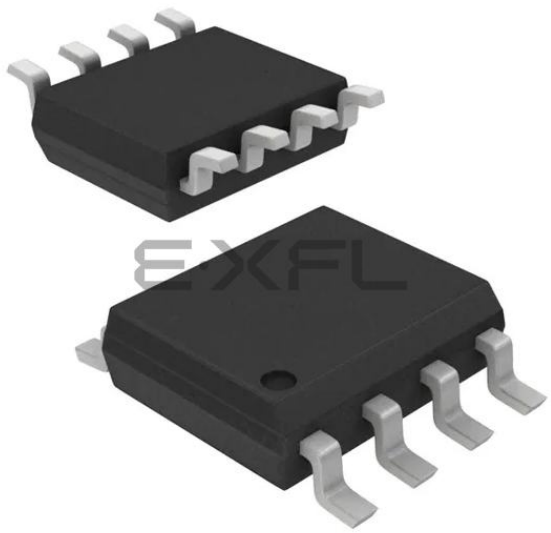


Welcome to E-XFL.COM

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	6MHz
Connectivity	-
Peripherals	WDT
Number of I/O	6
Program Memory Size	1KB (512 x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	-
Voltage - Supply (Vcc/Vdd)	4V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	8-SOIC (0.209", 5.30mm Width)
Supplier Device Package	8-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/attiny11-6su

ATtiny11 Block Diagram

See Figure 1 on page 3. The ATtiny11 provides the following features: 1K bytes of Flash, up to five general-purpose I/O lines, one input line, 32 general-purpose working registers, an 8-bit timer/counter, internal and external interrupts, programmable Watchdog Timer with internal oscillator, and two software-selectable power-saving modes. The Idle Mode stops the CPU while allowing the timer/counters and interrupt system to continue functioning. The Power-down Mode saves the register contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset. The wake-up or interrupt on pin change features enable the ATtiny11 to be highly responsive to external events, still featuring the lowest power consumption while in the power-down modes.

The device is manufactured using Atmel's high-density nonvolatile memory technology. By combining an RISC 8-bit CPU with Flash on a monolithic chip, the Atmel ATtiny11 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

Figure 1. The ATtiny11 Block Diagram

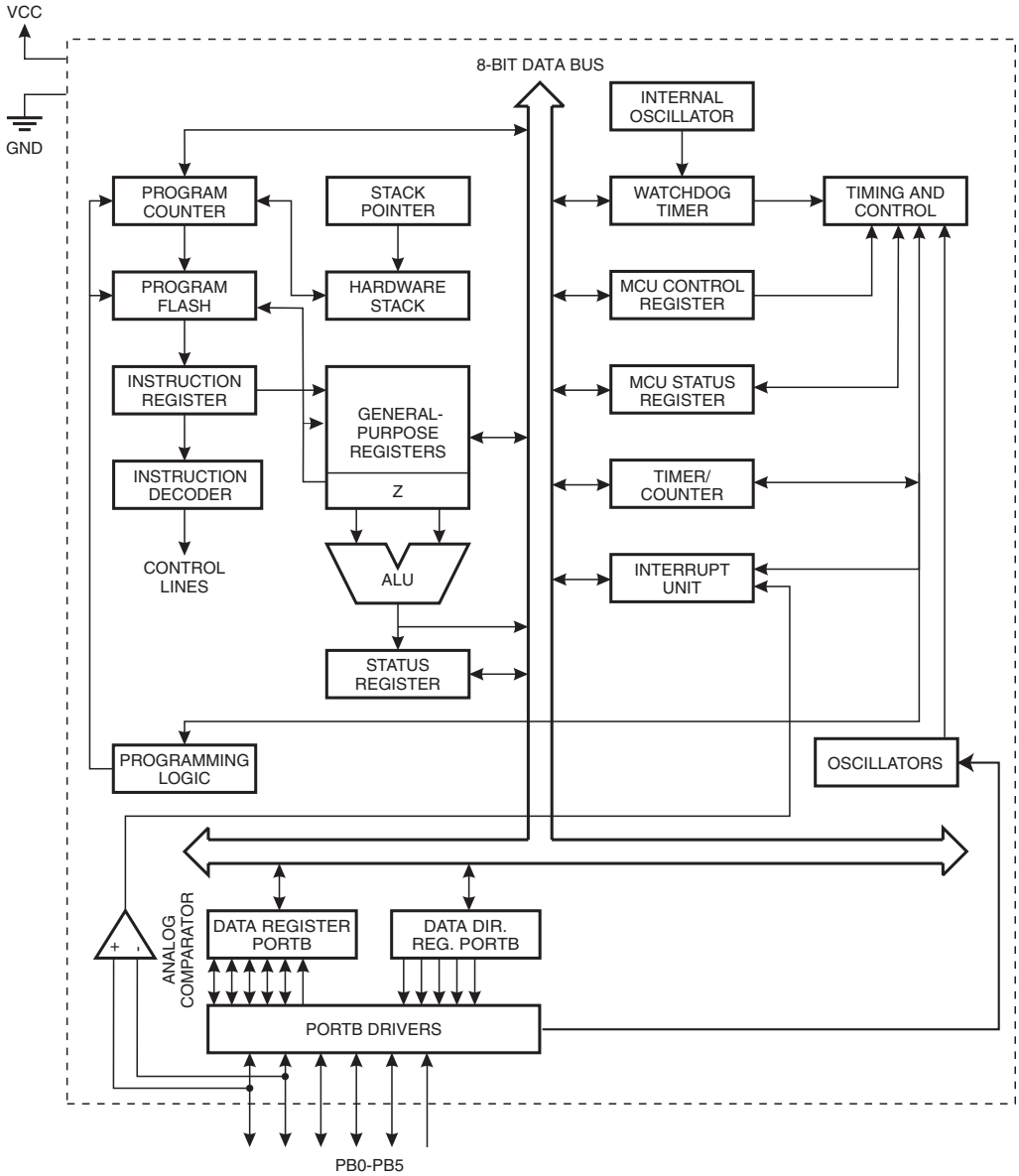
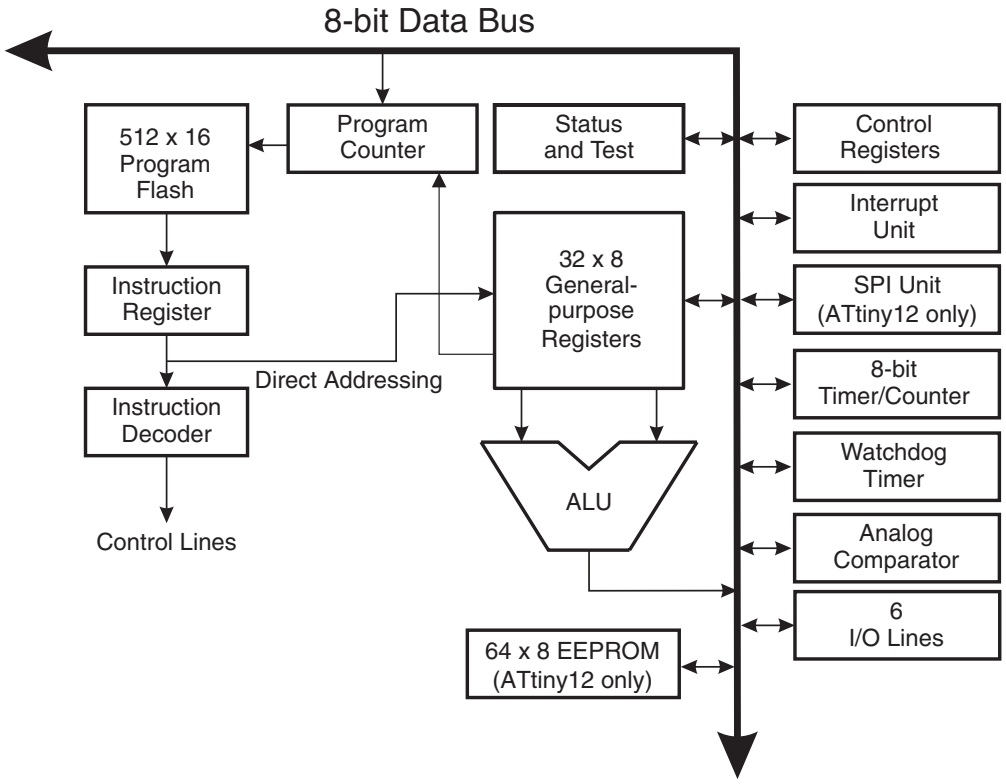


