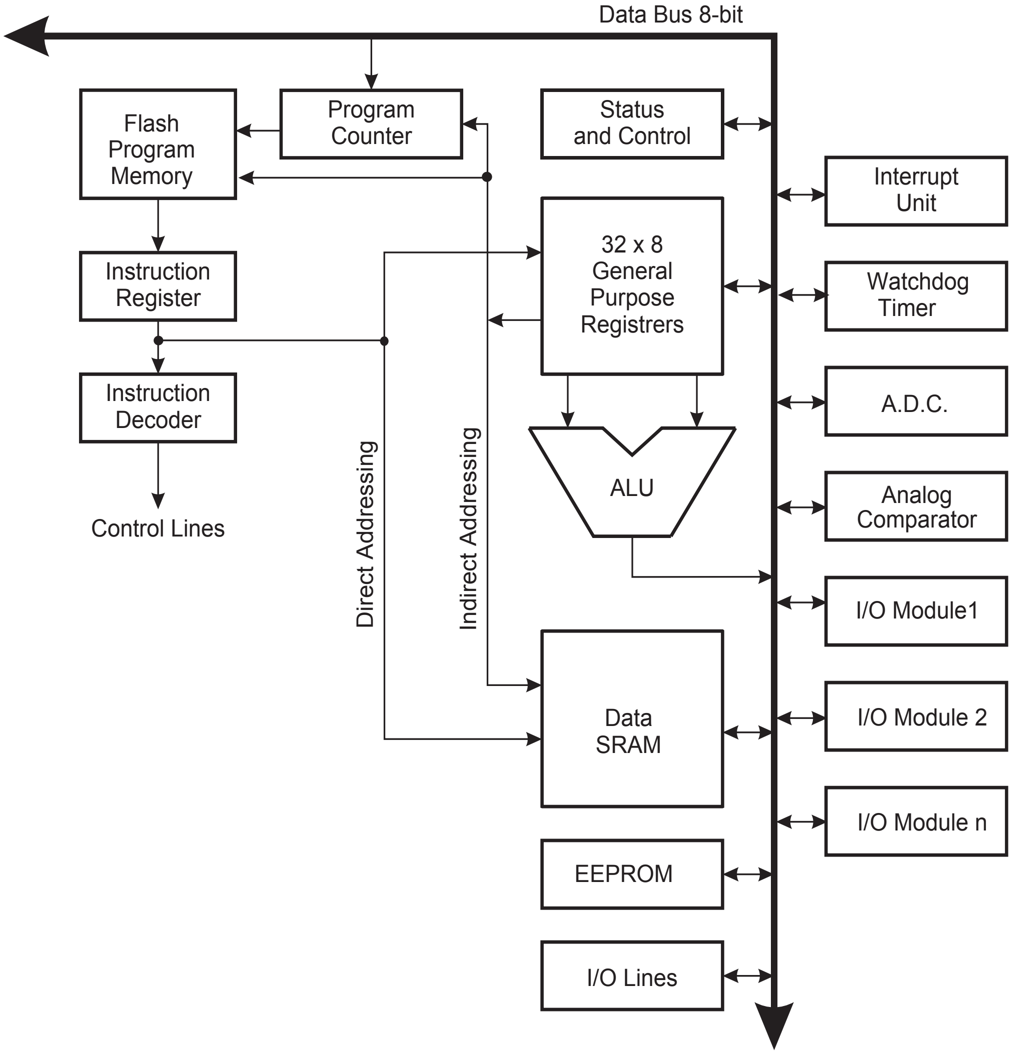
Details	
Product Status	Active
Core Processor	AVR
Core Size	8-Bit
Speed	16MHz
Connectivity	I²C, LINbus, SPI, UART/USART, USI
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	16
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 11x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	20-TSSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/attiny167-xur

2. AVR CPU Core

2.1 Overview

This section discusses the AVR core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

Figure 2-1. Block Diagram of the AVR Architecture



In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the Program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the Program memory. This concept enables instructions to be executed in every clock cycle. The Program memory is In-System Reprogrammable Flash memory. The fast-access Register File contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical 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.

3.4 I/O Memory

The I/O space definition of the ATtiny87/167 is shown in Section 24. "Register Summary" on page 270.

All ATtiny87/167 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 I/O addresses 0x00 - 0x3F must be used. When addressing I/O Registers as data space using LD and ST instructions, 0x20 must be added to these addresses. The ATtiny87/167 is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.

Some of the Status Flags are cleared by writing a logical one to them. Note that, unlike most other AVRs, the CBI and SBI instructions will only operate on the specified bit, and can therefore be used on registers containing such Status Flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.

The I/O and Peripherals Control Registers are explained in later sections.

3.4.1 General Purpose I/O Registers

The ATtiny87/167 contains three General Purpose I/O Registers. These registers can be used for storing any information, and they are particularly useful for storing global variables and Status Flags.

The General Purpose I/O Registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI, CBI, SBIS, and SBIC instructions.

3.5 Register Description

3.5.1 EEARH and EEARL – EEPROM Address Register

Bit	7	6	5	4	3	2	1	0	
0x22 (0x42)	–	–	–	–	–	–	–	EEAR8	EEARH
0x21 (0x41)	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	EEARL
Bit	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R/W	
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	X	
Initial Value	X	X	X	X	X	X	X	X	

- **Bits 7:1 – Res: Reserved Bits**

These bits are reserved for future use and will always read as 0 in ATtiny87/167.

- **Bits 8:0 – EEAR[8:0]: EEPROM Address**

The EEPROM Address Registers – EEARH and EEARL – specifies the high EEPROM address in the EEPROM space (see "E2 size" in Table 3-1 on page 15). The EEPROM data bytes are

Here is a “light” C-code that describes such a sequence of commands.

