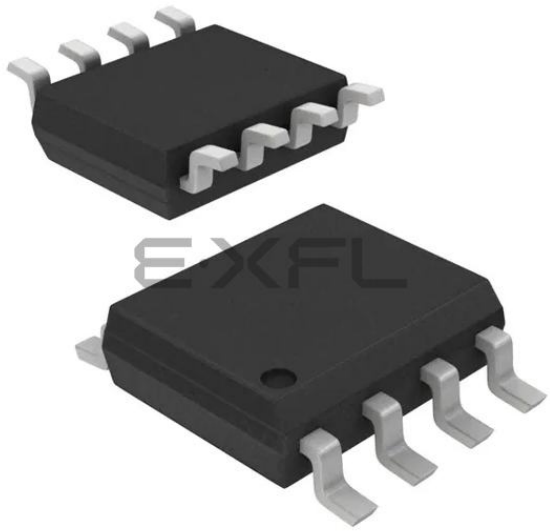


Welcome to E-XFL.COM

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	USI
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	6
Program Memory Size	8KB (4K x 16)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 4x10b
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/attiny85-20sfr

5. AVR Memories

This section describes the different memories in the ATtiny25/45/85. The AVR architecture has two main memory spaces, the Data memory and the Program memory space. In addition, the ATtiny25/45/85 features an EEPROM Memory for data storage. All three memory spaces are linear and regular.

5.1 In-System Re-programmable Flash Program Memory

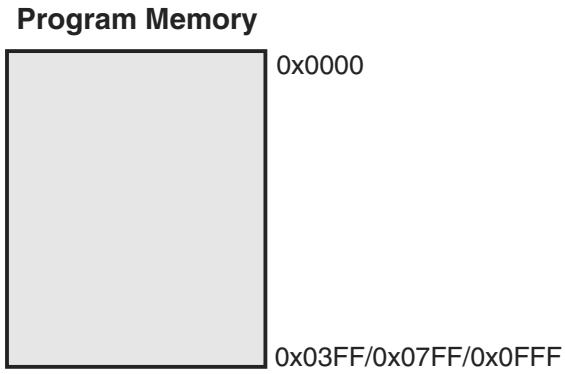
The ATtiny25/45/85 contains 2/4/8K bytes On-chip In-System Reprogrammable Flash memory for program storage. Since all AVR instructions are 16 or 32 bits wide, the Flash is organized as 1024/2048/4096 x 16.

The Flash memory has an endurance of at least 10,000 write/erase cycles. The ATtiny25/45/85 Program Counter (PC) is 10/11/12 bits wide, thus addressing the 1024/2048/4096 Program memory locations. “Memory Programming” on page 147 contains a detailed description on Flash data serial downloading using the SPI pins.

Constant tables can be allocated within the entire Program memory address space (see the LPM – Load Program memory instruction description).

Timing diagrams for instruction fetch and execution are presented in “Instruction Execution Timing” on page 11.

Figure 5-1. Program Memory Map



5.2 SRAM Data Memory

Figure 5-2 shows how the ATtiny25/45/85 SRAM Memory is organized.

The lower 224/352/607 Data memory locations address both the Register File, the I/O memory and the internal data SRAM. The first 32 locations address the Register File, the next 64 locations the standard I/O memory, and the last 128/256/512 locations address the internal data SRAM.

The five different addressing modes for the Data memory cover: Direct, Indirect with Displacement, Indirect, Indirect with Pre-decrement, and Indirect with Post-increment. In the Register File, registers R26 to R31 feature the indirect addressing pointer registers.

The direct addressing reaches the entire data space.

The Indirect with Displacement mode reaches 63 address locations from the base address given by the Y- or Z-register.

When using register indirect addressing modes with automatic pre-decrement and post-increment, the address registers X, Y, and Z are decremented or incremented.

The 32 general purpose working registers, 64 I/O Registers, and the 128/256/512 bytes of internal data SRAM in the ATtiny25/45/85 are all accessible through all these addressing modes. The Register File is described in “General Purpose Register File” on page 10.

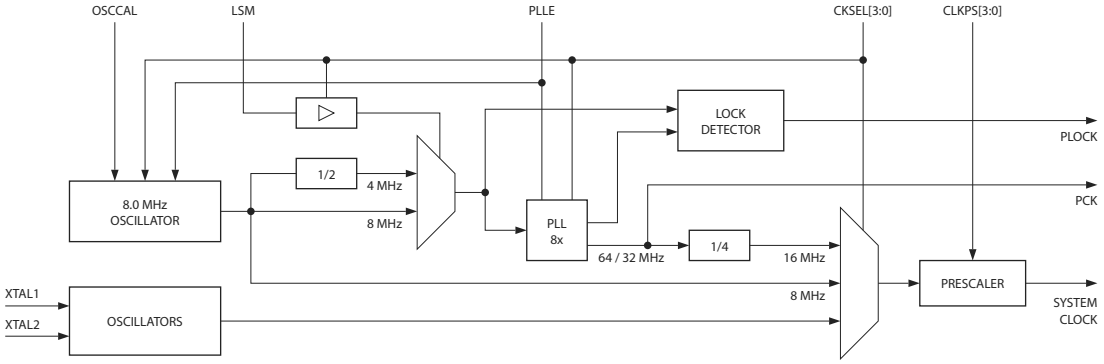
6.1.4 ADC Clock – clk_{ADC}

The ADC is provided with a dedicated clock domain. This allows halting the CPU and I/O clocks in order to reduce noise generated by digital circuitry. This gives more accurate ADC conversion results.

6.1.5 Internal PLL for Fast Peripheral Clock Generation - clk_{PCK}

The internal PLL in ATtiny25/45/85 generates a clock frequency that is 8x multiplied from a source input. By default, the PLL uses the output of the internal, 8.0 MHz RC oscillator as source. Alternatively, if bit LSM of PLLCSR is set the PLL will use the output of the RC oscillator divided by two. Thus the output of the PLL, the fast peripheral clock is 64 MHz. The fast peripheral clock, or a clock prescaled from that, can be selected as the clock source for Timer/Counter1 or as a system clock. See Figure 6-2. The frequency of the fast peripheral clock is divided by two when LSM of PLLCSR is set, resulting in a clock frequency of 32 MHz. Note, that LSM can not be set if PLL_{CLK} is used as system clock.

Figure 6-2. PCK Clocking System.



The PLL is locked on the RC oscillator and adjusting the RC oscillator via OSCCAL register will adjust the fast peripheral clock at the same time. However, even if the RC oscillator is taken to a higher frequency than 8 MHz, the fast peripheral clock frequency saturates at 85 MHz (worst case) and remains oscillating at the maximum frequency. It should be noted that the PLL in this case is not locked any longer with the RC oscillator clock. Therefore, it is recommended not to take the OSCCAL adjustments to a higher frequency than 8 MHz in order to keep the PLL in the correct operating range.

The internal PLL is enabled when:

- The PLLE bit in the register PLLCSR is set.
- The CKSEL fuse is programmed to ‘0001’.
- The CKSEL fuse is programmed to ‘0011’.

The PLLCSR bit PLOCK is set when PLL is locked.