Figure 3. The ATtiny11/12 AVR RISC Architecture



A flexible interrupt module has its control registers in the I/O space with an additional global interrupt enable bit in the status register. All the different interrupts have a separate interrupt vector in the interrupt vector table at the beginning of the program memory. The different interrupts have priority in accordance with their interrupt vector position. The lower the interrupt vector address, the higher the priority.

ALU – Arithmetic Logic Unit

The high-performance AVR ALU operates in direct connection with all the 32 general-purpose working registers. Within a single clock cycle, ALU operations between registers in the register file are executed. The ALU operations are divided into three main categories – arithmetic, logic and bit-functions. Some microcontrollers in the AVR product family feature a hardware multiplier in the arithmetic part of the ALU.

Subroutine and Interrupt Hardware Stack

The ATtiny11/12 uses a 3-level-deep hardware stack for subroutines and interrupts. The hardware stack is 9 bits wide and stores the program counter (PC) return address while subroutines and interrupts are executed.

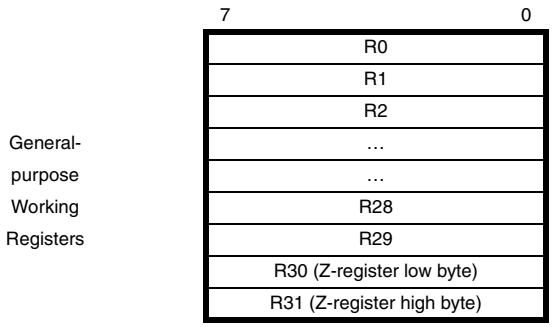
RCALL instructions and interrupts push the PC return address onto stack level 0, and the data in the other stack levels 1-2 are pushed one level deeper in the stack. When a RET or RETI instruction is executed the returning PC is fetched from stack level 0, and the data in the other stack levels 1-2 are popped one level in the stack.

If more than three subsequent subroutine calls or interrupts are executed, the first values written to the stack are overwritten. Pushing four return addresses A1, A2, A3, and A4, followed by four subroutine or interrupt returns, will pop A4, A3, A2, and once more A2 from the hardware stack.

General-purpose Register File

Figure 4 shows the structure of the 32 general-purpose registers in the CPU.

Figure 4. AVR CPU General-purpose Working Registers



All the register operating instructions in the instruction set have direct- and single-cycle access to all registers. The only exception is the five constant arithmetic and logic instructions SBCI, SUBI, CPI, ANDI, and ORI between a constant and a register and the LDI instruction for load-immediate constant data. These instructions apply to the second half of the registers in the register file – R16..R31. The general SBC, SUB, CP, AND, OR and all other operations between two registers or on a single register apply to the entire register file.

Registers 30 and 31 form a 16-bit pointer (the Z-pointer) which is used for indirect Flash memory and register file access. When the register file is accessed, the contents of R31 are discarded by the CPU.

Memories

I/O Memory

The I/O space definition of the ATtiny11/12 is shown in the following table:

Table 5. ATtiny11/12 I/O Space

Address Hex	Name	Device	Function
$3F	SREG	ATtiny11/12	Status Register
$3B	GIMSK	ATtiny11/12	General Interrupt Mask Register
$3A	GIFR	ATtiny11/12	General Interrupt Flag Register
$39	TIMSK	ATtiny11/12	Timer/Counter Interrupt Mask Register
$38	TIFR	ATtiny11/12	Timer/Counter Interrupt Flag Register
$35	MCUCR	ATtiny11/12	MCU Control Register
$34	MCUSR	ATtiny11/12	MCU Status Register
$33	TCCR0	ATtiny11/12	Timer/Counter0 Control Register
$32	TCNT0	ATtiny11/12	Timer/Counter0 (8-bit)
$31	OSCCAL	ATtiny12	Oscillator Calibration Register
$21	WDTCR	ATtiny11/12	Watchdog Timer Control Register
$1E	EEAR	ATtiny12	EEPROM Address Register
$1D	EEDR	ATtiny12	EEPROM Data Register
$1C	EECR	ATtiny12	EEPROM Control Register
$18	PORTB	ATtiny11/12	Data Register, Port B
$17	DDRB	ATtiny11/12	Data Direction Register, Port B
$16	PINB	ATtiny11/12	Input Pins, Port B
$08	ACSR	ATtiny11/12	Analog Comparator Control and Status Register

Note: Reserved and unused locations are not shown in the table.

