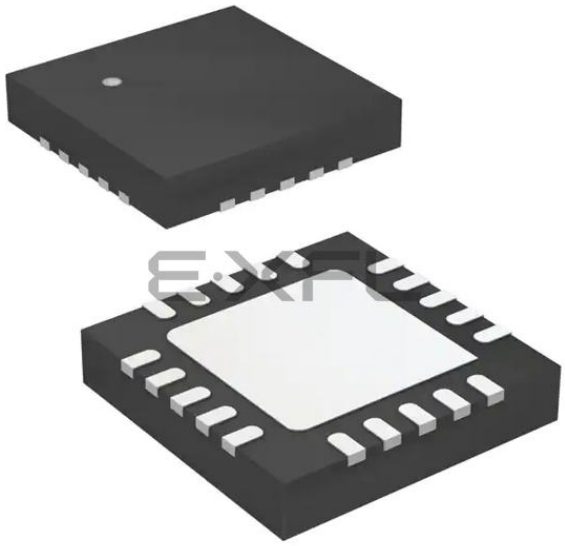


Welcome to E-XFL.COM

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	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 ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	20-WFQFN Exposed Pad
Supplier Device Package	20-QFN-EP (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/attiny85-15mt1

4.4 I/O Memory

The I/O space definition of the ATtiny25/45/85 is shown in Section "" on page 164.

All ATtiny25/45/85 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.

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 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.

5.1.4 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.

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

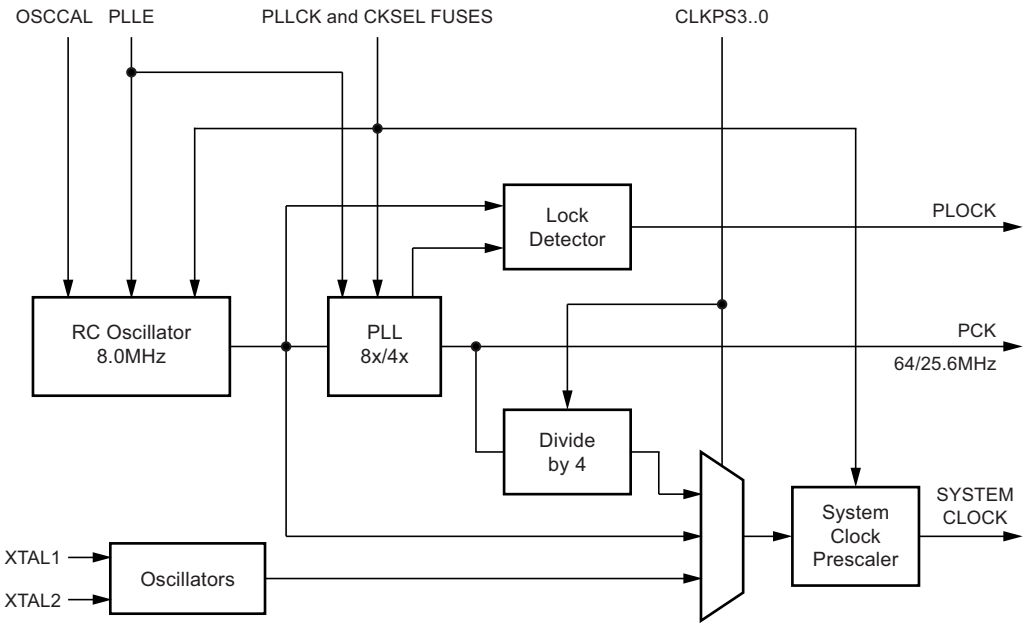
The internal PLL in ATtiny25/45/85 generates a clock frequency that is 8x multiplied from a source input. The source of the PLL input clock is the output of the internal RC oscillator having a frequency of 8.0MHz. Thus the output of the PLL, the fast peripheral clock is 64MHz. The fast peripheral clock, or a clock prescaled from that, can be selected as the clock source for Timer/Counter1. See the Figure 5-2.

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 8MHz, the fast peripheral clock frequency saturates at 85MHz (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 8MHz in order to keep the PLL in the correct operating range. The internal PLL is enabled only when the PLLE bit in PLLCSR is set or the PLLCK fuse is programmed (‘0’). The bit PLOCK from PLLCSR is set when PLL is locked.

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

Figure 5-2. PCK Clocking System



5.2 Clock Sources

The device has the following clock source options, selectable by flash fuse bits as shown below. The clock from the selected source is input to the AVR® clock generator, and routed to the appropriate modules.

Table 5-1. Device Clocking Options Select(1)

Device Clocking Option	CKSEL3..0
External clock	0000
PLL clock	0001
Calibrated internal RC oscillator 8.0MHz	0010
Watchdog oscillator 128kHz	0100
External low-frequency crystal	0110
External crystal/ceramic resonator	1000-1111
Reserved	0101, 0111, 0011

Note: 1. For all fuses “1” means unprogrammed while “0” means programmed.

The various choices for each clocking option is given in the following sections. When the CPU wakes up from power-down or power-save, the selected clock source is used to time the start-up, ensuring stable oscillator operation before instruction execution starts. When the CPU starts from reset, there is an additional delay allowing the power to reach a stable level before commencing normal operation. The watchdog oscillator is used for timing this real-time part of the start-up time. The number of WDT oscillator cycles used for each time-out is shown in Table 5-2.

Table 5-2. Number of Watchdog Oscillator Cycles

Typ Time-out	Number of Cycles
4ms	512
64ms	8K (8,192)

5.3 Default Clock Source

The device is shipped with CKSEL = “0010”, SUT = “10”, and CKDIV8 programmed. The default clock source setting is therefore the internal RC oscillator running at 8MHz with longest start-up time and an initial system clock prescaling of 8. This default setting ensures that all users can make their desired clock source setting using an in-system or high-voltage programmer.

5.4 Crystal Oscillator

XTAL1 and XTAL2 are input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 5-3. Either a quartz crystal or a ceramic resonator may be used.

C1 and C2 should always be equal for both crystals and resonators. The optimal value of the capacitors depends on the crystal or resonator in use, the amount of stray capacitance, and the electromagnetic noise of the environment. Some initial guidelines for choosing capacitors for use with crystals are given in Table 5-3. For ceramic resonators, the capacitor values given by the manufacturer should be used.