Both internal RC oscillator and PLL are switched off in power down and stand-by sleep modes.

6.1.6 Internal PLL in ATtiny15 Compatibility Mode

Since ATtiny25/45/85 is a migration device for ATtiny15 users there is an ATtiny15 compatibility mode for backward compatibility. The ATtiny15 compatibility mode is selected by programming the CKSEL fuses to ‘0011’.

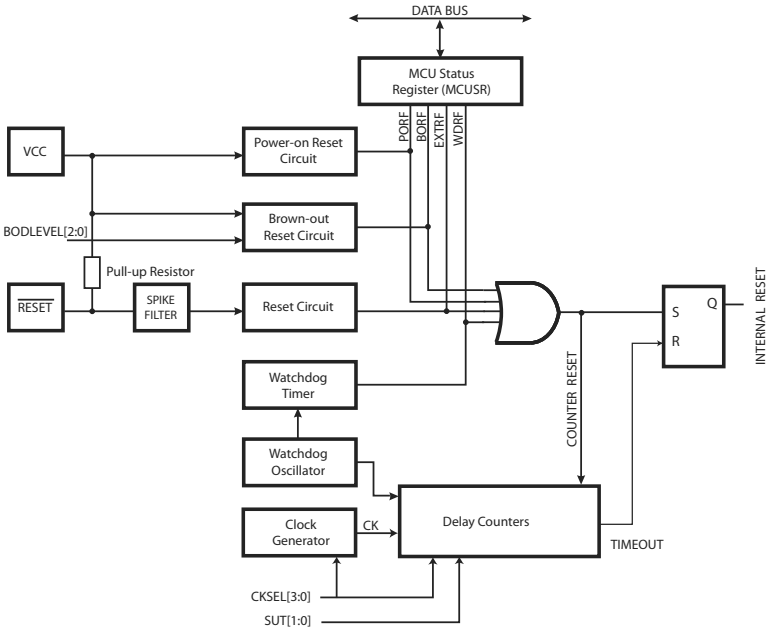
In the ATtiny15 compatibility mode the frequency of the internal RC oscillator is calibrated down to 6.4 MHz and the multiplication factor of the PLL is set to 4x. See Figure 6-3. With these adjustments the clocking system is ATtiny15-compatible and the resulting fast peripheral clock has a frequency of 25.6 MHz (same as in ATtiny15).

8. System Control and Reset

8.1 Resetting the AVR

During reset, all I/O Registers are set to their initial values, and the program starts execution from the Reset Vector. The instruction placed at the Reset Vector must be a 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 8-1 shows the reset logic. Electrical parameters of the reset circuitry are given in “System and Reset Characteristics” on page 165.

Figure 8-1. Reset Logic



The I/O ports of the AVR are immediately reset to their initial state when a reset source goes active. This does not require any clock source to be running.

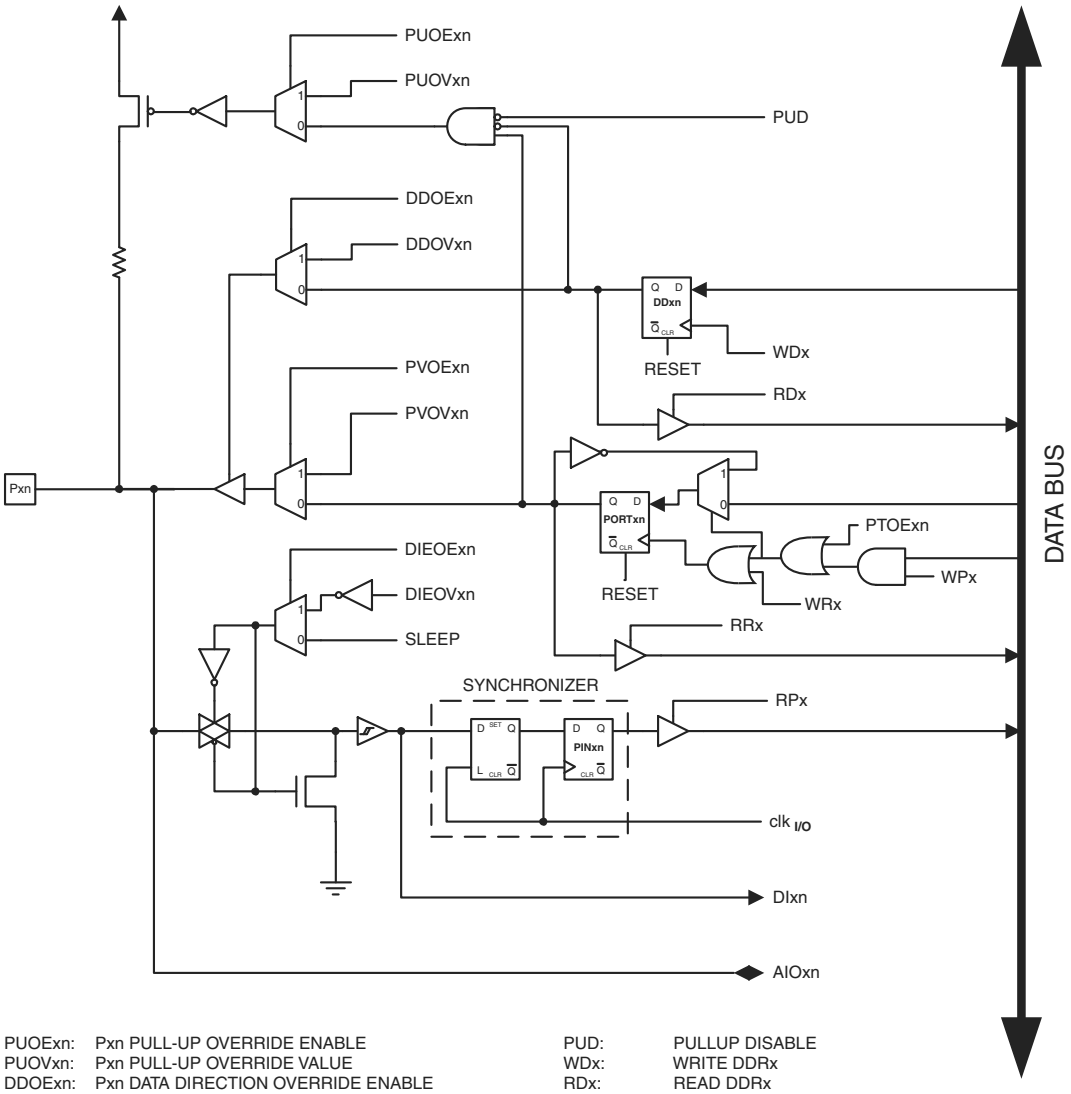
After all reset sources have gone inactive, a delay counter is invoked, stretching the internal reset. This allows the power to reach a stable level before normal operation starts. The time-out period of the delay counter is defined by the user through the SUT and CKSEL Fuses. The different selections for the delay period are presented in “Clock Sources” on page 25.

8.2 Reset Sources

The ATtiny25/45/85 has 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 longer than the minimum pulse length.
- 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} is below the Brown-out Reset threshold (V_{BOT}) and the Brown-out Detector is enabled.