All the different ATtiny11/12 I/O and peripherals are placed in the I/O space. The different I/O locations are accessed by the IN and OUT instructions transferring data between the 32 general-purpose working registers and the I/O space. I/O registers within the address range $00 - $1F 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 Summary for more details.

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

The different I/O and peripherals control registers are explained in the following sections.

Program and Data Addressing Modes

The ATtiny11/12 AVR RISC Microcontroller supports powerful and efficient addressing modes. This section describes the different addressing modes supported in the ATtiny11/12. In the figures, OP means the operation code part of the instruction word. To simplify, not all figures show the exact location of the addressing bits.

System Control and Reset

Reset Sources

The ATtiny11/12 provides three or four sources of reset:

- Power-on Reset. The MCU is reset when the supply voltage is below the power-on reset threshold (V_{POT}).
- External Reset. The MCU is reset when a low level is present on the $\overline{\text{RESET}}$ pin for more than 50 ns.
- Watchdog Reset. The MCU is reset when the Watchdog timer period expires and the Watchdog is enabled.
- Brown-out Reset. The MCU is reset when the supply voltage V_{CC} falls below a certain voltage (ATtiny12 only).

During reset, all I/O registers are then set to their initial values, and the program starts execution from address $000. The instruction placed in address $000 must be an RJMP – relative jump – instruction to the reset handling routine. If the program never enables an interrupt source, the interrupt vectors are not used, and regular program code can be placed at these locations. The circuit diagram in Figure 15 shows the reset logic for the ATtiny11. Figure 16 shows the reset logic for the ATtiny12. Table 7 defines the electrical parameters of the reset circuitry for ATtiny11. Table 9 shows the parameters of the reset circuitry for ATtiny12.

Figure 15. Reset Logic for the ATtiny11

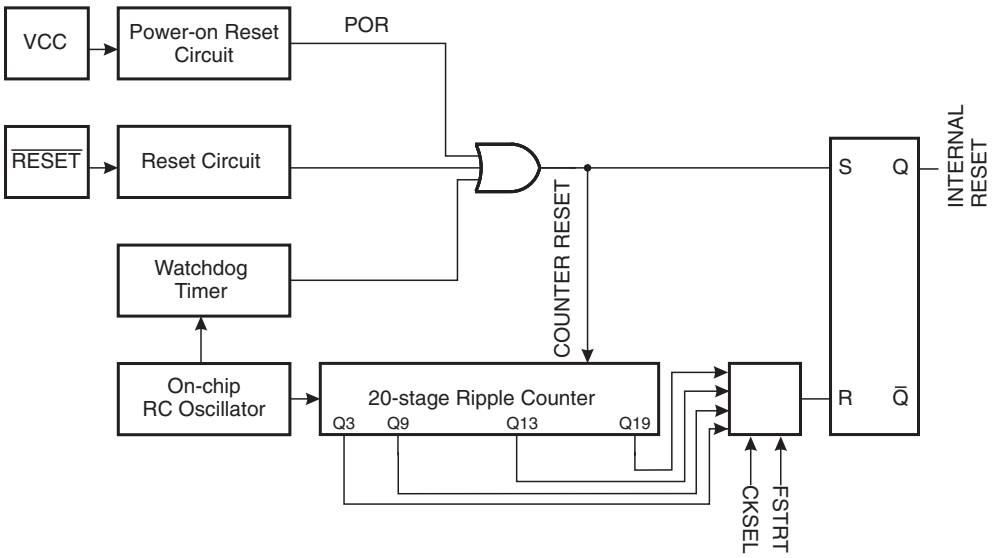


Table 7. Reset Characteristics for the ATtiny11

Symbol	Parameter	Min	Typ	Max	Units
V_{POT}[1]	Power-on Reset Threshold Voltage (rising)	1.0	1.4	1.8	V
	Power-on Reset Threshold Voltage (falling)	0.4	0.6	0.8	V
V_{RST}	$\overline{\text{RESET}}$ Pin Threshold Voltage		0.6 V_{CC}		V

Note: 1. The Power-on Reset will not work unless the supply voltage has been below V_{POT} (falling).

To identify a reset condition, the user software should clear both the PORF and EXTRF bits as early as possible in the program. Checking the PORF and EXTRF values is done before the bits are cleared. If the bit is cleared before an external or watchdog reset occurs, the source of reset can be found by using the following truth table:

Table 13. Reset Source Identification

EXTRF	PORF	Reset Source
0	0	Watchdog Reset
1	0	External Reset
0	1	Power-on Reset
1	1	Power-on Reset

MCU Status Register – MCUSR for the ATtiny12

The MCU Status Register provides information on which reset source caused an MCU reset.

Bit	7	6	5	4	3	2	1	0	
$34	-	-	-	-	WDRF	BORF	EXTRF	PORF	MCUSR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	See Bit Description				

- **Bit 7..4 - Res: Reserved Bits**

These bits are reserved bits in the ATtiny12 and always read as zero.

- **Bit 3 - WDRF: Watchdog Reset Flag**

This bit is set if a watchdog reset occurs. The bit is reset by a power-on reset, or by writing a logic zero to the flag.

- **Bit 2 - BORF: Brown-out Reset Flag**

This bit is set if a brown-out reset occurs. The bit is reset by a power-on reset, or by writing a logic zero to the flag.

- **Bit 1 - EXTRF: EXTernal Reset Flag**

This bit is set if an external reset occurs. The bit is reset by a power-on reset, or by writing a logic zero to the flag.

- **Bit 0 - PORF: Power-on Reset Flag**

This bit is set if a power-on reset occurs. The bit is reset by writing a logic zero to the flag.

To use the reset flags to identify a reset condition, the user should read and then reset the MCUSR as early as possible in the program. If the register is cleared before another reset occurs, the source of the reset can be found by examining the reset flags.

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

This bit is set (one) when a comparator output event triggers the interrupt mode defined by ACI1 and ACI0. The Analog Comparator Interrupt routine is executed if the ACIE bit is set (one) and the I-bit in SREG is set (one). ACI is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ACI is cleared by writing a logic one to the flag.

- **Bit 3 - ACIE: Analog Comparator Interrupt Enable**

When the ACIE bit is set (one) and the I-bit in the Status Register is set (one), the Analog Comparator Interrupt is activated. When cleared (zero), the interrupt is disabled.