C Code Example

```c
void ClockSwiching (unsigned char clk_number, unsigned char sut) {

#define CLOCK_RECOVER   0x05
#define CLOCK_ENABLE    0x02
#define CLOCK_SWITCH    0x04
#define CLOCK_DISABLE   0x01

unsigned char previous_clk, temp;

  // Disable interrupts
   temp = SREG; asm ("cli");
  // Save the current system clock source
   CLKCSR = 1 << CLKCCE;
   CLKCSR = CLOCK_RECOVER;
   previous_clk = CLKSELR & 0x0F;
  // Enable the new clock source
   CLKSELR = ((sut << 4 ) & 0x30) | (clk_number & 0x0F);
   CLKCSR = 1 << CLKCCE;
   CLKCSR = CLOCK_ENABLE;
  // Wait for clock validity
   while ((CLKCSR & (1 << CLKRDY)) == 0);
  // Switch clock source
   CLKCSR = 1 << CLKCCE;
   CLKCSR = CLOCK_SWITCH;
  // Wait for effective switching
   while (1){
       CLKCSR = 1 << CLKCCE;
       CLKCSR = CLOCK_RECOVER;
       if ((CLKSELR & 0x0F) == (clk_number & 0x0F)) break;
   }
  // Shut down unneeded clock source
   if (previous_clk != (clk_number & 0x0F)) {
       CLKSELR = previous_clk;
       CLKCSR = 1 << CLKCCE;
       CLKCSR = CLOCK_DISABLE;
   }
  // Re-enable interrupts
   SREG = temp;
}
```

Warning:

In the ATtiny87/167, only one among the three external clock sources can be enabled at a given time. Moreover, the enables of the external clock and of the external low-frequency oscillator are shared with the asynchronous timer.

4.3.8 Clock Monitoring

A safe system needs to monitor its clock sources. Two domains need to be monitored:
- Clock sources for peripherals,
- Clocks sources for system clock generation.

In the first domain, the user (code) can easily check the validity of the clock(s) (See “COUT Command” on page 33.).

TOSC1 cycle, the interrupt will not occur, and the device will fail to wake up. If the user is in doubt whether the time before re-entering Power-save mode is sufficient, the following algorithm can be used to ensure that one TOSC1 cycle has elapsed:

a. Write a value to TCCR0A, TCNT0, or OCR0A.

b. Wait until the corresponding Update Busy flag in ASSR returns to zero.

c. Enter Power-save or ADC Noise Reduction mode.

- When the asynchronous operation is selected, the oscillator for Timer/Counter0 is always running, except in Power-down mode. After a Power-up Reset or wake-up from Power-down mode, the user should be aware of the fact that this oscillator might take as long as one second to stabilize. The user is advised to wait for at least one second before using Timer/Counter0 after power-up or wake-up from Power-down mode. The contents of all Timer/Counter0 Registers must be considered lost after a wake-up from Power-down mode due to unstable clock signal upon start-up, no matter whether the oscillator is in use or a clock signal is applied to the XTAL1 pin.
- Description of wake up from Power-save mode when the timer is clocked asynchronously: When the interrupt condition is met, the wake up process is started on the following cycle of the timer clock, that is, the timer is always advanced by at least one before the processor can read the counter value. After wake-up, the MCU is halted for four cycles, it executes the interrupt routine, and resumes execution from the instruction following SLEEP.
- Reading of the TCNT0 Register shortly after wake-up from Power-save may give an incorrect result. Since TCNT0 is clocked on the asynchronous clock, reading TCNT0 must be done through a register synchronized to the internal I/O clock domain (CPU main clock). Synchronization takes place for every rising XTAL1 edge. When waking up from Power-save mode, and the I/O clock ($clk_{I/O}$) again becomes active, TCNT0 will read as the previous value (before entering sleep) until the next rising XTAL1 edge. The phase of the XTAL1 clock after waking up from Power-save mode is essentially unpredictable, as it depends on the wake-up time. The recommended procedure for reading TCNT0 is thus as follows:

  a. Write any value to either of the registers OCR0A or TCCR0A.

  b. Wait for the corresponding Update Busy Flag to be cleared.

  c. Read TCNT0.
- During asynchronous operation, the synchronization of the interrupt flags for the asynchronous timer takes 3 processor cycles plus one timer cycle. The timer is therefore advanced by at least one before the processor can read the timer value causing the setting of the interrupt flag. The Output Compare pin is changed on the timer clock and is not synchronized to the processor clock.

11.2 Register Description

11.2.1 GTCCR – General Timer/Counter Control Register

Bit	7	6	5	4	3	2	1	0	
0x23 (0x43)	**TSM**	–	–	–	–	–	**PSR0**	**PSR1**	**GTCCR**
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – TSM: Timer/Counter Synchronization Mode**

Writing the TSM bit to one activates the Timer/Counter Synchronization mode. In this mode, the value that is written to the PSR0 and PSR1 bits is kept, hence keeping the corresponding prescaler reset signals asserted. This ensures that the corresponding Timer/Counters are halted and can be configured to the same value without the risk of one of them advancing during configuration. When the TSM bit is written to zero, the PSR0 and PSR1 bits are cleared by hardware, and the Timer/Counters start counting simultaneously.

- **Bit 0 – PSR1: Prescaler Reset Timer/Counter1**

When this bit is one, Timer/Counter1 prescaler will be reset. This bit is normally cleared immediately by hardware, except if the TSM bit is set.

12.11.10 TIFR1 – Timer/Counter1 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x16 (0x36)	–	–	ICF1	–	–	OCF1B	OCF1A	TOV1	TIFR1
Read/Write	R	R	R/W	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:6 – Res: Reserved Bits**

These bits are reserved for future use.

- **Bit 5 – ICF1: Input Capture Flag**

This flag is set when a capture event occurs on the ICP1 pin. When the Input Capture Register (ICR1) is set by the WGM1[3:0] to be used as the TOP value, the ICF1 flag is set when the counter reaches the TOP value.

ICF1 is automatically cleared when the Input Capture Interrupt Vector is executed. Alternatively, ICF1 can be cleared by writing a logic one to its bit location.

- **Bits 4:3 – Res: Reserved Bits**

These bits are reserved for future use.

- **Bit 2 – OCF1B: Output Compare B Match Flag**

This flag is set in the timer clock cycle after the counter (TCNT1) value matches the Output Compare Register B (OCR1B).

Note that a Forced Output Compare (FOC1B) strobe will not set the OCF1B flag.

OCF1B is automatically cleared when the Output Compare Match B Interrupt Vector is executed. Alternatively, OCF1B can be cleared by writing a logic one to its bit location.