Figure 10-5. Alternate Port Functions(1)



PUOExn:	Pxn PULL-UP OVERRIDE ENABLE
PUOVxn:	Pxn PULL-UP OVERRIDE VALUE
DDOExn:	Pxn DATA DIRECTION OVERRIDE ENABLE
DDOVxn:	Pxn DATA DIRECTION OVERRIDE VALUE
PVOExn:	Pxn PORT VALUE OVERRIDE ENABLE
PVOVxn:	Pxn PORT VALUE OVERRIDE VALUE
DIEOExn:	Pxn DIGITAL INPUT-ENABLE OVERRIDE ENABLE
DIEOVxn:	Pxn DIGITAL INPUT-ENABLE OVERRIDE VALUE
SLEEP:	SLEEP CONTROL
PTOExn:	Pxn, PORT TOGGLE OVERRIDE ENABLE

PUD:	PULLUP DISABLE
WDx:	WRITE DDRx
RDx:	READ DDRx
RRx:	READ PORTx REGISTER
WRx:	WRITE PORTx
RPx:	READ PORTx PIN
WPx:	WRITE PINx
$clk_{I/O}$:	I/O CLOCK
DIxn:	DIGITAL INPUT PIN n ON PORTx
AIOxn:	ANALOG INPUT/OUTPUT PIN n ON PORTx

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

Table 13-2. PWM Outputs OCR1A = $00 or OCR1C

COM1A1	COM1A0	OCR1A	Output OC1A
1	0	OCR1C	H
1	1	$00	H
1	1	OCR1C	L

In PWM mode, the Timer Overflow Flag - TOV1 is set when the TCNT1 counts to the OCR1C value and the TCNT1 is reset to $00. The Timer Overflow Interrupt1 is executed when TOV1 is set provided that Timer Overflow Interrupt and global interrupts are enabled. This also applies to the Timer Output Compare flags and interrupts.

The PWM frequency can be derived from the timer/counter clock frequency using the following equation:

$$f = \frac{f_{TCK1}}{(OCR1C + 1)}$$

The duty cycle of the PWM waveform can be calculated using the following equation:

$$D = \frac{(OCR1A + 1) \times T_{TCK1} - T_{PCK}}{(OCR1C + 1) \times T_{TCK1}}$$

...where T_{PCK} is the period of the fast peripheral clock (1/25.6 MHz = 39.1 ns).

Resolution indicates how many bits are required to express the value in the OCR1C register. It can be calculated using the following equation:

$$R = \log 2(OCR1C + 1)$$

Table 13-3. Timer/Counter1 Clock Prescale Select in the Asynchronous Mode

PWM Frequency	Clock Selection	CS1[3:0]	OCR1C	RESOLUTION
20 kHz	PCK/16	0101	199	7.6
30 kHz	PCK/16	0101	132	7.1
40 kHz	PCK/8	0100	199	7.6
50 kHz	PCK/8	0100	159	7.3
60 kHz	PCK/8	0100	132	7.1
70 kHz	PCK/4	0011	228	7.8
80 kHz	PCK/4	0011	199	7.6
90 kHz	PCK/4	0011	177	7.5
100 kHz	PCK/4	0011	159	7.3
110 kHz	PCK/4	0011	144	7.2
120 kHz	PCK/4	0011	132	7.1
130 kHz	PCK/2	0010	245	7.9
140 kHz	PCK/2	0010	228	7.8
150 kHz	PCK/2	0010	212	7.7

- **Bits 3:0 – CS1[3:0]: Clock Select Bits 3, 2, 1, and 0**

The Clock Select bits 3, 2, 1, and 0 define the prescaling source of Timer/Counter1.

Table 13-5. Timer/Counter1 Prescale Select

CS13	CS12	CS11	CS10	T/C1 Clock
0	0	0	0	T/C1 stopped
0	0	0	1	PCK
0	0	1	0	PCK/2
0	0	1	1	PCK/4
0	1	0	0	PCK/8
0	1	0	1	CK
0	1	1	0	CK/2
0	1	1	1	CK/4
1	0	0	0	CK/8
1	0	0	1	CK/16
1	0	1	0	CK/32
1	0	1	1	CK/64
1	1	0	0	CK/128
1	1	0	1	CK/256
1	1	1	0	CK/512
1	1	1	1	CK/1024

The Stop condition provides a Timer Enable/Disable function.

13.3.2 GTCCR – General Timer/Counter1 Control Register

Bit	7	6	5	4	3	2	1	0	
0x2C	TSM	PWM1B	COM1B1	COM1B0	FOC1B	FOC1A	PSR1	PSR0	GTCCR
Read/Write	R/W	R/W	R/W	R/W	W	W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bit 2 – FOC1A: Force Output Compare Match 1A**

Writing a logical one to this bit forces a change in the compare match output pin PB1 (OC1A) according to the values already set in COM1A1 and COM1A0. If COM1A1 and COM1A0 written in the same cycle as FOC1A, the new settings will be used. The Force Output Compare bit can be used to change the output pin value regardless of the timer value. The automatic action programmed in COM1A1 and COM1A0 takes place as if a compare match had occurred, but no interrupt is generated. The FOC1A bit always reads as zero. FOC1A is not in use if PWM1A bit is set.

- **Bit 1 – PSR1 : Prescaler Reset Timer/Counter1**

When this bit is set (one), the Timer/Counter prescaler (TCNT1 is unaffected) will be reset. The bit will be cleared by hardware after the operation is performed. Writing a zero to this bit will have no effect. This bit will always read as zero.

13.3.3 TCNT1 – Timer/Counter1

Bit	7	6	5	4	3	2	1	0	
0x2F	MSB							LSB	TCNT1
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	

This 8-bit register contains the value of Timer/Counter1.

Timer/Counter1 is realized as an up counter with read and write access. Due to synchronization of the CPU, Timer/Counter1 data written into Timer/Counter1 is delayed by one CPU clock cycle in synchronous mode and at most two CPU clock cycles for asynchronous mode.

13.3.4 OCR1A – Timer/Counter1 Output Compare RegisterA

Bit	7	6	5	4	3	2	1	0	
0x2E	MSB							LSB	OCR1A
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

The output compare register A is an 8-bit read/write register.

The Timer/Counter Output Compare Register A contains data to be continuously compared with Timer/Counter1. Actions on compare matches are specified in TCCR1. A compare match does only occur if Timer/Counter1 counts to the OCR1A value. A software write that sets TCNT1 and OCR1A to the same value does not generate a compare match.

A compare match will set the compare interrupt flag OCF1A after a synchronization delay following the compare event.

13.3.5 OCR1C – Timer/Counter1 Output Compare Register C

Bit	7	6	5	4	3	2	1	0	
0x2D	MSB							LSB	OCR1C
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	1	1	1	1	1	1	1	1	

The Output Compare Register B - OCR1B from ATtiny15 is replaced with the output compare register C - OCR1C that is an 8-bit read/write register. This register has the same function as the Output Compare Register B in ATtiny15.