- **Bit 2 - Res: Reserved Bit**

This bit is a reserved bit in the ATtiny11/12 and will always read as zero.

- **Bits 1,0 - ACIS1, ACIS0: Analog Comparator Interrupt Mode Select**

These bits determine which comparator events that trigger the Analog Comparator Interrupt. The different settings are shown in Table 20.

Table 20. ACIS1/ACIS0 Settings

ACIS1	ACIS0	Interrupt Mode
0	0	Comparator Interrupt on Output Toggle
0	1	Reserved
1	0	Comparator Interrupt on Falling Output Edge
1	1	Comparator Interrupt on Rising Output Edge

Note: When changing the ACIS1/ACIS0 bits, the Analog Comparator Interrupt must be disabled by clearing its interrupt enable bit in the ACSR register. Otherwise, an interrupt can occur when the bits are changed.

Caution: Using the SBI or CBI instruction on bits other than ACI in this register will write a one back into ACI if it is read as set, thus clearing the flag.

High-voltage Serial Programming Algorithm

To program and verify the ATtiny11/12 in the High-voltage Serial Programming mode, the following sequence is recommended (See instruction formats in Table 23):

1. Power-up sequence: Apply 4.5 - 5.5V between V_{CC} and GND. Set PB5 and PB0 to “0” and wait at least 100 ns. Toggle PB3 at least four times with minimum 100 ns pulse-width. Set PB3 to “0”. Wait at least 100 ns. Apply 12V to PB5 and wait at least 100 ns before changing PB0. Wait 8 µs before giving any instructions.
2. The Flash array is programmed one byte at a time by supplying first the address, then the low and high data byte. The write instruction is self-timed, wait until the PB2 (RDY/$\overline{\text{BSY}}$) pin goes high.
3. The EEPROM array (ATtiny12 only) is programmed one byte at a time by supplying first the address, then the data byte. The write instruction is self-timed, wait until the PB2 (RDY/$\overline{\text{BSY}}$) pin goes high.
4. Any memory location can be verified by using the Read instruction which returns the contents at the selected address at serial output PB2.
5. Power-off sequence:Set PB3 to “0”.
   Set PB5 to “1”.
   Turn V_{CC} power off.

When writing or reading serial data to the ATtiny11/12, data is clocked on the rising edge of the serial clock, see Figure 27, Figure 28 and Table 24 for details.

Figure 27. High-voltage Serial Programming Waveforms

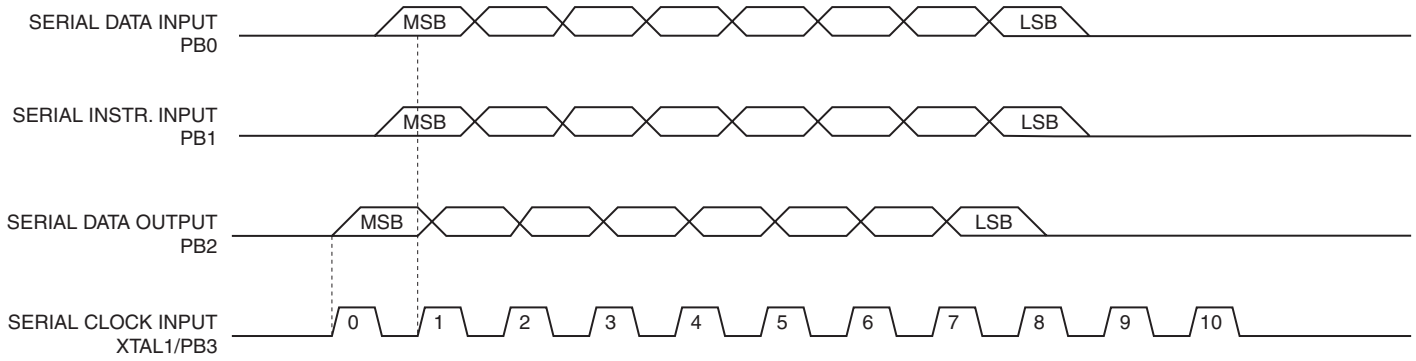


Table 23. High-voltage Serial Programming Instruction Set for ATtiny11/12

Instruction		Instruction Format				Operation Remarks
		Instr.1	Instr.2	Instr.3	Instr.4	
Chip Erase	PB0 PB1 PB2	0_1000_0000_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_0100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0100_1100_00 x_xxxx_xxxx_xx	Wait after Instr.4 until PB2 goes high for the Chip Erase cycle to finish.
Write Flash High and Low Address	PB0 PB1 PB2	0_0001_0000_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_0000_000**a**_00 0_0001_1100_00 x_xxxx_xxxx_xx	0_**bbbb_bbbb**_00 0_0000_1100_00 x_xxxx_xxxx_xx		Repeat Instr.2 for a new 256 byte page. Repeat Instr.3 for each new address.
Write Flash Low byte	PB0 PB1 PB2	0_**iiii_iiii**_00 0_0010_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_0100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1100_00 0_0000_0000_00		Wait after Instr.3 until PB2 goes high. Repeat Instr.1, Instr. 2 and Instr.3 for each new address.
Write Flash High byte	PB0 PB1 PB2	0_**iiii_iiii**_00 0_0011_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0111_0100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0111_1100_00 0_0000_0000_00		Wait after Instr.3 until PB2 goes high. Repeat Instr.1, Instr. 2 and Instr.3 for each new address.
Read Flash High and Low Address	PB0 PB1 PB2	0_0000_0010_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_0000_000**a**_00 0_0001_1100_00 x_xxxx_xxxx_xx	0_**bbbb_bbbb**_00 0_0000_1100_00 x_xxxx_xxxx_xx		Repeat Instr.2 and Instr.3 for each new address.
Read Flash Low byte	PB0 PB1 PB2	0_0000_0000_00 0_0110_1000_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1100_00 **o_oooo_ooo**x_xx			Repeat Instr.1 and Instr.2 for each new address.
Read Flash High byte	PB0 PB1 PB2	0_0000_0000_00 0_0111_1000_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0111_1100_00 **o_oooo_ooo**x_xx			Repeat Instr.1 and Instr.2 for each new address.
Write EEPROM Low Address (ATtiny12)	PB0 PB1 PB2	0_0001_0001_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_00**bb_bbbb**_00 0_0000_1100_00 x_xxxx_xxxx_xx			Repeat Instr.2 for each new address.
Write EEPROM byte (ATtiny12)	PB0 PB1 PB2	0_**iiii_iiii**_00 0_0010_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_0100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1100_00 0_0000_0000_00		Wait after Instr.3 until PB2 goes high
Read EEPROM Low Address (ATtiny12)	PB0 PB1 PB2	0_0000_0011_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_00**bb_bbbb**_00 0_0000_1100_00 x_xxxx_xxxx_xx			Repeat Instr.2 for each new address.