- **Bit 1 – OCF1A: Output Compare A Match Flag**

This flag is set in the timer clock cycle after the counter (TCNT1) value matches the Output Compare Register A (OCR1A).

Note that a Forced Output Compare (FOC1A) strobe will not set the OCF1A flag.

OCF1A is automatically cleared when the Output Compare Match A Interrupt Vector is executed. Alternatively, OCF1A can be cleared by writing a logic one to its bit location.

- **Bit 0 – TOV1: Timer/Counter Overflow Flag**

The setting of this flag is dependent of the WGM1[3:0] bits setting. In Normal and CTC modes, the TOV1 flag is set when the timer overflows. Refer to Table 12-4 on page 133 for the TOV1 flag behavior when using another WGM1[3:0] bit setting.

TOV1 is automatically cleared when the Timer/Counter1 Overflow Interrupt Vector is executed. Alternatively, TOV1 can be cleared by writing a logic one to its bit location.

14.4 Alternative USI Usage

When the USI unit is not used for serial communication, it can be set up to do alternative tasks due to its flexible design.

14.4.1 Half-duplex Asynchronous Data Transfer

By utilizing the USI Data Register in Three-wire mode, it is possible to implement a more compact and higher performance UART than by software only.

14.4.2 4-bit Counter

The 4-bit counter can be used as a stand-alone counter with overflow interrupt. Note that if the counter is clocked externally, both clock edges will generate an increment.

14.4.3 12-bit Timer/Counter

Combining the USI 4-bit counter and Timer/Counter0 allows them to be used as a 12-bit counter.

14.4.4 Edge Triggered External Interrupt

By setting the counter to maximum value (F) it can function as an additional external interrupt. The Overflow Flag and Interrupt Enable bit are then used for the external interrupt. This feature is selected by the USICS1 bit.

14.4.5 Software Interrupt

The counter overflow interrupt can be used as a software interrupt triggered by a clock strobe.

14.5 Register Description

14.5.1 USIDR – USI Data Register

Bit	7	6	5	4	3	2	1	0	
(0xBA)	USID7	USID6	USID5	USID4	USID3	USID2	USID1	USID0	USIDR
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	

- **Bits 7:0 – USID[7:0]: USI Data**

When accessing the USI Data Register (USIDR) the Serial Register can be accessed directly. If a serial clock occurs at the same cycle the register is written, the register will contain the value written and no shift is performed. A (left) shift operation is performed depending of the USICS[1:0] bits setting. The shift operation can be controlled by an external clock edge, by a Timer/Counter0 Compare Match, or directly by software using the USICLK strobe bit. Note that even when no wire mode is selected (USIWM[1:0] = 0) both the external data input (DI/SDA) and the external clock input (USCK/SCL) can still be used by the USI Data Register.

The output pin in use, DO or SDA depending on the wire mode, is connected via the output latch to the most significant bit (bit 7) of the Data Register. The output latch is open (transparent) during the first half of a serial clock cycle when an external clock source is selected (USICS1 = 1), and constantly open when an internal clock source is used (USICS1 = 0). The output will be changed immediately when a new MSB written as long as the latch is open. The latch ensures that data input is sampled and data output is changed on opposite clock edges.

15.3.3 Data Transport

Two types of data may be transported in a frame; signals or diagnostic messages.

- Signals
  Signals are scalar values or byte arrays that are packed into the data field of a frame. A signal is always present at the same position in the data field for all frames with the same identifier.
- Diagnostic messages
  Diagnostic messages are transported in frames with two reserved identifiers. The interpretation of the data field depends on the data field itself as well as the state of the communicating nodes.

15.3.4 Schedule Table

The master task (in the master node) transmits frame headers based on a schedule table. The schedule table specifies the identifiers for each header and the interval between the start of a frame and the start of the following frame. The master application may use different schedule tables and select among them.

15.3.5 Compatibility with LIN 1.3

LIN 2.1 is a super-set of LIN 1.3.

A LIN 2.1 master node can handle clusters consisting of both LIN 1.3 slaves and/or LIN 2.1 slaves. The master will then avoid requesting the new LIN 2.1 features from a LIN 1.3 slave:

- Enhanced checksum,
- Re-configuration and diagnostics,
- Automatic baud rate detection,
- "Response error" status monitoring.

LIN 2.1 slave nodes can not operate with a LIN 1.3 master node (e.g. the LIN1.3 master does not support the enhanced checksum).

The LIN 2.1 physical layer is backwards compatible with the LIN1.3 physical layer. But not the other way around. The LIN 2.1 physical layer sets greater requirements, i.e. a master node using the LIN 2.1 physical layer can operate in a LIN 1.3 cluster.

15.4 LIN / UART Controller

The LIN/UART controller is divided in three main functions:

- Tx LIN Header function,
- Rx LIN Header function,
- LIN Response function.

These functions mainly use two services:

- Rx service,
- Tx service.

Because these two services are basically UART services, the controller is also able to switch into an UART function.

$$\text{HECKSUM} = 255 - \left(\text{unsigned char}\left(\sum_{0}^{n}\text{DATA}_n\right) + \text{unsigned char}\left(\left(\sum_{0}^{n}\text{DATA}_n\right) \gg 8\right)\right)$$