6. Power Management and Sleep Modes

The high performance and industry leading code efficiency makes the AVR® microcontrollers an ideal choice for low power applications.

Sleep modes enable the application to shut down unused modules in the MCU, thereby saving power. The AVR provides various sleep modes allowing the user to tailor the power consumption to the application's requirements.

To enter any of the three sleep modes, the SE bit in MCUCR must be written to logic one and a SLEEP instruction must be executed. The SM1..0 bits in the MCUCR register select which sleep mode (idle, ADC noise reduction, or power-down) will be activated by the SLEEP instruction. See Table 6-1 for a summary. If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU wakes up. The MCU is then halted for four cycles in addition to the start-up time, executes the interrupt routine, and resumes execution from the instruction following SLEEP. The contents of the register file and SRAM are unaltered when the device wakes up from sleep. If a reset occurs during sleep mode, the MCU wakes up and executes from the reset vector.

Figure 5-1 on page 19 presents the different clock systems in the Atmel ATtiny25/45/85, and their distribution. The figure is helpful in selecting an appropriate sleep mode.

6.1 MCU Control Register – MCUCR

The MCU control register contains control bits for power management.

Bit	7	6	5	4	3	2	1	0	
	BODS	PUD	SE	SM1	SM0	BODSE	ISC01	ISC00	MCUCR
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 – BODS: BOD Sleep**

BOD disable functionality is available in some devices, only. See Section 6.5 "Limitations" on page 29.

In order to disable BOD during sleep (see Table 6-2 on page 29) the BODS bit must be written to logic one. This is controlled by a timed sequence and the enable bit, BODSE in MCUCR. First both BODS and BODSE must be set to one. Second, within four clock cycles, BODS must be set to one and BODSE must be set to zero. The BODS bit is active three clock cycles after it is set. A sleep instruction must be executed while BODS is active in order to turn off the BOD for the actual sleep mode. The BODS bit is automatically cleared after three clock cycles.

In devices where sleeping BOD has not been implemented this bit is unused and will always read zero.

- **Bit 5 – SE: Sleep Enable**

The SE bit must be written to logic one to make the MCU enter the sleep mode when the SLEEP instruction is executed. To avoid the MCU entering the sleep mode unless it is the programmer's purpose, it is recommended to write the sleep enable (SE) bit to one just before the execution of the SLEEP instruction and to clear it immediately after waking up.

- **Bits 4, 3 – SM1..0: Sleep Mode Select Bits 2..0**

These bits select between the three available sleep modes as shown in Table 6-1.

Table 6-1. Sleep Mode Select

SM1	SM0	Sleep Mode
0	0	Idle
0	1	ADC noise reduction
1	0	Power-down
1	1	Stand-by mode

- **Bit 2 – BODSE: BOD Sleep Enable**

BOD disable functionality is available in some devices, only. See Section 6.5 "Limitations" on page 29.

The BODSE bit enables setting of BODS control bit, as explained on BODS bit description. BOD disable is controlled by a timed sequence.

This bit is unused in devices where software BOD disable has not been implemented and will read as zero in those devices.

9.2.1 Configuring the Pin

Each port pin consists of three register bits: DDxn, PORTxn, and PINxn. As shown in Section 9.4 “Register Description for I/O-Ports” on page 53, the DDxn bits are accessed at the DDRx I/O address, the PORTxn bits at the PORTx I/O address, and the PINxn bits at the PINx I/O address.

The DDxn bit in the DDRx register selects the direction of this pin. If DDxn is written logic one, Pxn is configured as an output pin. If DDxn is written logic zero, Pxn is configured as an input pin.

If PORTxn is written logic one when the pin is configured as an input pin, the pull-up resistor is activated. To switch the pull-up resistor off, PORTxn has to be written logic zero or the pin has to be configured as an output pin. The port pins are tri-stated when reset condition becomes active, even if no clocks are running.

If PORTxn is written logic one when the pin is configured as an output pin, the port pin is driven high (one). If PORTxn is written logic zero when the pin is configured as an output pin, the port pin is driven low (zero).

9.2.2 Toggling the Pin

Writing a logic one to PINxn toggles the value of PORTxn, independent on the value of DDRxn. Note that the SBI instruction can be used to toggle one single bit in a port.

9.2.3 Switching Between Input and Output

When switching between tri-state ({DDxn, PORTxn} = 0b00) and output high ({DDxn, PORTxn} = 0b11), an intermediate state with either pull-up enabled {DDxn, PORTxn} = 0b01) or output low ({DDxn, PORTxn} = 0b10) must occur. Normally, the pull-up enabled state is fully acceptable, as a high-impedant environment will not notice the difference between a strong high driver and a pull-up. If this is not the case, the PUD bit in the MCUCR Register can be set to disable all pull-ups in all ports.

Switching between input with pull-up and output low generates the same problem. The user must use either the tri-state ({DDxn, PORTxn} = 0b00) or the output high state ({DDxn, PORTxn} = 0b10) as an intermediate step.

Table 9-1 summarizes the control signals for the pin value.

Table 9-1. Port Pin Configurations

DDxn	PORTxn	PUD (in MCUCR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output low (sink)
1	1	X	Output	No	Output high (source)

In fast PWM mode, the compare unit allows generation of PWM waveforms on the OC0x pins. Setting the COM0x1:0 bits to two will produce a non-inverted PWM and an inverted PWM output can be generated by setting the COM0x1:0 to three: Setting the COM0A1:0 bits to one allows the AC0A pin to toggle on compare matches if the WGM02 bit is set. This option is not available for the OC0B pin (See Table 11-3 on page 67). The actual OC0x value will only be visible on the port pin if the data direction for the port pin is set as output. The PWM waveform is generated by setting (or clearing) the OC0x register at the compare match between OCR0x and TCNT0, and clearing (or setting) the OC0x register at the timer clock cycle the counter is cleared (changes from TOP to BOTTOM).

The PWM frequency for the output can be calculated by the following equation:

$$f_{OCnxPWM} = \frac{f_{clk\_I/O}}{N \times 256}$$

The *N* variable represents the prescale factor (1, 8, 64, 256, or 1024).

The extreme values for the OCR0A register represents special cases when generating a PWM waveform output in the fast PWM mode. If the OCR0A is set equal to BOTTOM, the output will be a narrow spike for each MAX+1 timer clock cycle.