Table 23. High-voltage Serial Programming Instruction Set for ATtiny11/12 (Continued)

Instruction		Instruction Format				Operation Remarks
		Instr.1	Instr.2	Instr.3	Instr.4	
Read EEPROM byte (ATtiny12)	PB0 PB1 PB2	0_0000_0000_00 0_0110_1000_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1100_00 **o_oooo_ooo**x_xx			Repeat Instr.2 for each new address
Write Fuse bits (ATtiny11)	PB0 PB1 PB2	0_0100_0000_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_000**7_6543**_00 0_0010_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_0100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1100_00 x_xxxx_xxxx_xx	Wait t_{WLWH_PFB} after Instr.3 for the Write fuse bits cycle to finish. Write **7** - **3** = “0” to program the fuse bit.
Write Fuse bits (ATtiny12)	PB0 PB1 PB2	0_0100_0000_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_**CBA9_8543**_00 0_0010_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_0100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1100_00 x_xxxx_xxxx_xx	Wait after Instr.4 until PB2 goes high. Write **C** - **A**, **9, 8,** 5 - **3** = “0” to program the fuse bit.
Write Lock bits	PB0 PB1 PB2	0_0010_0000_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_0000_0210_00 0_0010_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_0100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1100_00 0_0000_0000_00	Wait after Instr.4 until PB2 goes high. Write ***2***, ***1*** = “0” to program the lock bit.
Read Fuse bits (ATtiny11)	PB0 PB1 PB2	0_0000_0100_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1000_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1100_00 x_xx**76_543**x_xx		Reading **7** - **3** = “0” means the fuse bit is programmed.
Read Fuse bits (ATtiny12)	PB0 PB1 PB2	0_0000_0100_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1000_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1100_00 **C_BA98_543**x_xx		Reading **C** - **A**, **9**, **8**, **5** - **3** = “0” means the fuse bit is programmed.
Read Lock bits	PB0 PB1 PB2	0_0000_0100_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0111_1000_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0111_1100_00 x_xxxx_***21***xx_xx		Reading ***2***, ***1*** = “0” means the lock bit is programmed.
Read Signature Bytes	PB0 PB1 PB2	0_0000_1000_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_0000_00**bb**_00 0_0000_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1000_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1100_00 **o_oooo_ooo**x_xx	Repeat Instr.2 - Instr.4 for each signature byte address
Read Calibration Byte (ATtiny12)	PB0 PB1 PB2	0_0000_1000_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0000_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0111_1000_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0111_1100_00 **o_oooo_ooo**x_xx	

Note: **a** = address high bits
b = address low bits
i = data in
o = data out
x = don’t care
1 = Lock Bit1
2 = Lock Bit2
3 = CKSEL0 Fuse
4 = CKSEL1 Fuse
5 = CKSEL2 Fuse
9, **6** = RSTDISBL Fuse
7 = FSTRT Fuse
8 = CKSEL3 Fuse
A = SPIEN Fuse
B = BODEN Fuse
C = BODLEVEL Fuse

If the chip Erase command in Low-voltage Serial Programming is executed only once, one data byte may be written to the flash after erase. Using the following algorithm guarantees that the flash will be erased:

- Execute a chip erase command
- Write $FF to address $00 in the flash
- Execute a second chip erase command

For the EEPROM, an auto-erase cycle is provided within the self-timed write instruction and there is no need to first execute the Chip Erase instruction. The Chip Erase instruction turns the content of every memory location in both the program and EEPROM arrays into $FF.

The program and EEPROM memory arrays have separate address spaces:

$0000 to $01FF for program memory and $000 to $03F for EEPROM memory.

The device can be clocked by any clock option during Low-voltage Serial Programming. The minimum low and high periods for the serial clock (SCK) input are defined as follows:

Low: > 2 MCU clock cycles

High: > 2 MCU clock cycles

Low-voltage Serial Programming Algorithm

When writing serial data to the ATtiny12, data is clocked on the rising edge of SCK. When reading data from the ATtiny12, data is clocked on the falling edge of SCK. See Figure 30, Figure 31 and Table 26 for timing details. To program and verify the ATtiny12 in the serial programming mode, the following sequence is recommended (See 4 byte instruction formats in Table 25):

1. Power-up sequence:

   Apply power between VCC and GND while RESET and SCK are set to “0”. In accordance with the setting of CKSEL fuses, apply a crystal/resonator, external clock or RC network, or let the device run on the internal RC oscillator. 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 MCU 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 the MOSI (PB0) pin.
3. The serial programming instructions will not work if the communication is out of synchronization. When in sync, the second byte ($53) will echo back when issuing the third byte of the Programming Enable instruction. Whether the echo is correct or not, all 4 bytes of the instruction must be transmitted. If the $53 did not echo back, give SCK a positive pulse and issue a new Programming Enable instruction. If the $53 is not seen within 32 attempts, there is no functional device connected.
4. If a Chip Erase is performed (must be done to erase the Flash), wait t_{WD_ERASE} after the instruction, give $\overline{\text{RESET}}$ a positive pulse, and start over from Step 2. See Table 27 on page 58 for t_{WD_ERASE} value.
5. The Flash or 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. Use Data Polling to detect when the next byte in the Flash or EEPROM can be written. If polling is not used, wait t_{WD_FLASH} or t_{WD_EEPROM} before transmitting the