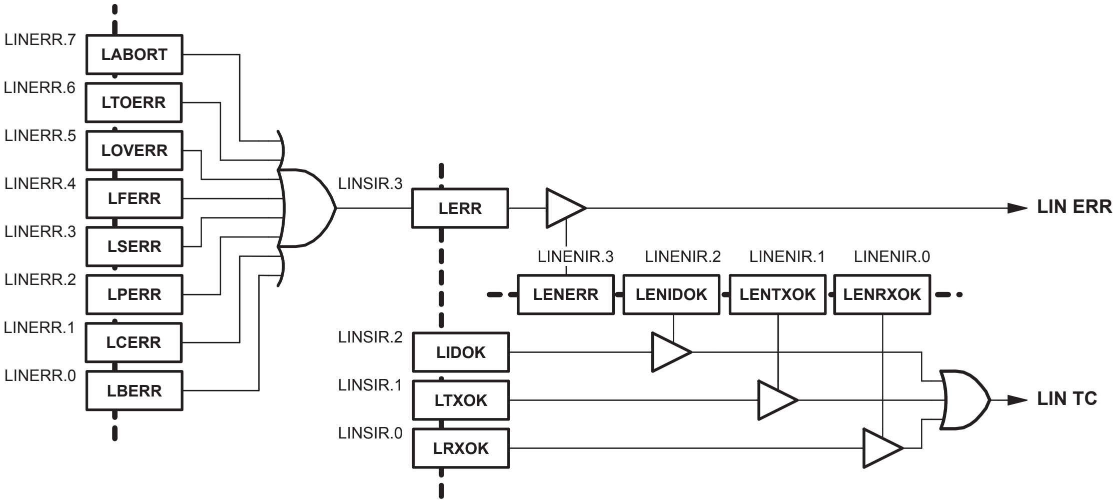
Frame identifiers 60 (0x3C) to 61 (0x3D) shall always use classic checksum.

15.5.13 Interrupts

As shown in Figure 15-13 on page 177, the four communication flags of the LINSIR register are combined to drive two interrupts. Each of these flags have their respective enable interrupt bit in LINENIR register.

(see Section 15.5.8 “xxOK Flags” on page 174 and Section 15.5.9 “xxERR Flags” on page 175).

Figure 15-13. LIN Interrupt Mapping



15.5.14 Message Filtering

Message filtering based upon the whole identifier is not implemented. Only a status for frame headers having 0x3C, 0x3D, 0x3E and 0x3F as identifier is available in the LINSIR register.

Table 15-4. Frame Status

LIDST[2:0]	Frame Status
$0xx_b$	No specific identifier
100_b	60 (0x3C) identifier
101_b	61 (0x3D) identifier
110_b	62 (0x3E) identifier
111_b	63 (0x3F) identifier

15.6.2 LINSIR – LIN Status and Interrupt Register

Bit	7	6	5	4	3	2	1	0	
(0xC9)	LIDST2	LIDST1	LIDST0	LBUSY	LERR	LIDOK	LTXOK	LRXOK	LINSIR
Read/Write	R	R	R	R	R/Wone	R/Wone	R/Wone	R/Wone	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:5 – LIDST[2:0]: Identifier Status**

  – 0xx = no specific identifier,

  – 100 = Identifier 60 (0x3C),

  – 101 = Identifier 61 (0x3D),

  – 110 = Identifier 62 (0x3E),

  – 111 = Identifier 63 (0x3F).

- **Bit 4 – LBUSY: Busy Signal**

  – 0 = Not busy,

  – 1 = Busy (receiving or transmitting).

- **Bit 3 – LERR: Error Interrupt**

  It is a logical OR of LINERR register bits. This bit generates an interrupt if its respective enable bit - LENERR - is set in LINENIR.

  – 0 = No error,

  – 1 = An error has occurred.

  The user clears this bit by writing 1 in order to reset this interrupt. Resetting LERR also resets all LINERR bits.

  In UART mode, this bit is also cleared by reading LINDAT.

- **Bit 2 – LIDOK: Identifier Interrupt**

  This bit generates an interrupt if its respective enable bit - LENIDOK - is set in LINENIR.

  – 0 = No identifier,

  – 1 = Slave task: Identifier present, master task: Tx Header complete.

  The user clears this bit by writing 1, in order to reset this interrupt.

- **Bit 1 – LTXOK: Transmit Performed Interrupt**

  This bit generates an interrupt if its respective enable bit - LENTXOK - is set in LINENIR.

  – 0 = No Tx,

  – 1 = Tx Response complete.

  The user clears this bit by writing 1, in order to reset this interrupt.

  In UART mode, this bit is also cleared by writing LINDAT.

- **Bit 0 – LRXOK: Receive Performed Interrupt**

  This bit generates an interrupt if its respective enable bit - LENRXOK - is set in LINENIR.

  – 0 = No Rx

  – 1 = Rx Response complete.

  The user clears this bit by writing 1, in order to reset this interrupt.

  In UART mode, this bit is also cleared by reading LINDAT.

15.6.3 LINENIR – LIN Enable Interrupt Register

Bit	7	6	5	4	3	2	1	0	
(0xCA)	–	–	–	–	LENERR	LENIDOK	LENTXOK	LENRXOK	LINENIR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:4 – Reserved Bits**

  These bits are reserved for future use. For compatibility with future devices, they must be written to zero when LINENIR is written.

- **Bit 3 – LENERR: Enable Error Interrupt**

  – 0 = Error interrupt masked,

  – 1 = Error interrupt enabled.

- **Bit 2 – LENIDOK: Enable Identifier Interrupt**

  – 0 = Identifier interrupt masked,

  – 1 = Identifier interrupt enabled.

- **Bit 1 – LENTXOK: Enable Transmit Performed Interrupt**

  – 0 = Transmit performed interrupt masked,

  – 1 = Transmit performed interrupt enabled.

- **Bit 0 – LENRXOK: Enable Receive Performed Interrupt**

  – 0 = Receive performed interrupt masked,

  – 1 = Receive performed interrupt enabled.

15.6.4 LINERR – LIN Error Register

Bit	7	6	5	4	3	2	1	0	
(0xCB)	LABORT	LTOERR	LOVERR	LFERR	LSERR	LPERR	LCERR	LBERR	LINERR
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – LABORT: Abort Flag**

  – 0 = No warning,

  – 1 = LIN abort command occurred.

  This bit is cleared when LERR bit in LINSIR is cleared.

- **Bit 6 – LTOERR: Frame_Time_Out Error Flag**

– 0 = No error,

– 1 = Frame_Time_Out error.

This bit is cleared when LERR bit in LINSIR is cleared.

- **Bit 5 – LOVERR: Overrun Error Flag**

– 0 = No error,

– 1 = Overrun error.

This bit is cleared when LERR bit in LINSIR is cleared.

- **Bit 4 – LFERR: Framing Error Flag**

– 0 = No error,

– 1 = Framing error.

This bit is cleared when LERR bit in LINSIR is cleared.