Setting the OCR0A equal to MAX will result in a constantly high or low output (depending on the polarity of the output set by the COM0A1:0 bits.)

A frequency (with 50% duty cycle) waveform output in fast PWM mode can be achieved by setting OC0x to toggle its logical level on each compare match (COM0x1:0 = 1). The waveform generated will have a maximum frequency of $f_{OC0} = f_{clk\_I/O}/2$ when OCR0A is set to zero. This feature is similar to the OC0A toggle in CTC mode, except the double buffer feature of the output compare unit is enabled in the fast PWM mode.

11.6.4 Phase Correct PWM Mode

The phase correct PWM mode (WGM02:0 = 1 or 5) provides a high resolution phase correct PWM waveform generation option. The phase correct PWM mode is based on a dual-slope operation. The counter counts repeatedly from BOTTOM to TOP and then from TOP to BOTTOM. TOP is defined as 0xFF when WGM2:0 = 1, and OCR0A when WGM2:0 = 5.
In non-inverting compare output mode, the output compare (OC0x) is cleared on the compare match between TCNT0 and OCR0x while up counting, and set on the compare match while down-counting. In inverting output compare mode, the operation is inverted. The dual-slope operation has lower maximum operation frequency than single slope operation. However, due to the symmetric feature of the dual-slope PWM modes, these modes are preferred for motor control applications.

In phase correct PWM mode the counter is incremented until the counter value matches TOP. When the counter reaches TOP, it changes the count direction. The TCNT0 value will be equal to TOP for one timer clock cycle. The timing diagram for the phase correct PWM mode is shown on Figure 11-7 on page 64. The TCNT0 value is in the timing diagram shown as a histogram for illustrating the dual-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT0 slopes represent compare matches between OCR0x and TCNT0.

13.1.1 Timer/Counter1 Control Register - TCCR1

Bit	7	6	5	4	3	2	1	0	
$30 ($50)	CTC1	PWM1A	COM1A1	COM1A0	CS13	CS12	CS11	CS10	TCCR1
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- CTC1: Clear Timer/Counter on Compare Match**

When the CTC1 control bit is set (one), Timer/Counter1 is reset to $00 in the CPU clock cycle after a compare match with OCR1C register value. If the control bit is cleared, Timer/Counter1 continues counting and is unaffected by a compare match.

- **Bit 6- PWM1A: Pulse Width Modulator A Enable**

When set (one) this bit enables PWM mode based on comparator OCR1A in Timer/Counter1 and the counter value is reset to $00 in the CPU clock cycle after a compare match with OCR1C register value.

- **Bits 5,4 - COM1A1, COM1A0: Comparator A Output Mode, Bits 1 and 0**

The COM1A1 and COM1A0 control bits determine any output pin action following a compare match with compare register A in Timer/Counter1. Output pin actions affect pin PB1 (OC1A). Since this is an alternative function to an I/O port, the corresponding direction control bit must be set (one) in order to control an output pin. Note that $\overline{\text{OC1A}}$ is not connected in normal mode.

Table 13-1. Comparator A Mode Select

COM1A1	COM1A0	Description
0	0	Timer/Counter comparator A disconnected from output pin OC1A.
0	1	Toggle the OC1A output line.
1	0	Clear the OC1A output line.
1	1	Set the OC1A output line

In PWM mode, these bits have different functions. Refer to Table 13-4 on page 83 for a detailed description.

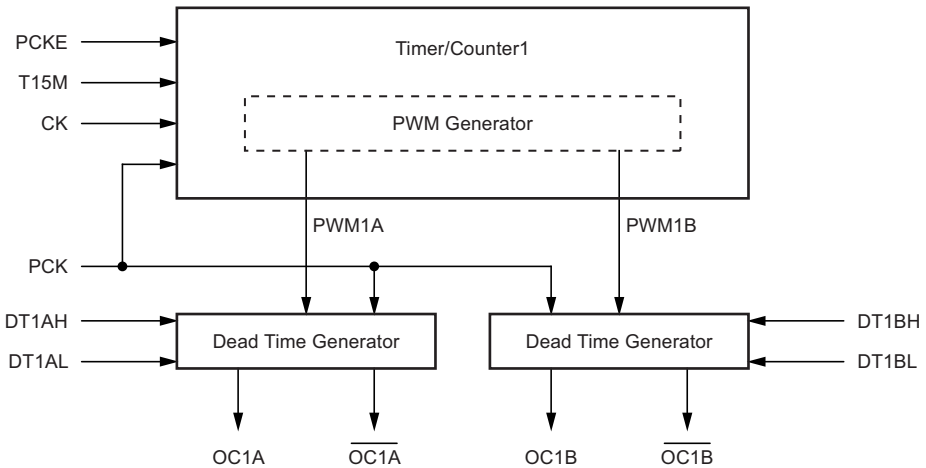
- **Bits 3..0 - CS13, CS12, CS11, CS10: Clock Select Bits 3, 2, 1, and 0**

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

14. Dead Time Generator

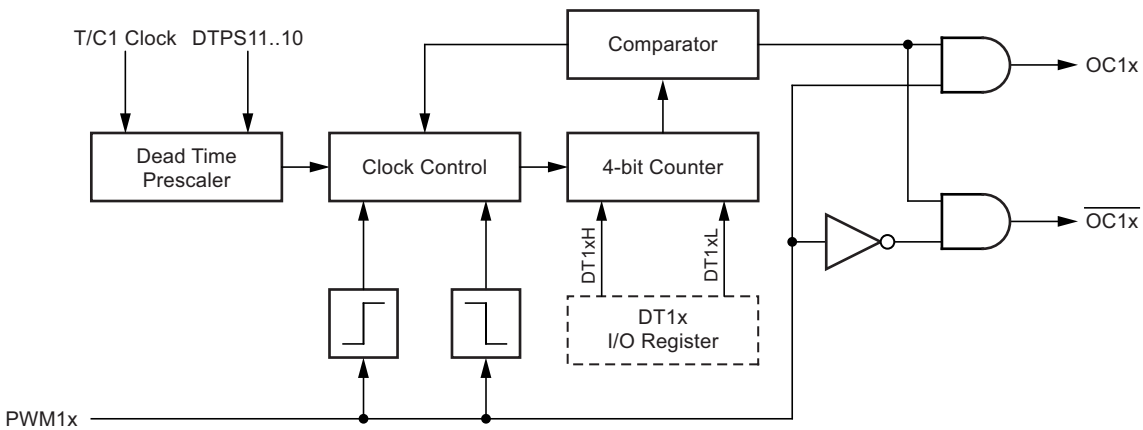
The dead time generator is provided for the Timer/Counter1 PWM output pairs to allow driving external power control switches safely. The dead time generator is a separate block that can be connected to Timer/Counter1 and it is used to insert dead times (non-overlapping times) for the Timer/Counter1 complementary output pairs (OC1A-$\overline{\text{OC1A}}$ and OC1B-$\overline{\text{OC1B}}$). The sharing of tasks is as follows: the Timer/Counter generates the PWM output and the dead time generator generates the non-overlapping PWM output pair from the Timer/Counter PWM signal. Two dead time generators are provided, one for each PWM output. The non-overlap time is adjustable and the PWM output and it's complementary output are adjusted separately, and independently for both PWM outputs.