Table 25. Low-voltage Serial Programming Instruction Set

Instruction	Instruction Format				Operation
	Byte 1	Byte 2	Byte 3	Byte4	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx	Enable serial programming while $\overline{\text{RESET}}$ is low.
Chip Erase	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Chip erase Flash and EEPROM memory arrays.
Read Program Memory	0010 **H**000	xxxx xxx**a**	**bbbb bbbb**	**oooo oooo**	Read **H** (high or low) data **o** from program memory at word address **a**:**b**.
Write Program Memory	0100 **H**000	xxxx xxx**a**	**bbbb bbbb**	**iiii iiii**	Write **H** (high or low) data **i** to program memory at word address **a**:**b**.
Read EEPROM Memory	1010 0000	xxxx xxxx	xx**bb bbbb**	**oooo oooo**	Read data **o** from EEPROM memory at address **b**.
Write EEPROM Memory	1100 0000	xxxx xxxx	xx**bb bbbb**	**iiii iiii**	Write data **i** to EEPROM memory at address **b**.
Write Lock Bits	1010 1100	1111 1***21***1	xxxx xxxx	xxxx xxxx	Write lock bits. Set bits ***1,2*** = “0” to program lock bits.
Read Lock Bits	0101 1000	xxxx xxxx	xxxx xxxx	xxxx x***21***x	Read lock bits. “0” = programmed, “1” = unprogrammed.
Read Signature Bytes	0011 0000	xxxx xxxx	0000 00**bb**	**oooo oooo**	Read signature byte **o** at address **b**.(1)
Read Calibration Byte	0011 1000	xxxx xxxx	0000 0000	**oooo oooo**	
Write Fuse Bits	1010 1100	101x xxxx	xxxx xxxx	**A987 6543**	Set bits **A**, **9** - **3** = “0” to program, “1” to unprogram.
Read Fuse Bits	0101 0000	xxxx xxxx	xxxx xxxx	**A987 6543**	Read fuse bits. “0” = programmed, “1” = unprogrammed.

Note: **a** = address high bits
b = address low bits
H = 0 - Low byte, 1 - High byte
o = data out
i = data in
x = don’t care
1 = Lock bit 1
2 = Lock bit 2
3 = CKSEL0 Fuse
4 = CKSEL1 Fuse
5 = CKSEL2 Fuse
6 = CKSEL3 Fuse
7 = RSTDISBL Fuse
8 = SPIEN Fuse
9 = BODEN Fuse
A = BODLEVEL Fuse

Note: 1. The signature bytes are not readable in Lock mode 3, i.e. both lock bits programmed.

Low-voltage Serial Programming Characteristics

Figure 31. Low-voltage Serial Programming Timing

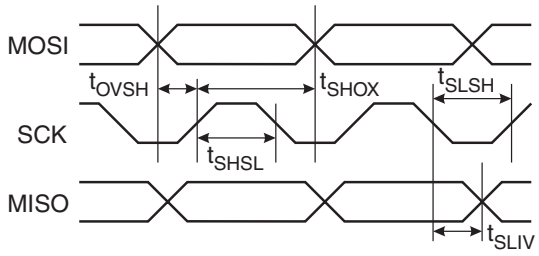


Table 26. Low-voltage Serial Programming Characteristics T_A = -40°C to 85°C, V_{CC} = 2.2 - 5.5V (Unless otherwise noted)

Symbol	Parameter	Min	Typ	Max	Units
$1/t_{CLCL}$	Oscillator Frequency (V_{CC} = 2.2 - 2.7V)	0		1	MHz
t_{CLCL}	Oscillator Period (V_{CC} = 2.2 - 2.7V)	1000			ns
$1/t_{CLCL}$	Oscillator Frequency (V_{CC} = 2.7 - 4.0V)	0		4	MHz
t_{CLCL}	Oscillator Period (V_{CC} = 2.7 - 4.0V)	250			ns
$1/t_{CLCL}$	Oscillator Frequency (V_{CC} = 4.0 - 5.5V)	0		8	MHz
t_{CLCL}	Oscillator Period (V_{CC} = 4.0 - 5.5V)	125			ns
t_{SHSL}	SCK Pulse Width High	2 t_{CLCL}			ns
t_{SLSH}	SCK Pulse Width Low	2 t_{CLCL}			ns
t_{OVSH}	MOSI Setup to SCK High	t_{CLCL}			ns
t_{SHOX}	MOSI Hold after SCK High	2 t_{CLCL}			ns
t_{SLIV}	SCK Low to MISO Valid	10	16	32	ns

Table 27. Minimum Wait Delay after the Chip Erase Instruction

Symbol	Minimum Wait Delay
t_{WD_ERASE}	6.8 ms

Table 28. Minimum Wait Delay after Writing a Flash or EEPROM Location

Symbol	Minimum Wait Delay
t_{WD_FLASH}	3.4 ms
t_{WD_EEPROM}	6.8 ms

Electrical Characteristics

Absolute Maximum Ratings

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground	-1.0V to V_{CC}+0.5V
Voltage on $\overline{\text{RESET}}$ with respect to Ground	-1.0V to +13.0V
Maximum Operating Voltage	6.0V
DC Current per I/O Pin	40.0 mA
DC Current V_{CC} and GND Pins	100.0 mA

*NOTICE: Stresses beyond those ratings listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics – Preliminary Data

T_A = -40°C to 85°C, V_{CC} = 2.7V to 5.5V for ATtiny11, V_{CC} = 1.8V to 5.5V for ATtiny12 (Unless otherwise noted)

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{IL}	Input Low Voltage	Except (XTAL)	-0.5		0.3 V_{CC}[1]	V
V_{IL1}	Input Low Voltage	XTAL	-0.5		0.1 V_{CC}[1]	V
V_{IH}	Input High Voltage	Except (XTAL, $\overline{\text{RESET}}$)	0.6 V_{CC}[2]		V_{CC} + 0.5	V
V_{IH1}	Input High Voltage	XTAL	0.7 V_{CC}[2]		V_{CC} + 0.5	V
V_{IH2}	Input High Voltage	$\overline{\text{RESET}}$	0.85 V_{CC}[2]		V_{CC} + 0.5	V
V_{OL}	Output Low Voltage[3] Port B	I_{OL} = 20 mA, V_{CC} = 5V I_{OL} = 10 mA, V_{CC} = 3V			0.6 0.5	V V
V_{OL}	Output Low Voltage PB5 (ATtiny12)	I_{OL} = 12 mA, V_{CC} = 5V I_{OL} = 6 mA, V_{CC} = 3V			0.6 0.5	V V
V_{OH}	Output High Voltage[4] Port B	I_{OH} = -3 mA, V_{CC} = 5V I_{OH} = -1.5 mA, V_{CC} = 3V	4.3 2.3			V V
I_{IL}	Input Leakage Current I/O Pin	V_{CC} = 5.5V, Pin Low (Absolute value)			8.0	µA
I_{IH}	Input Leakage Current I/O Pin	V_{CC} = 5.5V, Pin High (Absolute value)			8.0	µA
$R_{I/O}$	I/O Pin Pull-Up		35		122	kΩ