The Timer/Counter Output Compare Register C contains data to be continuously compared with Timer/Counter1. A compare match does only occur if Timer/Counter1 counts to the OCR1C value. A software write that sets TCNT1 and OCR1C to the same value does not generate a compare match. If the CTC1 bit in TCCR1 is set, a compare match will clear TCNT1.

13.3.6 TIMSK – Timer/Counter Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
0x39	–	OCIE1A	OCIE1B	OCIE0A	OCIE0B	TOIE1	TOIE0	–	TIMSK
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R	
Initial value	0	0	0	0	0	0	0	0	

- **Bit 7 – Res: Reserved Bit**

This bit is a reserved bit in the ATtiny25/45/85 and always reads as zero.

The code is size optimized using only eight instructions (plus return). The code example assumes that the DO and USCK pins have been enabled as outputs in DDRB. The value stored in register r16 prior to the function is called is transferred to the slave device, and when the transfer is completed the data received from the slave is stored back into the register r16.

The second and third instructions clear the USI Counter Overflow Flag and the USI counter value. The fourth and fifth instructions set three-wire mode, positive edge clock, count at USITC strobe, and toggle USCK. The loop is repeated 16 times.

The following code demonstrates how to use the USI as an SPI master with maximum speed ($f_{SCK} = f_{CK}/2$):

```
SPITransfer_Fast:
    out    USIDR,r16
    ldi    r16,(1<<USIWM0)|(0<<USICS0)|(1<<USITC)
    ldi    r17,(1<<USIWM0)|(0<<USICS0)|(1<<USITC)|(1<<USICLK)

    out    USICR,r16 ; MSB
    out    USICR,r17
    out    USICR,r16
    out    USICR,r17
    out    USICR,r16
    out    USICR,r17
    out    USICR,r16
    out    USICR,r17
    out    USICR,r16
    out    USICR,r17
    out    USICR,r16
    out    USICR,r17
    out    USICR,r16
    out    USICR,r17
    out    USICR,r16 ; LSB
    out    USICR,r17

    in     r16,USIDR
ret
```

15.3.3 SPI Slave Operation Example

The following code demonstrates how to use the USI as an SPI slave:

```
init:
    ldi    r16,(1<<USIWM0)|(1<<USICS1)
    out    USICR,r16
...
SlaveSPITransfer:
    out    USIDR,r16
    ldi    r16,(1<<USIOIF)
    out  USISR,r16
SlaveSPITransfer_loop:
    in     r16, USISR
```

```
sbrs   r16, USIOIF
rjmp   SlaveSPITransfer_loop
in     r16,USIDR
ret
```

The code is size optimized using only eight instructions (plus return). The code example assumes that the DO and USCK pins have been enabled as outputs in DDRB. The value stored in register r16 prior to the function is called is transferred to the master device, and when the transfer is completed the data received from the master is stored back into the register r16.

Note that the first two instructions is for initialization, only, and need only be executed once. These instructions set three-wire mode and positive edge clock. The loop is repeated until the USI Counter Overflow Flag is set.

15.3.4 Two-wire Mode

The USI two-wire mode is compliant to the Inter IC (TWI) bus protocol, but without slew rate limiting on outputs and without input noise filtering. Pin names used in this mode are SCL and SDA.

Figure 15-4. Two-wire Mode Operation, Simplified Diagram

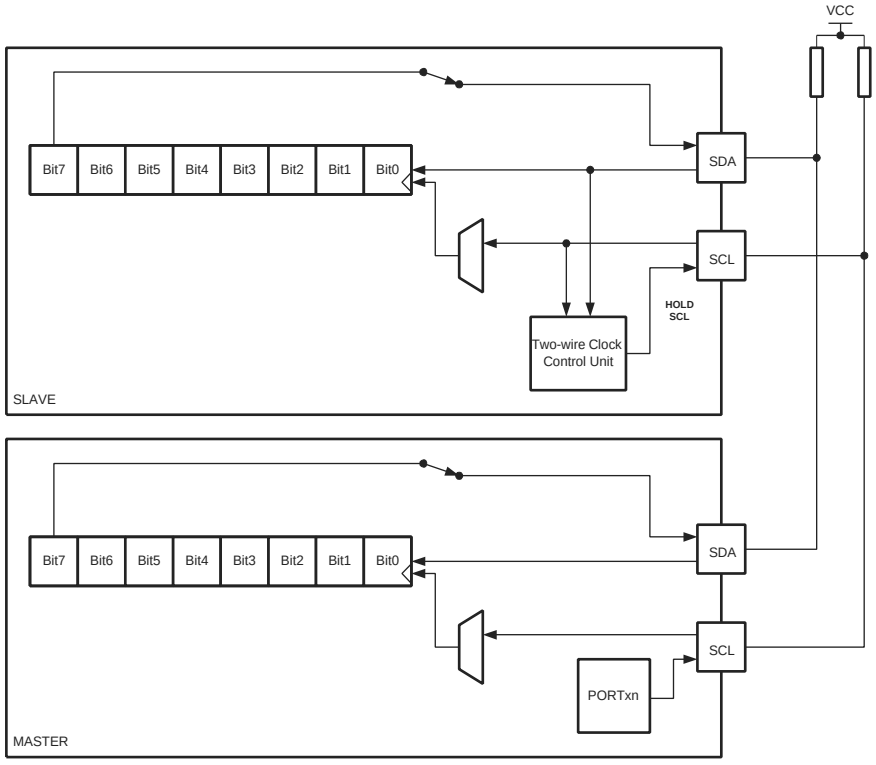


Figure 15-4 shows two USI units operating in two-wire mode, one as master and one as slave. It is only the physical layer that is shown since the system operation is highly dependent of the communication scheme used. The main differences between the master and slave operation at this level is the serial clock generation which is always done by the master. Only the slave uses the clock control unit.

Clock generation must be implemented in software, but the shift operation is done automatically in both devices. Note that clocking only on negative edges for shifting data is of practical use in this mode. The slave can insert wait states at start or end of transfer by forcing the SCL clock low. This means that the master must always check if the SCL line was actually released after it has generated a positive edge.

Table 15-2 shows the relationship between the USICS[1:0] and USICLK setting and clock source used for the USI Data Register and the 4-bit counter.

Table 15-2. Relationship between the USICS[1:0] and USICLK Setting

USICS1	USICS0	USICLK	Clock Source	4-bit Counter Clock Source
0	0	0	No Clock	No Clock
0	0	1	Software clock strobe (USICLK)	Software clock strobe (USICLK)
0	1	X	Timer/Counter0 Compare Match	Timer/Counter0 Compare Match
1	0	0	External, positive edge	External, both edges
1	1	0	External, negative edge	External, both edges
1	0	1	External, positive edge	Software clock strobe (USITC)
1	1	1	External, negative edge	Software clock strobe (USITC)

- **Bit 1 – USICLK: Clock Strobe**

Writing a one to this bit location strobes the USI Data Register to shift one step and the counter to increment by one, provided that the software clock strobe option has been selected by writing USICS[1:0] bits to zero. The output will change immediately when the clock strobe is executed, i.e., during the same instruction cycle. The value shifted into the USI Data Register is sampled the previous instruction cycle.