Figure 14-1. Timer/Counter1 and Dead Time Generators



The dead time generation is based on the 4-bit down counters that count the dead time, as shown in Figure 14-1 There is a dedicated prescaler in front of the dead time generator that can divide the Timer/Counter1 clock (PCK or CK) by 1, 2, 4 or 8. This provides for large range of dead times that can be generated. The prescaler is controlled by two control bits DTPS11..10 from the I/O register at address 0x23. The block has also a rising and falling edge detector that is used to start the dead time counting period. Depending on the edge, one of the transitions on the rising edges, OC1x or $\overline{\text{OC1x}}$ is delayed until the counter has counted to zero. The comparator is used to compare the counter with zero and stop the dead time insertion when zero has been reached. The counter is loaded with a 4-bit DT1xH or DT1xL value from DT1x I/O register, depending on the edge of the PWM generator output when the dead time insertion is started.

Figure 14-2. Dead Time Generator



14.2 Timer/Counter1 Dead Time A - DT1A

Bit	7	6	5	4	3	2	1	0	
$25 ($45)	DT1AH3	DT1AH2	DT1AH1	DT1AH0	DT1AL3	DT1AL2	DT1AL1	DT1AL0	DT1A
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 dead time value register A is an 8-bit read/write register.

The dead time delay of is adjusted by the dead time value register, DT1A. The register consists of two fields, DT1AH3..0 and DT1AL3..0, one for each complementary output. Therefore a different dead time delay can be adjusted for the rising edge of OC1A and the rising edge of $\overline{OC1A}$.

- **Bits 7..4- DT1AH3..DT1AH0: Dead Time Value for OC1A Output**

The dead time value for the OC1A output. The dead time delay is set as a number of the prescaled Timer/Counter clocks. The minimum dead time is zero and the maximum dead time is the prescaled time/counter clock period multiplied by 15.

- **Bits 3..0- DT1AL3..DT1AL0: Dead Time Value for $\overline{OC1A}$ Output**

The dead time value for the $\overline{OC1A}$ output. The dead time delay is set as a number of the prescaled Timer/Counter clocks. The minimum dead time is zero and the maximum dead time is the prescaled time/counter clock period multiplied by 15.

14.3 Timer/Counter1 Dead Time B - DT1B

Bit	7	6	5	4	3	2	1	0	
$25 ($45)	DT1BH3	DT1BH2	DT1BH1	DT1BH0	DT1BL3	DT1BL2	DT1BL1	DT1BL0	DT1B
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 dead time value register bit an 8-bit read/write register.

The dead time delay of is adjusted by the dead time value register, DT1B. The register consists of two fields, DT1BH3..0 and DT1BL3..0, one for each complementary output. Therefore a different dead time delay can be adjusted for the rising edge of OC1A and the rising edge of $\overline{OC1A}$.

- **Bits 7..4- DT1BH3..DT1BH0: Dead Time Value for OC1B Output**

The dead time value for the OC1B output. The dead time delay is set as a number of the prescaled Timer/Counter clocks. The minimum dead time is zero and the maximum dead time is the prescaled time/counter clock period multiplied by 15.

- **Bits 3..0- DT1BL3..DT1BL0: Dead Time Value for $\overline{OC1B}$ Output**

The dead time value for the $\overline{OC1B}$ output. The dead time delay is set as a number of the prescaled Timer/Counter clocks. The minimum dead time is zero and the maximum dead time is the prescaled time/counter clock period multiplied by 15.

16.3 Analog Comparator Multiplexed Input

It is possible to select any of the ADC3..0 pins to replace the negative input to the analog comparator. The ADC multiplexer is used to select this input, and consequently, the ADC must be switched off to utilize this feature. If the analog comparator multiplexer enable bit (ACME in ADCSRB) is set and the ADC is switched off (ADEN in ADCSRA is zero), MUX1..0 in ADMUX select the input pin to replace the negative input to the analog comparator, as shown in Table 16-2. If ACME is cleared or ADEN is set, AIN1 is applied to the negative input to the analog comparator.

Table 16-2. Analog Comparator Multiplexed Input

ACME	ADEN	MUX1..0	Analog Comparator Negative Input
0	x	xx	AIN1
1	1	xx	AIN1
1	0	00	ADC0
1	0	01	ADC1
1	0	10	ADC2
1	0	11	ADC3

16.3.1 Digital Input Disable Register 0 – DIDR0

Bit	7	6	5	4	3	2	1	0	
	–	–	ADC0D	ADC2D	ADC3D	ADC1D	AIN1D	AIN0D	DIDR0
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 1, 0 – AIN1D, AIN0D: AIN1, AIN0 Digital Input Disable**

When this bit is written logic one, the digital input buffer on the AIN1/0 pin is disabled. The corresponding PIN register bit will always read as zero when this bit is set. When an analog signal is applied to the AIN1/0 pin and the digital input from this pin is not needed, this bit should be written logic one to reduce power consumption in the digital input buffer.