Figure 36. Active Supply Current vs. V_{CC}, Device Clocked by External 32kHz Crystal

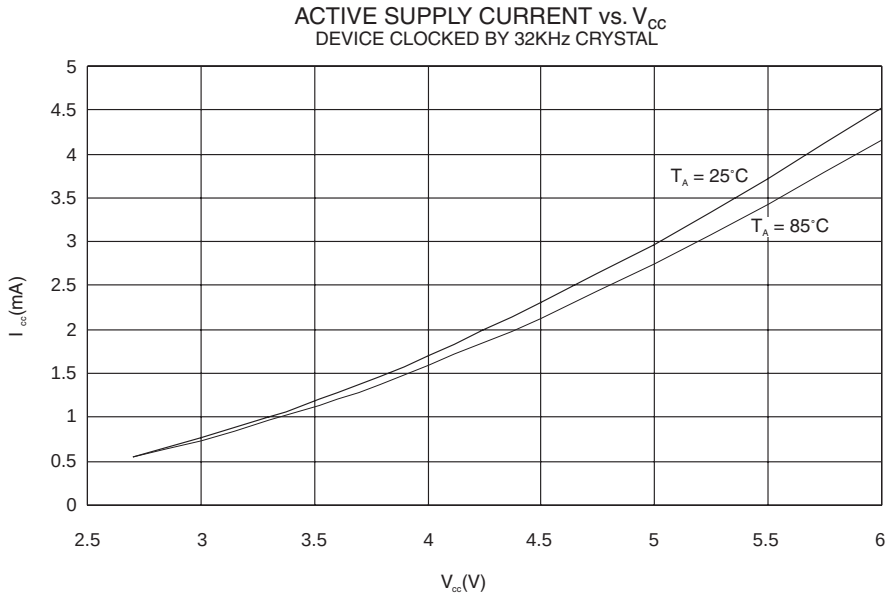


Figure 37. Idle Supply Current vs. Frequency

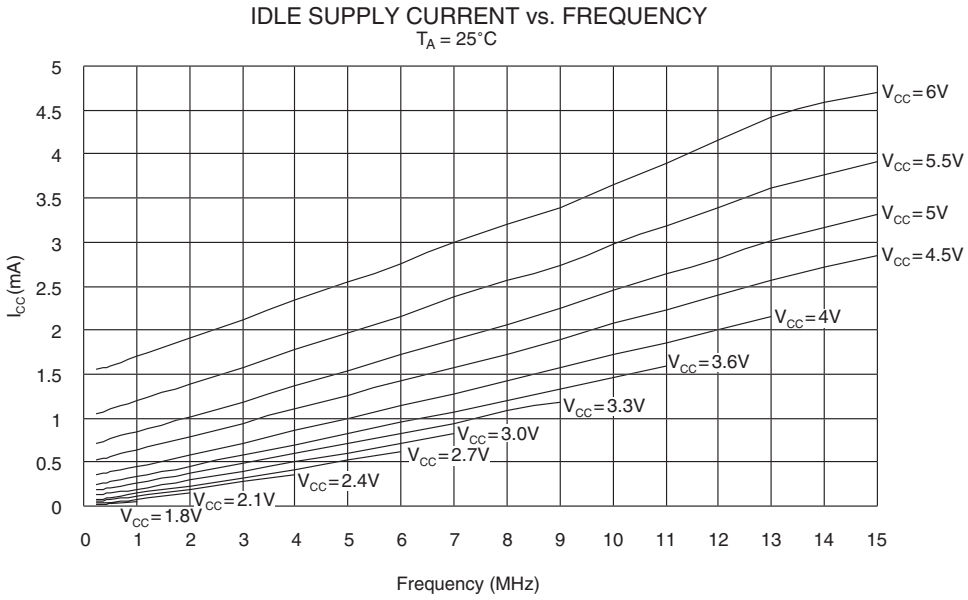


Figure 71. I/O Pin Input Threshold Voltage vs. V_{CC} (T_A = 25°C)

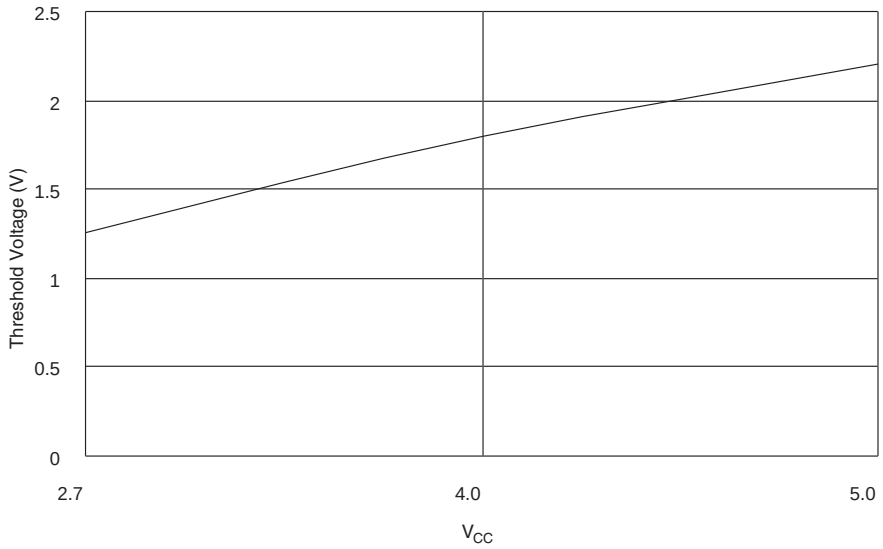
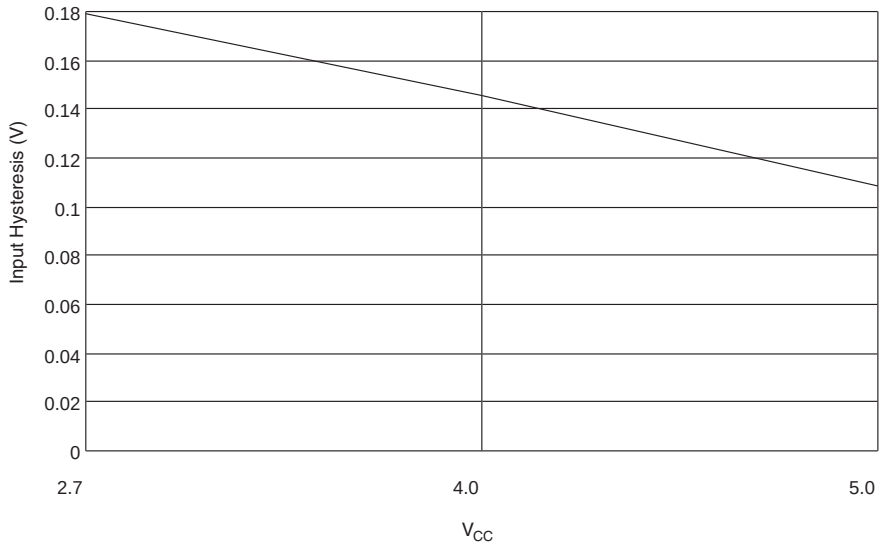