When an external clock source is selected (USICS1 = 1), the USICLK function is changed from a clock strobe to a Clock Select Register. Setting the USICLK bit in this case will select the USITC strobe bit as clock source for the 4-bit counter (see Table 15-2).

The bit will be read as zero.

- **Bit 0 – USITC: Toggle Clock Port Pin**

Writing a one to this bit location toggles the USCK/SCL value either from 0 to 1, or from 1 to 0. The toggling is independent of the setting in the Data Direction Register, but if the PORT value is to be shown on the pin the corresponding DDR pin must be set as output (to one). This feature allows easy clock generation when implementing master devices.

When an external clock source is selected (USICS1 = 1) and the USICLK bit is set to one, writing to the USITC strobe bit will directly clock the 4-bit counter. This allows an early detection of when the transfer is done when operating as a master device.

The bit will read as zero.

is used for reading the internal temperature sensor, as described in Section 17.12 on page 133. A good system design with properly placed, external bypass capacitors does reduce the need for using ADC Noise Reduction Mode

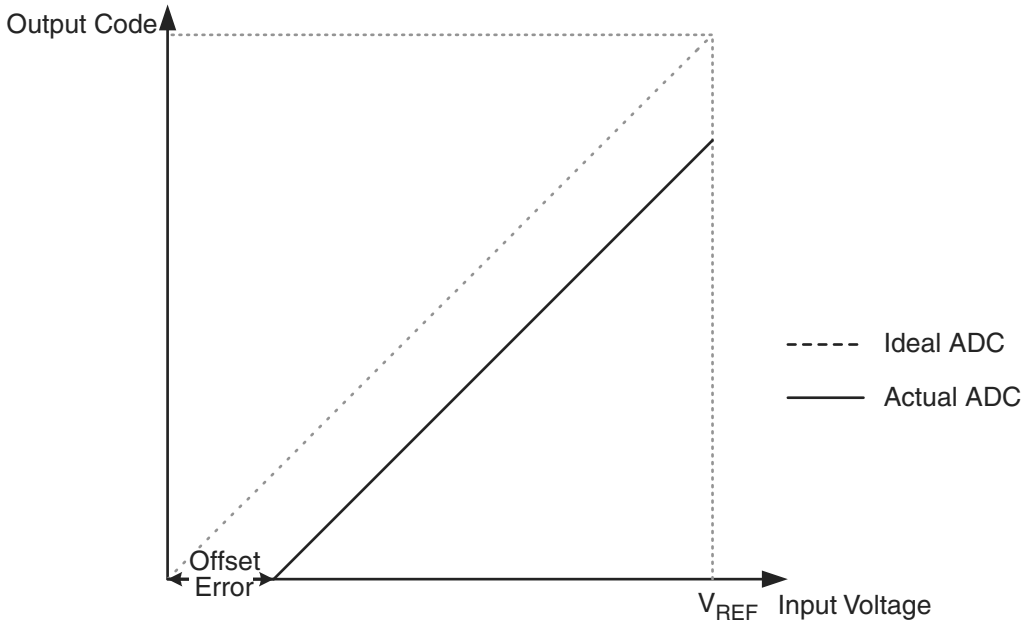
17.10 ADC Accuracy Definitions

An n-bit single-ended ADC converts a voltage linearly between GND and V_{REF} in 2^n steps (LSBs). The lowest code is read as 0, and the highest code is read as 2^n-1.

Several parameters describe the deviation from the ideal behavior, as follows:

- Offset: The deviation of the first transition (0x000 to 0x001) compared to the ideal transition (at 0.5 LSB). Ideal value: 0 LSB.

Figure 17-9. Offset Error



- Gain Error: After adjusting for offset, the Gain Error is found as the deviation of the last transition (0x3FE to 0x3FF) compared to the ideal transition (at 1.5 LSB below maximum). Ideal value: 0 LSB

17.13.2 ADCSRA – ADC Control and Status Register A

Bit	7	6	5	4	3	2	1	0	
0x06	**ADEN**	**ADSC**	**ADATE**	**ADIF**	**ADIE**	**ADPS2**	**ADPS1**	**ADPS0**	ADCSRA
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 7 – ADEN: ADC Enable**

Writing this bit to one enables the ADC. By writing it to zero, the ADC is turned off. Turning the ADC off while a conversion is in progress, will terminate this conversion.

- **Bit 6 – ADSC: ADC Start Conversion**

In Single Conversion mode, write this bit to one to start each conversion. In Free Running mode, write this bit to one to start the first conversion. The first conversion after ADSC has been written after the ADC has been enabled, or if ADSC is written at the same time as the ADC is enabled, will take 25 ADC clock cycles instead of the normal 13. This first conversion performs initialization of the ADC.

ADSC will read as one as long as a conversion is in progress. When the conversion is complete, it returns to zero. Writing zero to this bit has no effect.

- **Bit 5 – ADATE: ADC Auto Trigger Enable**

When this bit is written to one, Auto Triggering of the ADC is enabled. The ADC will start a conversion on a positive edge of the selected trigger signal. The trigger source is selected by setting the ADC Trigger Select bits, ADTS in ADCSRB.

- **Bit 4 – ADIF: ADC Interrupt Flag**

This bit is set when an ADC conversion completes and the data registers are updated. The ADC Conversion Complete Interrupt is executed if the ADIE bit and the I-bit in SREG are set. ADIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ADIF is cleared by writing a logical one to the flag. Beware that if doing a Read-Modify-Write on ADCSRA, a pending interrupt can be disabled. This also applies if the SBI and CBI instructions are used.

- **Bit 3 – ADIE: ADC Interrupt Enable**

When this bit is written to one and the I-bit in SREG is set, the ADC Conversion Complete Interrupt is activated.

- **Bits 2:0 – ADPS[2:0]: ADC Prescaler Select Bits**

These bits determine the division factor between the system clock frequency and the input clock to the ADC.

Table 17-5. ADC Prescaler Selections

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

20.5.2 Serial Programming Instruction set

Table 20-11. Minimum Wait Delay Before Writing the Next Flash or EEPROM Location

Symbol	Minimum Wait Delay
t_{WD_FLASH}	4.5 ms
t_{WD_EEPROM}	4.0 ms
t_{WD_ERASE}	9.0 ms
t_{WD_FUSE}	4.5 ms

Table 20-12 on page 153 and Figure 20-2 on page 154 describes the Instruction set.