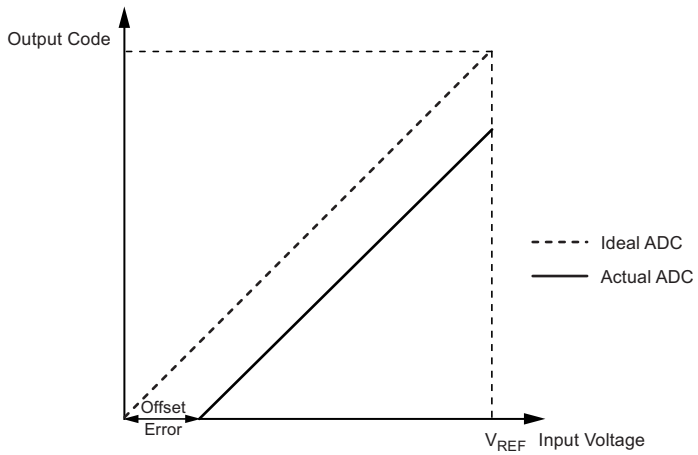
17.6.3 ADC Accuracy Definitions

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:

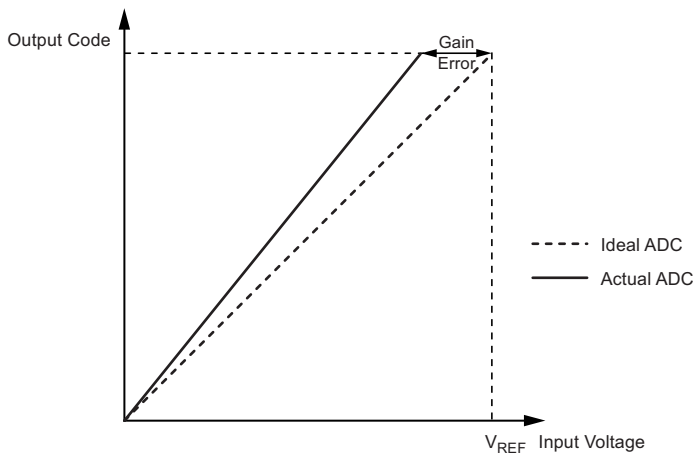
- Offset: The deviation of the first transition (0x000 to 0x001) compared to the ideal transition (at 0.5 LSB). Ideal value: 0 LSB.

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

Figure 17-10. Gain Error



19.4.1 Store Program Memory Control and Status Register – SPMCSR

The store program memory control and status register contains the control bits needed to control the program memory operations.

Bit	7	6	5	4	3	2	1	0	
	–	–	–	CTPB	RFLB	PGWRT	PGERS	SPMEN	SPMCSR
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..5 – Res: Reserved Bits**

These bits are reserved bits in the Atmel® ATtiny25/45/85 and always read as zero.

- **Bit 4 – CTPB: Clear Temporary Page Buffer**

If the CTPB bit is written while filling the temporary page buffer, the temporary page buffer will be cleared and the data will be lost.

- **Bit 3 – RFLB: Read Fuse and Lock Bits**

An LPM instruction within three cycles after RFLB and SPMEN are set in the SPMCSR register, will read either the lock bits or the fuse bits (depending on Z0 in the Z-pointer) into the destination register. See Section 19.4.2 “EEPROM Write Prevents Writing to SPMCSR” on page 121 for details.

- **Bit 2 – PGWRT: Page Write**

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

- **Bit 1 – PGERS: Page Erase**

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

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

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

Writing any other combination than “10001”, “01001”, “00101”, “00011” or “00001” in the lower five bits will have no effect.

19.4.2 EEPROM Write Prevents Writing to SPMCSR

Note that an EEPROM write operation will block all software programming to flash. Reading the fuses and lock bits from software will also be prevented during the EEPROM write operation. It is recommended that the user checks the status bit (EEWE) in the EECR register and verifies that the bit is cleared before writing to the SPMCSR register.

Table 20-5. Fuse Low Byte

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

Notes:
1. See Section 5.10 “System Clock Prescaler” on page 26 for details.
2. The CKOUT fuse allows the system clock to be output on PORTB4. See “Section 5.9 “Clock Output Buffer” on page 26 for details.
3. The default value of SUT1..0 results in maximum start-up time for the default clock source. See Table 5-7 on page 23 for details.
4. The default setting of CKSEL1..0 results in internal RC oscillator at 8.0MHz. See Table 5-6 on page 23 for details.

The status of the fuse bits is not affected by chip erase. Note that the fuse bits are locked if lock bit1 (LB1) is programmed. program the fuse bits before programming the lock bits.

20.2.1 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.

20.3 Signature Bytes

All Atmel® microcontrollers have a three-byte signature code which identifies the device. This code can be read in both serial and high-voltage programming mode, also when the device is locked. The three bytes reside in a separate address space.

20.3.1 ATtiny25 Signature Bytes

1. 0x000: 0x1E (indicates manufactured by Atmel).
2. 0x001: 0x91 (indicates 2KB flash memory).
3. 0x002: 0x08 (indicates ATtiny25 device when 0x001 is 0x91).

20.3.2 ATtiny45 Signature Bytes

1. 0x000: 0x1E (indicates manufactured by Atmel).
2. 0x001: 0x92 (indicates 4KB flash memory).
3. 0x002: 0x06 (indicates ATtiny45 device when 0x001 is 0x92).

20.3.3 ATtiny85 Signature Bytes

1. 0x000: 0x1E (indicates manufactured by Atmel).
2. 0x001: 0x93 (indicates 8KB flash memory).
3. 0x002: 0x0B (indicates ATtiny85 device when 0x001 is 0x93).

20.4 Calibration Byte

Signature area of the Atmel ATtiny25/45/85 has one byte of calibration data for the internal RC oscillator. This byte resides in the high byte of address 0x000. During reset, this byte is automatically written into the OSCCAL register to ensure correct frequency of the calibrated RC oscillator.