- **Bit 3 – LSERR: Synchronization Error Flag**

– 0 = No error,

– 1 = Synchronization error.

This bit is cleared when LERR bit in LINSIR is cleared.

- **Bit 2 – LPERR: Parity Error Flag**

– 0 = No error,

– 1 = Parity error.

This bit is cleared when LERR bit in LINSIR is cleared.

- **Bit 1 – LCERR: Checksum Error Flag**

– 0 = No error,

– 1 = Checksum error.

This bit is cleared when LERR bit in LINSIR is cleared.

- **Bit 0 – LBERR: Bit Error Flag**

– 0 = no error,

– 1 = Bit error.

This bit is cleared when LERR bit in LINSIR is cleared.

15.6.5 LINBTR – LIN Bit Timing Register

Bit	7	6	5	4	3	2	1	0	
(0xCC)	LDISR	–	LBT5	LBT4	LBT3	LBT2	LBT1	LBT0	LINBTR
Read/Write	R/W	R	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	
Initial Value	0	0	1	0	0	0	0	0	

- **Bit 7 – LDISR: Disable Bit Timing Re synchronization**

– 0 = Bit timing re-synchronization enabled (default),

– 1 = Bit timing re-synchronization disabled.

- **Bits 5:0 – LBT[5:0]: LIN Bit Timing**

18.2 Register Description

18.2.1 ADCSRB – ADC Control and Status Register B

Bit	7	6	5	4	3	2	1	0	
(0x7B)	BIN	ACME	ACIR1	ACIR0	–	ADTS2	ADTS1	ADTS0	ADCSRB
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 6 – ACME: Analog Comparator Multiplexer Enable**

When this bit is written logic one and the ADC is switched off (ADEN in ADCSRA is zero), the ADC multiplexer selects the positive input to the Analog Comparator. When this bit is written logic zero, AIN1 is applied to the positive input of the Analog Comparator.

When the Analog to Digital Converter (ADC) is configured as single ended input channel, it is possible to select any of the ADC[10:0] pins to replace the positive input to the Analog Comparator. The ADC multiplexer (MUX[4:0]) is used to select this input, and consequently, the ADC must be switched off to utilize this feature.

- **Bits 5:4 – ACIR[1:0]: Analog Comparator Internal Voltage Reference Select**

When ACIRS bit is set in ADCSRA register, these bits select a voltage reference for the negative input to the Analog Comparator, see Table 18-2 on page 211.

18.2.2 ACSR – Analog Comparator Control and Status Register

Bit	7	6	5	4	3	2	1	0	
0x30 (0x50)	ACD	ACIRS	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	ACSR
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	N/A	0	0	0	0	0	

- **Bit 7 – ACD: Analog Comparator Disable**

When this bit is written logic one, the power to the Analog Comparator is switched off. This bit can be set at any time to turn off the Analog Comparator. This will reduce power consumption in Active and Idle mode. When changing the ACD bit, the Analog Comparator Interrupt must be disabled by clearing the ACIE bit of ACSR register. Otherwise an interrupt can occur when the bit is changed.

- **Bit 6 – ACIRS: Analog Comparator Internal Reference Select**

When this bit is set an Internal Reference Voltage replaces the negative input to the Analog Comparator (c.f. Table 18-2 on page 211).
If ACIRS is cleared, AIN0 is applied to the negative input to the Analog Comparator.

- **Bit 5 – ACO: Analog Comparator Output**

The output of the Analog Comparator is synchronized and then directly connected to ACO. The synchronization introduces a delay of 1 - 2 clock cycles.

- **Bit 4 – ACI: Analog Comparator Interrupt Flag**

This bit is set by hardware when a comparator output event triggers the interrupt mode defined by ACIS1 and ACIS0. The Analog Comparator interrupt routine is executed if the ACIE bit is set and the I-bit in SREG is set. ACI is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ACI is cleared by writing a logic one to the flag.

21. Memory Programming

21.1 Program and Data Memory Lock Bits

The ATtiny87/167 provides two Lock bits which can be left unprogrammed ("1") or can be programmed ("0") to obtain the additional features listed in Table 21-2. The Lock bits can only be erased to "1" with the Chip Erase command. The ATtiny87/167 has no separate Boot Loader section.

Table 21-1. Lock Bit Byte(Note:)

Lock Bit Byte	Bit No	Description	Default Value
	7	–	1 (unprogrammed)
	6	–	1 (unprogrammed)
	5	–	1 (unprogrammed)
	4	–	1 (unprogrammed)
	3	–	1 (unprogrammed)
	2	–	1 (unprogrammed)
LB2	1	Lock bit	1 (unprogrammed)
LB1	0	Lock bit	1 (unprogrammed)

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

Table 21-2. Lock Bit Protection Modes[1][2]

Memory Lock Bits			Protection Type
LB Mode	LB2	LB1	
1	1	1	No memory lock features enabled.
2	1	0	Further programming of the Flash and EEPROM is disabled in Parallel and Serial Programming mode. The Fuse bits are locked in both Serial and Parallel Programming mode.[1]
3	0	0	Further programming and verification of the Flash and EEPROM is disabled in Parallel and Serial Programming mode. The Fuse bits are locked in both Serial and Parallel Programming mode.[1]

Notes: 1. Program the Fuse bits before programming the LB1 and LB2.

2. "1" means unprogrammed, "0" means programmed

21.2 Fuse Bits

The ATtiny87/167 has three Fuse bytes. Table 21-3, Table 21-4 & Table 21-5 describe briefly the functionality of all the fuses and how they are mapped into the Fuse bytes.

The SPM instruction is enabled for the whole Flash if the SELFPRGEN fuse is programmed ("0"), otherwise it is disabled.