Table 20-12. Serial Programming Instruction Set

	Instruction Format			
Instruction/Operation	**Byte 1**	**Byte 2**	**Byte 3**	**Byte4**
Programming Enable	$AC	$53	$00	$00
Chip Erase (Program Memory/EEPROM)	$AC	$80	$00	$00
Poll RDY/$\overline{\text{BSY}}$	$F0	$00	$00	data byte out
Load Instructions				
Load Extended Address byte(1)	$4D	$00	Extended adr	$00
Load Program Memory Page, High byte	$48	adr MSB	adr LSB	high data byte in
Load Program Memory Page, Low byte	$40	adr MSB	adr LSB	low data byte in
Load EEPROM Memory Page (page access)	$C1	$00	0000 000aa	data byte in
Read Instructions				
Read Program Memory, High byte	$28	adr MSB	adr LSB	high data byte out
Read Program Memory, Low byte	$20	adr MSB	adr LSB	low data byte out
Read EEPROM Memory	$A0	$00	00aa aaaa	data byte out
Read Lock bits	$58	$00	$00	data byte out
Read Signature Byte	$30	$00	0000 000aa	data byte out
Read Fuse bits	$50	$00	$00	data byte out
Read Fuse High bits	$58	$08	$00	data byte out
Read Extended Fuse Bits	$50	$08	$00	data byte out
Read Calibration Byte	$38	$00	$00	data byte out
Write Instructions(6)				
Write Program Memory Page	$4C	adr MSB	adr LSB	$00
Write EEPROM Memory	$C0	$00	00aa aaaa	data byte in
Write EEPROM Memory Page (page access)	$C2	$00	00aa aa00	$00
Write Lock bits	$AC	$E0	$00	data byte in
Write Fuse bits	$AC	$A0	$00	data byte in
Write Fuse High bits	$AC	$A8	$00	data byte in
Write Extended Fuse Bits	$AC	$A4	$00	data byte in

Notes: 1. Not all instructions are applicable for all parts.

2. a = address
3. Bits are programmed ‘0’, unprogrammed ‘1’.
4. To ensure future compatibility, unused Fuses and Lock bits should be unprogrammed (‘1’) .
5. Refer to the correspondig section for Fuse and Lock bits, Calibration and Signature bytes and Page size.
6. Instructions accessing program memory use a word address. This address may be random within the page range.
7. See htt://www.atmel.com/avr for Application Notes regarding programming and programmers.

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

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

After data is loaded to the page buffer, program the EEPROM page, see Figure 20-2 on page 154.

Figure 20-2. Serial Programming Instruction example

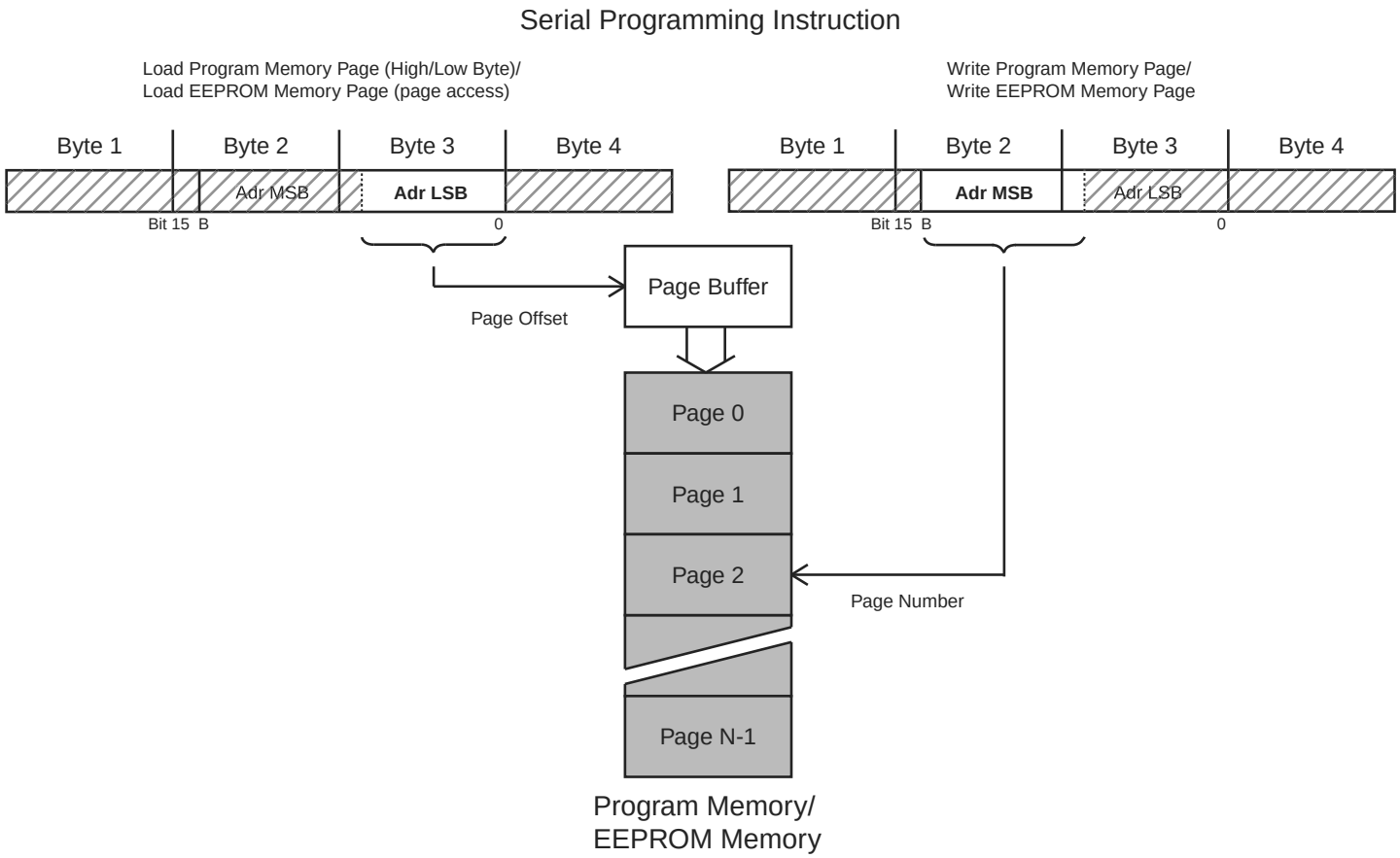


Figure 22-2. Active Supply Current vs. Frequency (1 - 20 MHz)

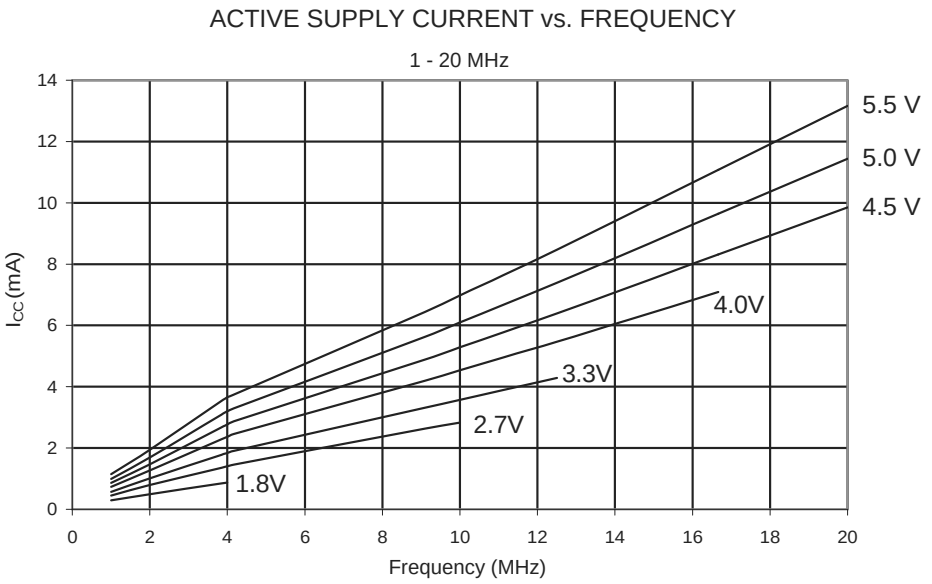


Figure 22-3. Active Supply Current vs. V_{CC} (Internal RC oscillator, 8 MHz)