Figure 72. I/O Pin Input Hysteresis vs. V_{CC} (T_A = 25°C)



Register Summary ATtiny11

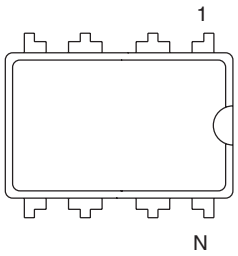
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
$3F	SREG	I	T	H	S	V	N	Z	C	page 9
$3E	Reserved									
$3D	Reserved									
$3C	Reserved									
$3B	GIMSK	-	INT0	PCIE	-	-	-	-	-	page 33
$3A	GIFR	-	INTF0	PCIF	-	-	-	-	-	page 34
$39	TIMSK	-	-	-	-	-	-	TOIE0	-	page 34
$38	TIFR	-	-	-	-	-	-	TOV0	-	page 35
$37	Reserved									
$36	Reserved									
$35	MCUCR	-	-	SE	SM	-	-	ISC01	ISC00	page 32
$34	MCUSR	-	-	-	-	-	-	EXTRF	PORF	page 28
$33	TCCR0	-	-	-	-	-	CS02	CS01	CS00	page 41
$32	TCNT0	Timer/Counter0 (8 Bit)								page 41
$31	Reserved									
$30	Reserved									
...	Reserved									
$22	Reserved									
$21	WDTCR	-	-	-	WDTOE	WDE	WDP2	WDP1	WDP0	page 43
$20	Reserved									
$1F	Reserved									
$1E	Reserved									
$1D	Reserved									
$1C	Reserved									
$1B	Reserved									
$1A	Reserved									
$19	Reserved									
$18	PORTB	-	-	-	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	page 37
$17	DDRB	-	-	-	DDB4	DDB3	DDB2	DDB1	DDB0	page 37
$16	PINB	-	-	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	page 37
$15	Reserved									
...	Reserved									
$0A	Reserved									
$09	Reserved									
$08	ACSR	ACD	-	ACO	ACI	ACIE	-	ACIS1	ACIS0	page 45
...	Reserved									
$00	Reserved									

Notes: 1. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.

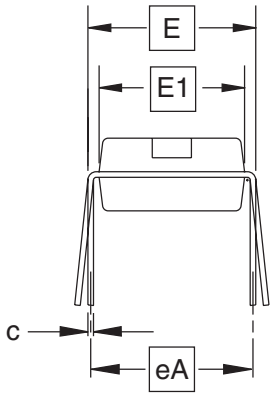
2. Some of the status flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers $00 to $1F only.

Packaging Information

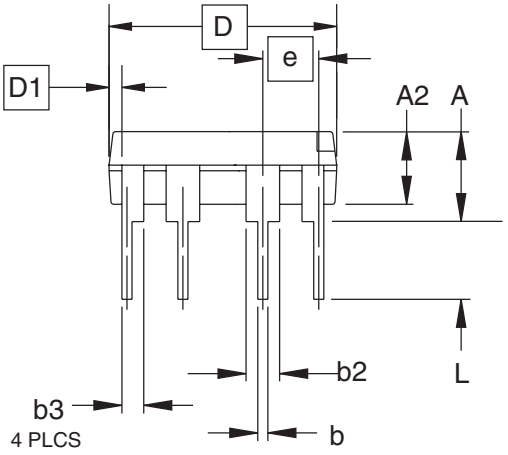
8P3



Top View



End View



Side View

COMMON DIMENSIONS
(Unit of Measure = inches)

SYMBOL	MIN	NOM	MAX	NOTE
A			0.210	2
A2	0.115	0.130	0.195	
b	0.014	0.018	0.022	5
b2	0.045	0.060	0.070	6
b3	0.030	0.039	0.045	6
c	0.008	0.010	0.014	
D	0.355	0.365	0.400	3
D1	0.005			3
E	0.300	0.310	0.325	4
E1	0.240	0.250	0.280	3
e	0.100 BSC			
eA	0.300 BSC			4
L	0.115	0.130	0.150	2

Notes:
1. This drawing is for general information only; refer to JEDEC Drawing MS-001, Variation BA for additional information.
2. Dimensions A and L are measured with the package seated in JEDEC seating plane Gauge GS-3.
3. D, D1 and E1 dimensions do not include mold Flash or protrusions. Mold Flash or protrusions shall not exceed 0.010 inch.
4. E and eA measured with the leads constrained to be perpendicular to datum.
5. Pointed or rounded lead tips are preferred to ease insertion.
6. b2 and b3 maximum dimensions do not include Dambar protrusions. Dambar protrusions shall not exceed 0.010 (0.25 mm).

01/09/02

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**8P3**, 8-lead, 0.300" Wide Body, Plastic Dual In-line Package (PDIP)	8P3	B

Datasheet Revision History

Please note that the page numbers listed in this section are refering to this document. The revision numbers are referring to the document revision.

Rev. 1006F-06/07

1. **“Not recommended for new design”.**

Rev. 1006E-07/06

1. **Updated chapter layout.**
2. **Updated Power-down in “Sleep Modes for the ATtiny11” on page 20.**
3. **Updated Power-down in “Sleep Modes for the ATtiny12” on page 20.**
4. **Updated Table 16 on page 36.**
5. **Updated “Calibration Byte in ATtiny12” on page 49.**
6. **Updated “Ordering Information” on page 87.**
7. **Updated “Packaging Information” on page 89.**

Rev. 1006D-07/03

1. **Updated V_{BOT} values in Table 9 on page 24.**

Rev. 1006C-09/01

1. **N/A**

Table of Contents