Table 21-5. Fuse Low Byte

Fuse Low Byte	Bit No	Description	Default Value
CKDIV8[4]	7	Divide clock by 8	0 (programmed)
CKOUT[3]	6	Clock output	1 (unprogrammed)
SUT1	5	Select start-up time	1 (unprogrammed)[1]
SUT0	4	Select start-up time	0 (programmed)[1]
CKSEL3	3	Select Clock source	0 (programmed)[2]
CKSEL2	2	Select Clock source	0 (programmed)[2]
CKSEL1	1	Select Clock source	1 (unprogrammed)[2]
CKSEL0	0	Select Clock source	0 (programmed)[2]

Notes:
1. The default value of SUT[1:0] results in maximum start-up time for the default clock source. See Table 4-4 on page 27 for details.
2. The default setting of CKSEL[3:0] results in internal RC Oscillator @ 8 MHz. See Table 4-3 on page 27 for details.
3. The CKOUT Fuse allows the system clock to be output on PORTB5. See ”Clock Output Buffer” on page 31. for details.
4. See ”System Clock Prescaler” on page 37.for details.

21.2.1 Latching of Fuses

The fuse values are latched when the device enters programming mode and changes of the fuse values will have no effect until the part leaves Programming mode. This does not apply to the EESAVE Fuse which will take effect once it is programmed. The fuses are also latched on Power-up in Normal mode.

21.3 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. The three bytes reside in a separate address space.

Table 21-6. Signature Bytes

Device	Address	Value	Signature Byte Description
ATtiny87	0	0x1E	Indicates manufactured by Atmel
	1	0x93	Indicates 8 KB Flash memory
	2	0x87	Indicates ATtiny87 device when address 1 contains 0x93
ATtiny167	0	0x1E	Indicates manufactured by Atmel
	1	0x94	Indicates 16 KB Flash memory
	2	0x87	Indicates ATtiny167 device when address 1 contains 0x94

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:

Low: > **2** CPU clock cycles for f_{ck} < 12 MHz, **3** CPU clock cycles for f_{ck} >= 12 MHz

High: > **2** CPU clock cycles for f_{ck} < 12 MHz, **3** CPU clock cycles for f_{ck} >= 12 MHz

21.8.1 Serial Programming Algorithm

When writing serial data to the ATtiny87/167, data is clocked on the rising edge of SCK.

When reading data from the ATtiny87/167, data is clocked on the falling edge of SCK. See Figure 21-7 and Figure 21-8 for timing details.

To program and verify the ATtiny87/167 in the Serial Programming mode, the following sequence is recommended (see four byte instruction formats in Table 21-15 on page 239):

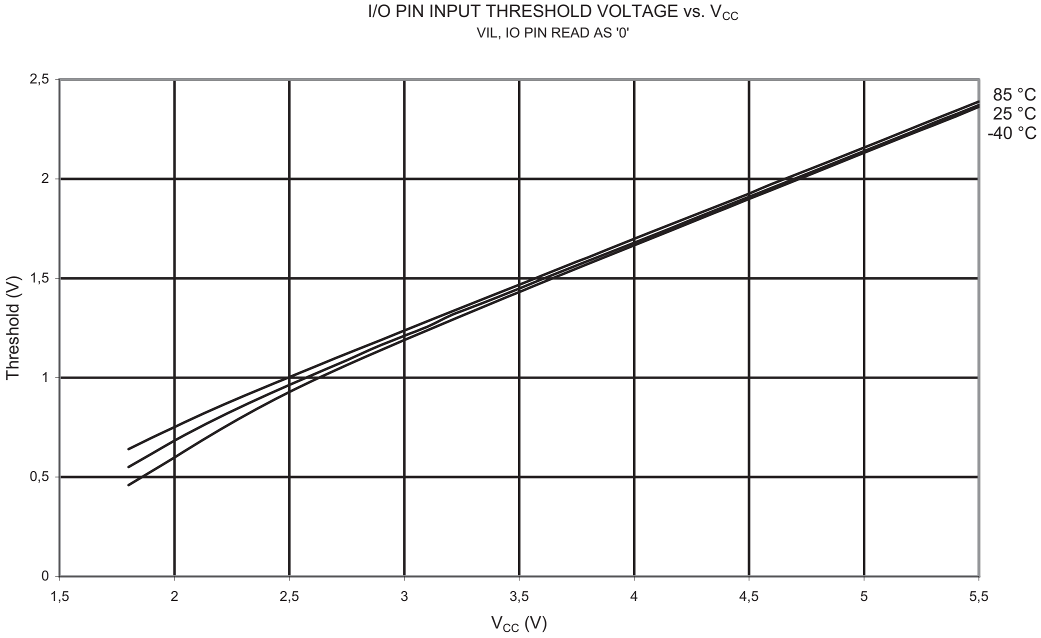
1. Power-up sequence:
   Apply power between V_{CC} and GND while $\overline{\text{RESET}}$ and SCK are set to “0”. In some systems, the programmer can not guarantee that SCK is held low during power-up. In this case, $\overline{\text{RESET}}$ must be given a positive pulse of at least two CPU clock cycles duration after SCK has been set to “0”.
2. Wait for at least 20 ms and enable serial programming by sending the Programming Enable serial instruction to pin MOSI.
3. The serial programming instructions will not work if the communication is out of synchronization. When in sync. the second byte (0x53), will echo back when issuing the third byte of the Programming Enable instruction. Whether the echo is correct or not, all four bytes of the instruction must be transmitted. If the 0x53 did not echo back, give $\overline{\text{RESET}}$ a positive pulse and issue a new Programming Enable command.
4. The Flash is programmed one page at a time. The memory page is loaded one byte at a time by supplying the 5 LSB of the address and data together with the Load Program memory Page instruction. To ensure correct loading of the page, the data low byte must be loaded before data high byte is applied for a given address. The Program memory Page is stored by loading the Write Program memory Page instruction with the 6 MSB of the address. If polling (RDY/$\overline{\text{BSY}}$) is not used, the user must wait at least t_{WD_FLASH} before issuing the next page. (See Table 21-14) Accessing the serial programming interface before the Flash write operation completes can result in incorrect programming.
5. **A:** The EEPROM array is programmed one byte at a time by supplying the address and data together with the appropriate Write instruction. An EEPROM memory location is first automatically erased before new data is written. If polling (RDY/$\overline{\text{BSY}}$) is not used, the user must wait at least t_{WD_EEPROM} before issuing the next byte. (See Table 21-14) In a chip erased device, no 0xFFs in the data file(s) need to be programmed.
   B: The EEPROM array is programmed one page at a time. The Memory page is loaded one byte at a time by supplying the 2 LSB of the address and data together with the Load EEPROM Memory Page instruction. The EEPROM Memory Page is stored by loading the Write EEPROM Memory Page Instruction with the 6 MSB of the address. When using EEPROM page access only byte locations loaded with the Load EEPROM Memory Page instruction is altered. The remaining locations remain unchanged. If polling (RDY/$\overline{\text{BSY}}$) is not used, the used must wait at least t_{WD_EEPROM} before issuing the next page (See Table 21-8). In a chip erased device, no 0xFF in the data file(s) need to be programmed.
6. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO.