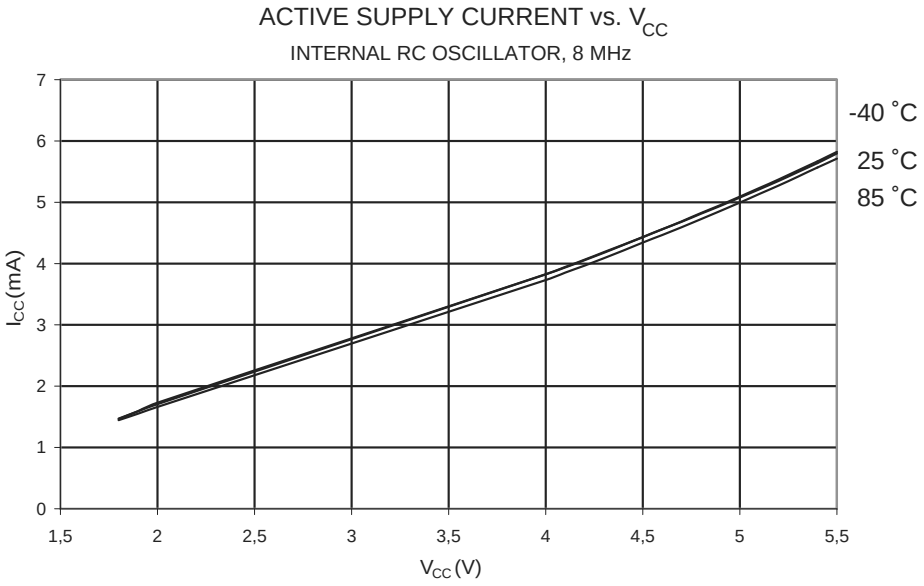
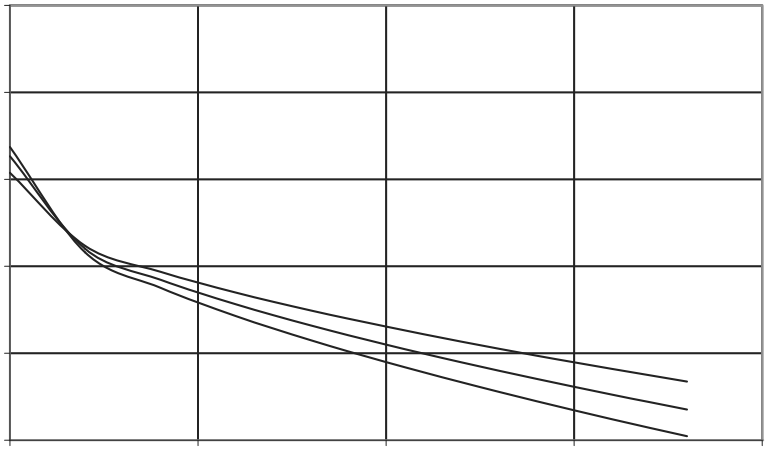
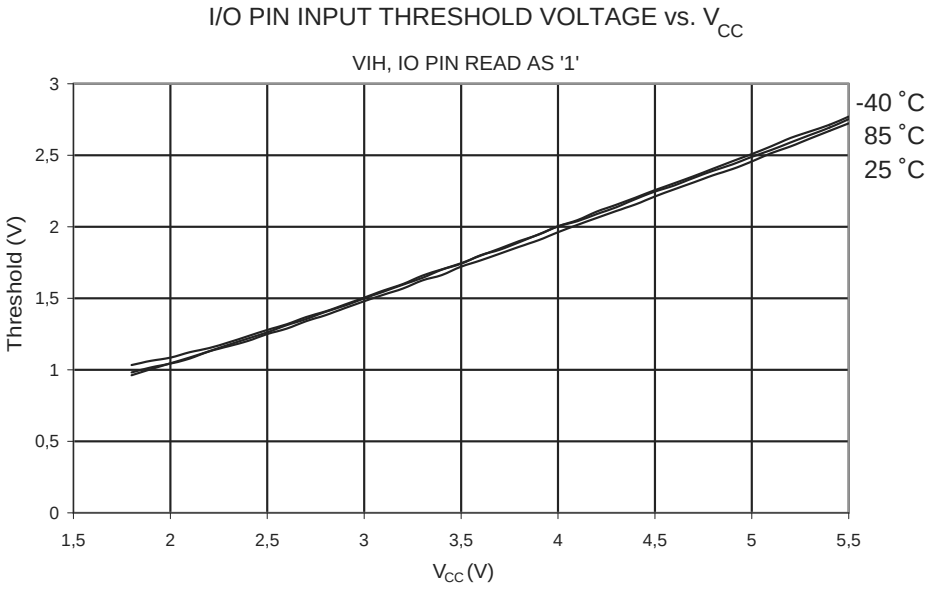


Figure 22-26. Reset Pin Output Voltage vs. Source Current (V_{CC} = 5V)



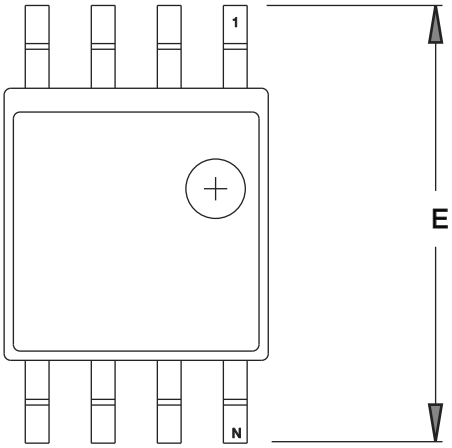
22.7 Pin Threshold and Hysteresis

Figure 22-27. I/O Pin Input Threshold Voltage vs. V_{CC} (V_{IH}, IO Pin Read as '1')

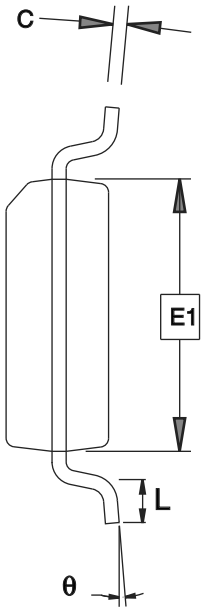


Mnemonics	Operands	Description	Operation	Flags	#Clocks
SWAP	Rd	Swap Nibbles	Rd(3..0)←Rd(7..4),Rd(7..4)←Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Twos Complement Overflow.	V ← 1	V	1
CLV		Clear Twos Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half Carry Flag in SREG	H ← 1	H	1
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1
DATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
MOVW	Rd, Rr	Copy Register Word	Rd+1:Rd ← Rr+1:Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LD	Rd, X	Load Indirect	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	Rd ← (X), X ← X + 1	None	2
LD	Rd, - X	Load Indirect and Pre-Dec.	X ← X - 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, - Y	Load Indirect and Pre-Dec.	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd,Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z), Z ← Z+1	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
LDS	Rd, k	Load Direct from SRAM	Rd ← (k)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr, X ← X + 1	None	2
ST	- X, Rr	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr, Y ← Y + 1	None	2
ST	- Y, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q,Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr, Z ← Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q,Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
STS	k, Rr	Store Direct to SRAM	(k) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
LPM	Rd, Z	Load Program Memory	Rd ← (Z)	None	3
LPM	Rd, Z+	Load Program Memory and Post-Inc	Rd ← (Z), Z ← Z+1	None	3
SPM		Store Program Memory	(z) ← R1:R0	None	
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
MCU CONTROL INSTRUCTIONS					
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	1
WDR		Watchdog Reset	(see specific descr. for WDR/Timer)	None	1
BREAK		Break	For On-chip Debug Only	None	N/A