Table 20-10. Serial Programming Instruction Set (Continued)

Instruction	Instruction Format				Operation
	Byte 1	Byte 2	Byte 3	Byte4	
Write lock bits	1010 1100	111x xxxx	xxxx xxxx	11**ii iiii**	Write lock bits. Set bits = “0” to program lock bits. See Table 20-1 on page 123 for details.
Read signature byte	0011 0000	000x xxxx	xxxx xx**bb**	**oooo oooo**	Read signature byte **o** at address **b**.
Write fuse bits	1010 1100	1010 0000	xxxx xxxx	**iiii iiii**	Set bits = “0” to program, “1” to unprogram. See Table 20-5 on page 125 for details.
Write fuse high bits	1010 1100	1010 1000	xxxx xxxx	**iiii iiii**	Set bits = “0” to program, “1” to unprogram. See Table 20-4 on page 124 for details.
Write extended fuse bits	1010 1100	1010 0100	xxxx xxxx	**xxxx xxxi**	Set bits = “0” to program, “1” to unprogram. See Table 20-3 on page 124 for details.
Read fuse bits	0101 0000	0000 0000	xxxx xxxx	**oooo oooo**	Read fuse bits. “0” = programmed, “1” = unprogrammed. See Table 20-5 on page 125 for details.
Read fuse high bits	0101 1000	0000 1000	xxxx xxxx	**oooo oooo**	Read fuse high bits. “0” = pro-grammed, “1” = unprogrammed. See Table 20-4 on page 124 for details.
Read extended fuse bits	0101 0000	0000 1000	xxxx xxxx	**oooo oooo**	Read extended fuse bits. “0” = pro-grammed, “1” = unprogrammed. See Table 20-3 on page 124 for details.
Read calibration byte	0011 1000	000x xxxx	0000 0000	**oooo oooo**	Read calibration byte
Poll RDY/$\overline{\text{BSY}}$	1111 0000	0000 0000	xxxx xxxx	xxxx xxx**o**	If **o** = “1”, a programming operation is still busy. Wait until this bit returns to “0” before applying another command.

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

20.6.2 Serial Programming Characteristics

Figure 20-3. Serial Programming Timing

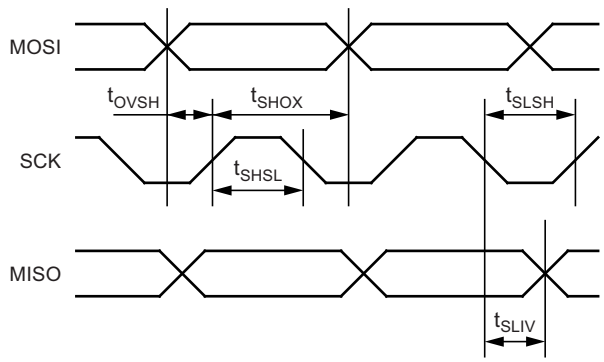


Table 20-11. Serial Programming Characteristics, T_A = –40°C to 125°C, V_{CC} = 2.7 to 5.5V (Unless Otherwise Noted)

Parameter	Symbol	Min	Typ	Max	Units
Oscillator frequency (ATtiny25/45/85V)	$1/t_{CLCL}$	0		4	MHz
Oscillator period (ATtiny25/45/85V)	t_{CLCL}	250			ns
Oscillator frequency (ATtiny25/45/85L, VCC = 2.7 to 5.5V)	$1/t_{CLCL}$	0		10	MHz
Oscillator period (ATtiny25/45/85L, VCC = 2.7 to 5.5V)	t_{CLCL}	100			ns
Oscillator frequency (ATtiny25/45/85, V_{CC} = 4.5V to 5.5V)	$1/t_{CLCL}$	0		20	MHz
Oscillator period (ATtiny25/45/85, V_{CC} = 4.5V to 5.5V)	t_{CLCL}	50			ns
SCK pulse width high	t_{SHSL}	2 t_{CLCL}*			ns
SCK pulse width low	t_{SLSH}	2 t_{CLCL}*			ns
MOSI setup to SCK high	t_{OVSH}	t_{CLCL}			ns
MOSI hold after SCK high	t_{SHOX}	2 t_{CLCL}			ns

Note: 2 t_{CLCL} for f_{ck} < 12MHz, 3 t_{CLCL} for f_{ck} ≥12MHz

20.7 High-voltage Serial Programming

This section describes how to program and verify flash program memory, EEPROM data memory, lock bits and fuse bits in the Atmel® ATtiny25/45/85.

Figure 20-4. High-voltage Serial Programming

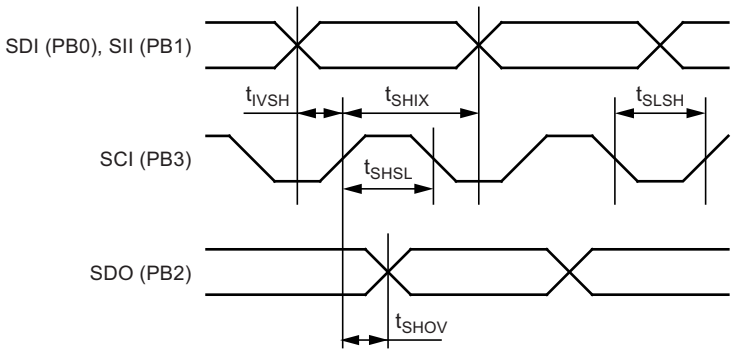
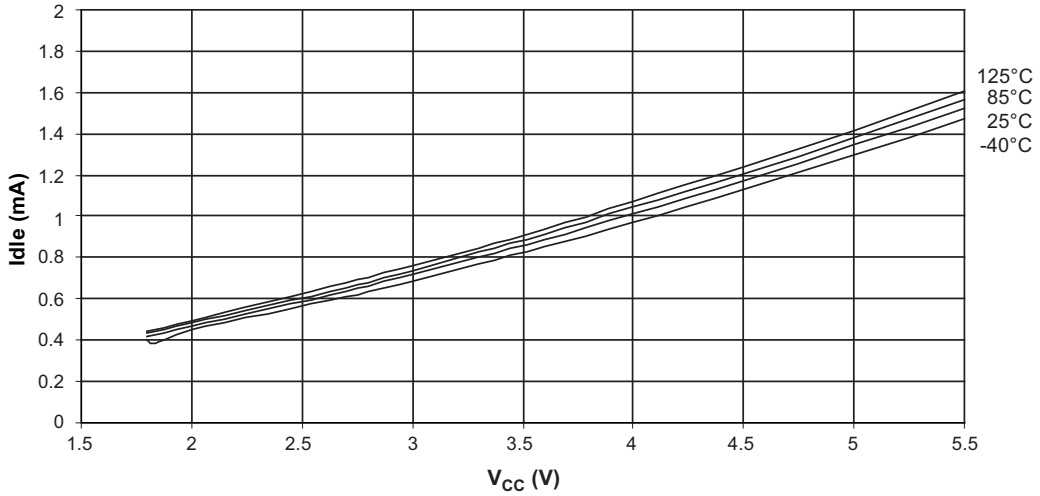


Table 20-12. Pin Name Mapping

Signal Name in High-voltage Serial Programming Mode	Pin Name	I/O	Function
SDI	PB0	I	Serial data input
SII	PB1	I	Serial instruction input
SDO	PB2	O	Serial data output
SCI	PB3	I	Serial clock input (min. 220ns period)

Figure 22-10. Idle Supply Current versus V_{CC} (Internal RC Oscillator, 8MHz)



22.2.1 Using the Power Reduction Register

The tables and formulas below can be used to calculate the additional current consumption for the different I/O modules in active and idle mode. The enabling or disabling of the I/O modules are controlled by the power reduction register. See Section 6.6 “Power Reduction Register” on page 30 for details.