T_A = -40°C to +85°C, V_{CC} = 1.8V to 5.5V (unless otherwise noted) (Continued)

Symbol	Parameter	Condition	Min.	Typ.[1]	Max.	Units
R_{RST}	Reset Pull-up Resistor		30		60	kΩ
R_{pu}	I/O Pin Pull-up Resistor		20		50	kΩ
I_{CC}	Power Supply Current [6] Active Mode (external clock)	8 MHz, V_{CC} = 5V		5.5	7.0	mA
		4 MHz, V_{CC} = 3V		1.8	2.5	mA
		1 MHz, V_{CC} = 2V		0.3	0.6	mA
	Power Supply Current [6] Idle Mode (external clock)	8 MHz, V_{CC} = 5V		1.8	2.5	mA
		4 MHz, V_{CC} = 3V		0.5	0.8	mA
		1 MHz, V_{CC} = 2V		0.05	0.2	mA
	Power Supply Current [7] Power-down Mode	WDT enabled, V_{CC} = 3V		5	10	µA
		WDT disabled, V_{CC} = 3V		0.15	4	µA
V_{ACIO}	Analog Comparator Input Offset Voltage	V_{CC} = 5V $V_{in} = V_{CC}/2$	-10	10	40	mV
I_{ACLK}	Analog Comparator Input Leakage Current	V_{CC} = 5V $V_{in} = V_{CC}/2$	-50		50	nA
t_{ACID}	Analog Comparator Propagation Delay Common Mode $V_{CC}/2$	V_{CC} = 2.7V		170		ns
		V_{CC} = 5.0V		180		ns

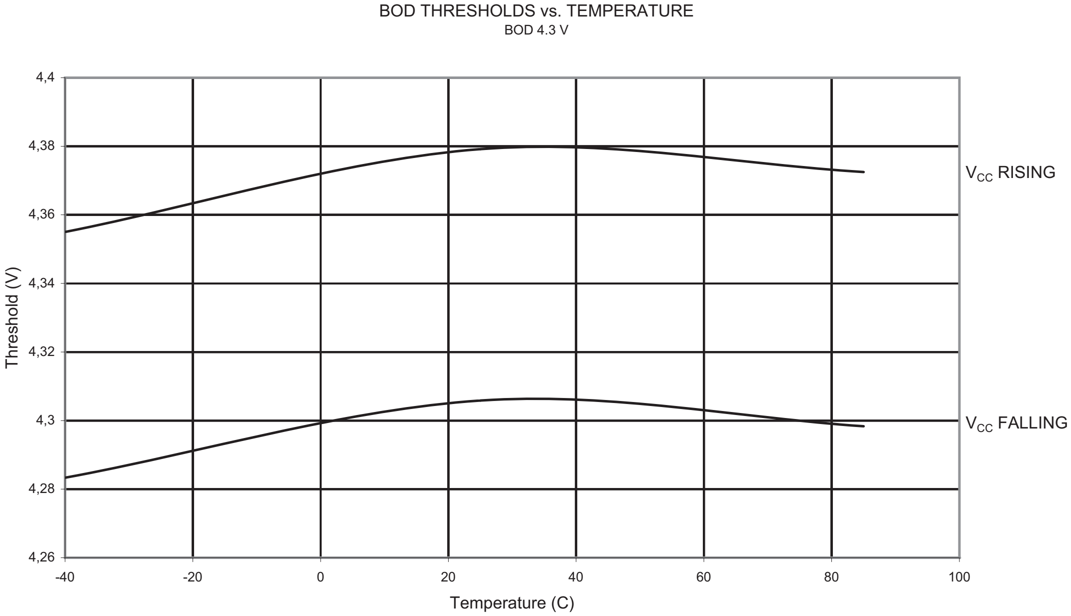
Notes:
1. “Typ.", typical values at 25°C. Maximum values are characterized values and not test limits in production.
2. "Max.” means the highest value where the pin is guaranteed to be read as low.
3. “Min.” means the lowest value where the pin is guaranteed to be read as high.
4. Although each I/O port can sink more than the test conditions (10 mA at V_{CC}= 5V, 5 mA at V_{CC} = 3V) under steady state conditions (non-transient), the following must be observed:
   - The sum of all IOL, for all ports, should not exceed 120 mA.
   If IOL exceeds the test condition, VOL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test condition.
5. Although each I/O port can source more than the test conditions (10 mA at V_{CC} = 5V, 5 mA at V_{CC} = 3V) under steady state conditions (non-transient), the following must be observed:
   - The sum of all IOH, for all ports, should not exceed 120 mA.
   If IOH exceeds the test condition, VOH may exceed the related specification. Pins are not guaranteed to source current greater than the listed test condition.
6. Values are with external clock using methods described in “Minimizing Power Consumption” on page 45. Power Reduction is enabled (PRR = 0xFF) and there is no I/O drive.
7. BOD Disabled.