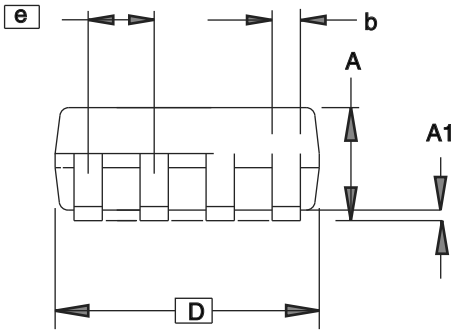
26.2 8S2



TOP VIEW



END VIEW



SIDE VIEW

COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	1.70		2.16	
A1	0.05		0.25	
b	0.35		0.48	4
C	0.15		0.35	4
D	5.13		5.35	
E1	5.18		5.40	2
E	7.70		8.26	
L	0.51		0.85	
θ	0°		8°	
e	1.27 BSC			3

Notes: 1. This drawing is for general information only; refer to EIAJ Drawing EDR-7320 for additional information.
2. Mismatch of the upper and lower dies and resin burrs aren't included.
3. Determines the true geometric position.
4. Values b,C apply to plated terminal. The standard thickness of the plating layer shall measure between 0.007 to .021 mm.

4/15/08

Package Drawing Contact: packagedrawings@atmel.com	TITLE	GPC	DRAWING NO.	REV.
	8S2, 8-lead, 0.208” Body, Plastic Small Outline Package (EIAJ)	STN	8S2	F

28. Datasheet Revision History

28.1 Rev. 2586Q-08/13

1. “Bit 3 – FOC1B: Force Output Compare Match 1B” description in “GTCCR – General Timer/Counter1 Control Register” on page 90 updated: PB3 in “compare match output pin PB3 (OC1B)” corrected to PB4.

28.2 Rev. 2586P-06/13

1. Updated description of “EEARH – EEPROM Address Register” and “EEARL – EEPROM Address Register” on page 20.

28.3 Rev. 2586O-02/13

Updated ordering codes on page 204, page 205, and page 206.

28.4 Rev. 2586N-04/11

1. Added:
   – Section “Capacitive Touch Sensing” on page 6.
2. Updated:
   – Document template.
   – Removed “Preliminary” on front page. All devices now final and in production.
   – Section “Limitations” on page 36.
   – Program example on page 49.
   – Section “Overview” on page 122.
   – Table 17-4 on page 135.
   – Section “Limitations of debugWIRE” on page 140.
   – Section “Serial Programming Algorithm” on page 151.
   – Table 21-7 on page 166.
   – EEPROM errata on pages 212, 212, 213, 214, and 215
   – Ordering information on pages 204, 205, and 206.

28.5 Rev. 2586M-07/10

1. Clarified Section 6.4 “Clock Output Buffer” on page 31.
2. Added Ordering Codes -SN and -SNR for ATtiny25 extended temperature.

28.6 Rev. 2586L-06/10

1. Added:
   – TSSOP for ATtiny45 in “Features” on page 1, Pinout Figure 1-1 on page 2, Ordering Information in Section 25.2 “ATtiny45” on page 205, and Packaging Information in Section 26.4 “8X” on page 210
   – Table 6-11, “Capacitance of Low-Frequency Crystal Oscillator,” on page 29
   – Figure 22-36 on page 191 and Figure 22-37 on page 191, Typical Characteristics plots for Bandgap Voltage vs. V_{CC} and Temperature
   – Extended temperature in Section 25.1 “ATtiny25” on page 204, Ordering Information
   – Tape & reel part numbers in Ordering Information, in Section 25.1 “ATtiny25” on page 204 and Section 25.2 “ATtiny45” on page 205

2. Updated:
   – “Features” on page 1, removed Preliminary from ATtiny25
   – Section 8.4.2 “Code Example” on page 44
   – “PCMSK – Pin Change Mask Register” on page 52, Bit Descriptions
   – “TCCR1 – Timer/Counter1 Control Register” on page 89 and “GTCCR – General Timer/Counter1 Control Register” on page 90, COM bit descriptions clarified
   – Section 20.3.2 “Calibration Bytes” on page 150, frequencies (8 MHz, 6.4 MHz)
   – Table 20-11, “Minimum Wait Delay Before Writing the Next Flash or EEPROM Location,” on page 153, value for t_{WD_ERASE}
   – Table 20-16, “High-voltage Serial Programming Instruction Set for ATtiny25/45/85,” on page 158
   – Table 21-1, “DC Characteristics. TA = -40°C to +85°C,” on page 161, notes adjusted
   – Table 21-11, “Serial Programming Characteristics, TA = -40°C to +85°C, VCC = 1.8 - 5.5V (Unless Otherwise Noted),” on page 170, added t_{SLIV}
   – Bit syntax throughout the datasheet, e.g. from CS02:0 to CS0[2:0].

28.7 Rev. 2586K-01/08

1. Updated Document Template.
2. Added Sections:
   – “Data Retention” on page 6
   – “Low Level Interrupt” on page 49
   – “Device Signature Imprint Table” on page 149
3. Updated Sections:
   – “Internal PLL for Fast Peripheral Clock Generation - clkPCK” on page 24
   – “System Clock and Clock Options” on page 23
   – “Internal PLL in ATtiny15 Compatibility Mode” on page 24
   – “Sleep Modes” on page 34
   – “Software BOD Disable” on page 35
   – “External Interrupts” on page 49
   – “Timer/Counter1 in PWM Mode” on page 97
   – “USI – Universal Serial Interface” on page 108
   – “Temperature Measurement” on page 133
   – “Reading Lock, Fuse and Signature Data from Software” on page 143
   – “Program And Data Memory Lock Bits” on page 147
   – “Fuse Bytes” on page 148
   – “Signature Bytes” on page 150
   – “Calibration Bytes” on page 150
   – “System and Reset Characteristics” on page 165
4. Added Figures:
   – “Reset Pin Output Voltage vs. Sink Current (VCC = 3V)” on page 184
   – “Reset Pin Output Voltage vs. Sink Current (VCC = 5V)” on page 185
   – “Reset Pin Output Voltage vs. Source Current (VCC = 3V)” on page 185
   – “Reset Pin Output Voltage vs. Source Current (VCC = 5V)” on page 186