Figure 23-21. V_{IL}: Input Threshold Voltage vs. V_{CC} (I/O Pin, Read as '0')



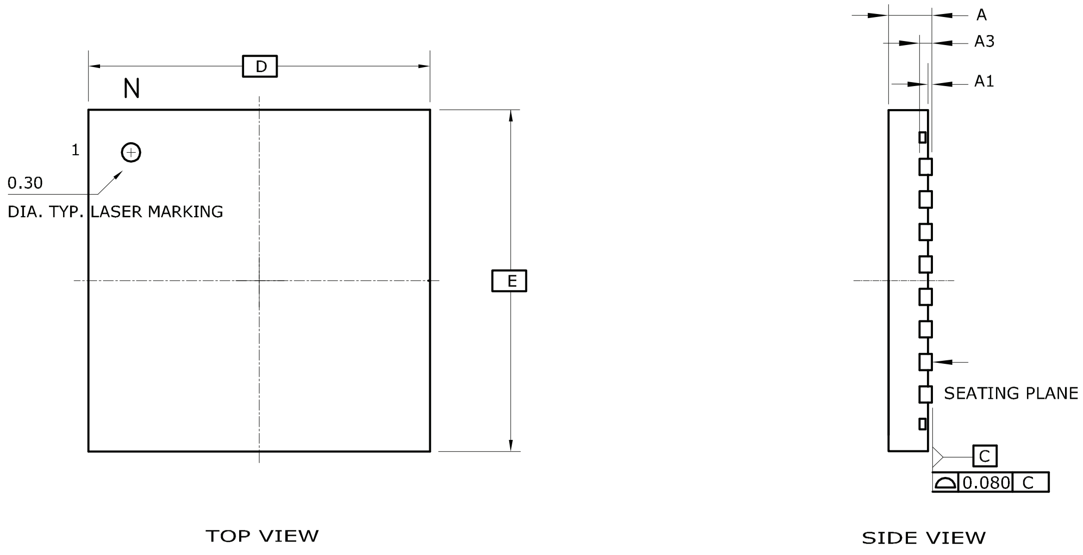
23.9 BOD, Bandgap and Reset

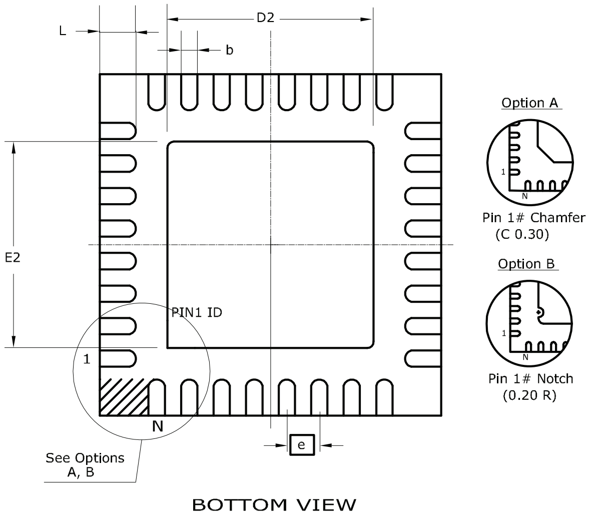
Figure 23-22. BOD Thresholds vs. Temperature (BOD Level is 4.3V)



27. Packaging Information

27.1 32PN





COMMON DIMENSIONS

(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	0.80	0.85	0.90	
A1	0.00	----	0.05	
A3	0.20 REF			
D/E	5.00 BSC			
D2/E2	3.00	3.10	3.20	
L	0.30	0.40	0.50	
b	0.18	0.25	0.30	2
e	0.50 BSC			
n	32			

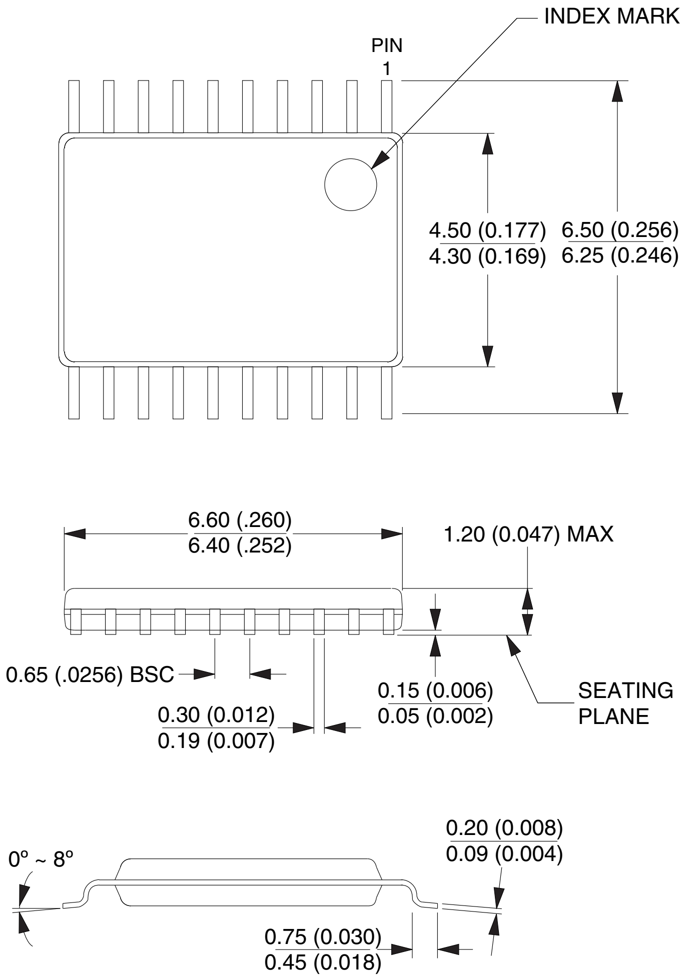
Notes : 1. This drawing is for general information only. Refer to JEDEC Drawing MO-220, Variation VHHD-2, for proper dimensions, tolerances, datums, etc.
2. Dimension b applies to metallized terminal and is measured between 0.15mm and 0.30mm from the terminal tip.
If the terminal has the optical radius on the other end of the terminal, the dimension should not be measured in that radius area.

01/31/2012

	TITLE	GPC	DRAWING NO.	REV.
Package Drawing Contact: **packagedrawings@atmel.com**	PN, 32 Leads , 0.50mm pitch, 5 x 5 mm Very Thin quad Flat No Lead Package (VQFN) Sawn	ZMF	PN	I

27.3 20X

Dimensions in Millimeters and (Inches).
Controlling dimension: Millimeters.
JEDEC Standard MO-153 AC



10/23/03

ATMEL 2325 Orchard Parkway San Jose, CA 95131	**TITLE** **20X**, (Formerly 20T), 20-lead, 4.4 mm Body Width, Plastic Thin Shrink Small Outline Package (TSSOP)	**DRAWING NO.** 20X	**REV.** C