Table 22-1. Additional Current Consumption for the different I/O modules (absolute values)

PRR bit	Typical numbers		
	V_{CC} = 2V, F = 1MHz	V_{CC} = 3V, F = 4MHz	V_{CC} = 5V, F = 8MHz
PRTIM1	43µA	270µA	1090µA
PRTIM0	5.0µA	28µA	116µA
PRUSI	4.0µA	25µA	102µA
PRADC	13µA	84µA	351µA

Table 22-2. Additional Current Consumption (percentage) in Active and Idle mode

PRR bit	Additional Current consumption compared to Active with external clock (see Figure 22-1 and Figure 22-2)	Additional Current consumption compared to Idle with external clock (see Figure 22-6 and Figure 22-7)
PRTIM1	17.3%	68.4%
PRTIM0	1.8%	7.3%
PRUSI	1.6%	6.4%
PRADC	5.4%	21.4%

It is possible to calculate the typical current consumption based on the numbers from Table 22-2 for other V_{CC} and frequency settings than listed in Table 22-1.

22.8 Internal Oscillator Speed

Figure 22-34. Watchdog Oscillator Frequency versus V_{CC})

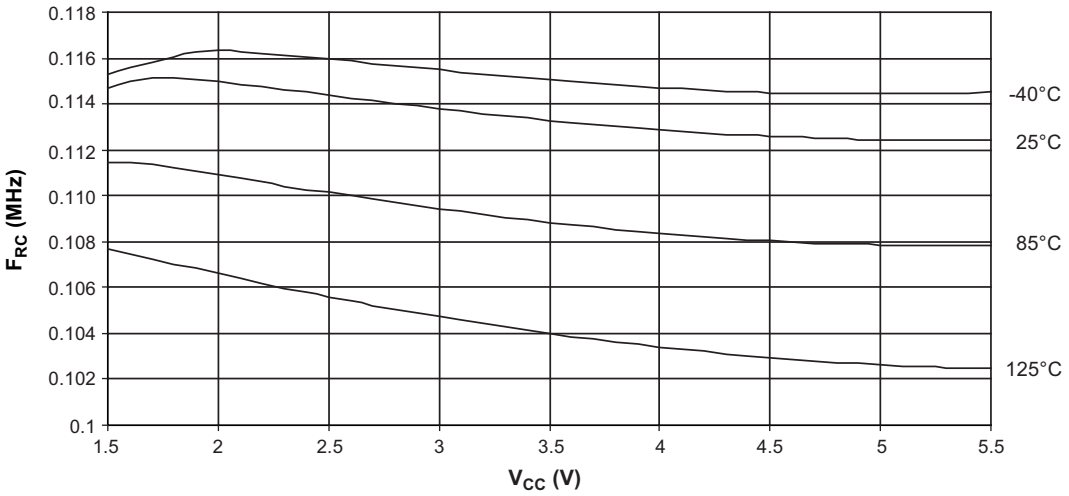


Figure 22-35. Watchdog Oscillator Frequency versus Temperature

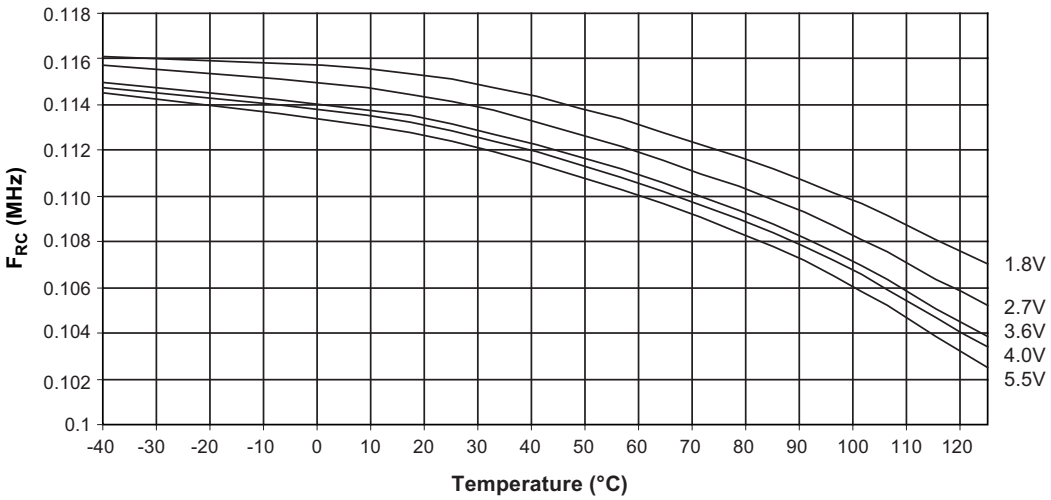


Figure 22-45. Analog to Digital Converter Single Ended Mode OFFSET versus V_{CC}

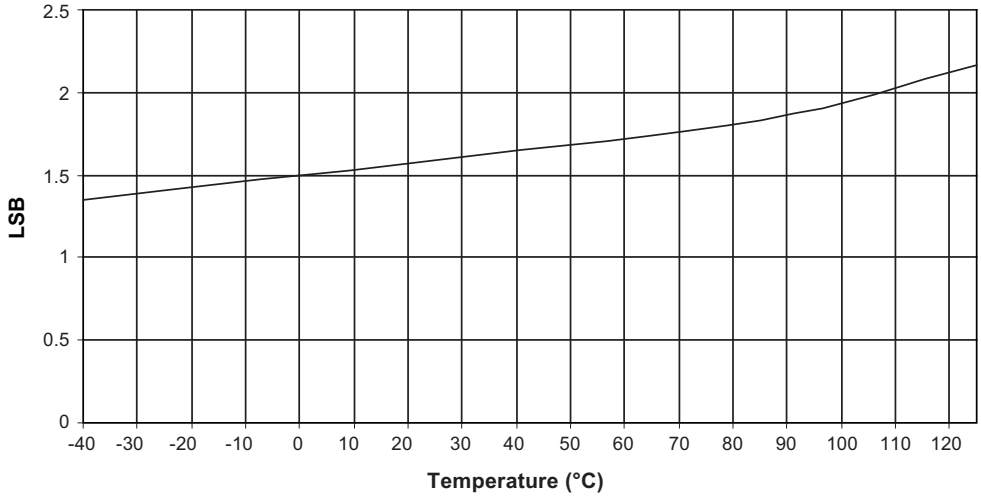


Figure 22-46. Analog to Digital Converter Differential Mode GAIN versus V_{CC}

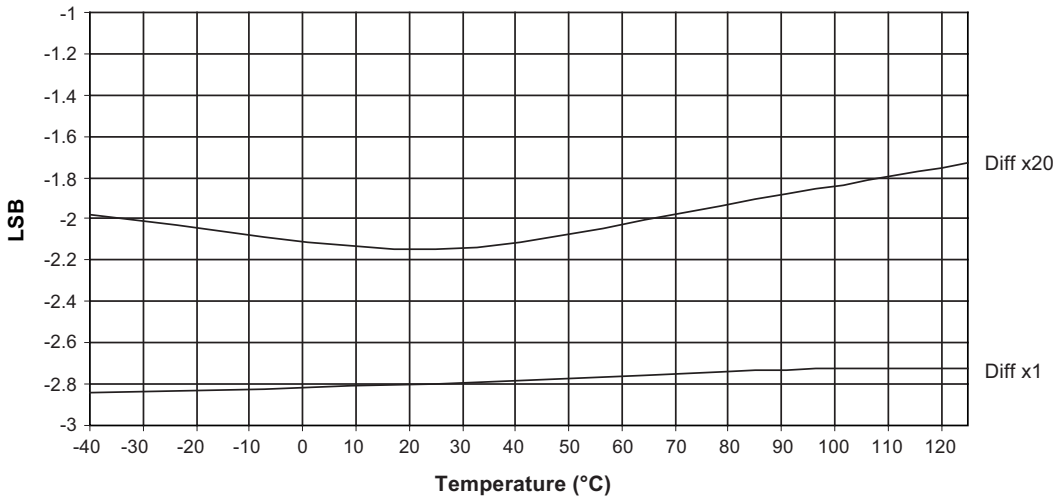


Figure 22-47. Analog to Digital Converter Single Ended Mode GAIN versus V_{CC}

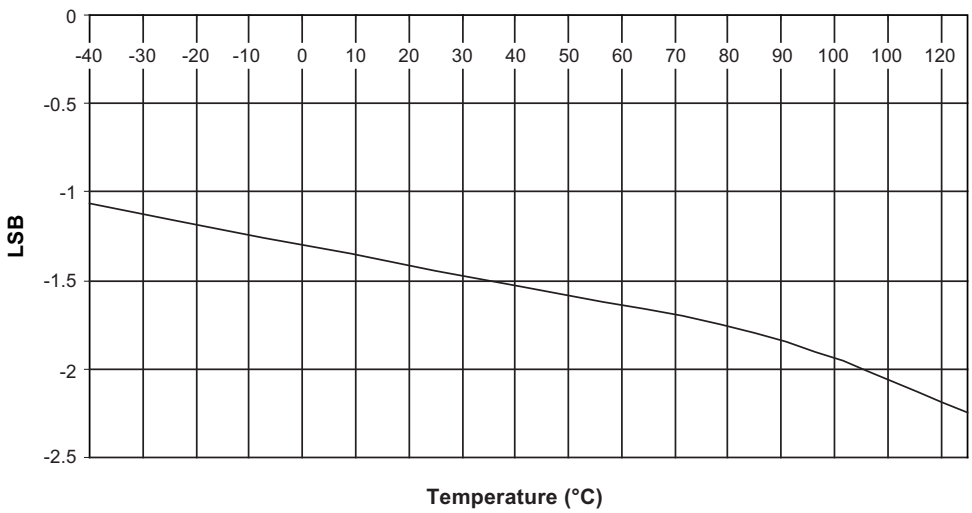


Figure 22-48. Analog to Digital Converter Differential Mode DNL versus V_{CC}

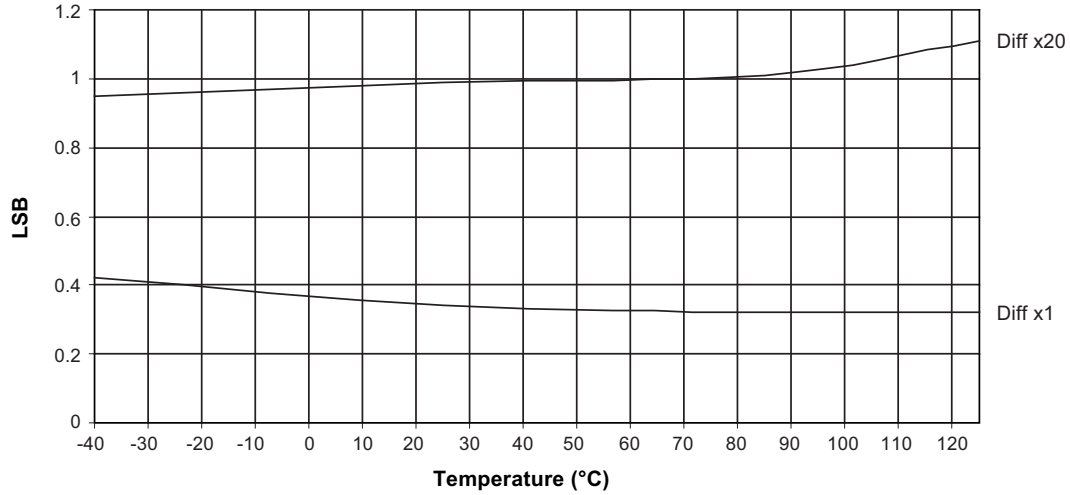


Figure 22-49. Analog to Digital Converter Single Ended Mode DNL versus V_{CC}

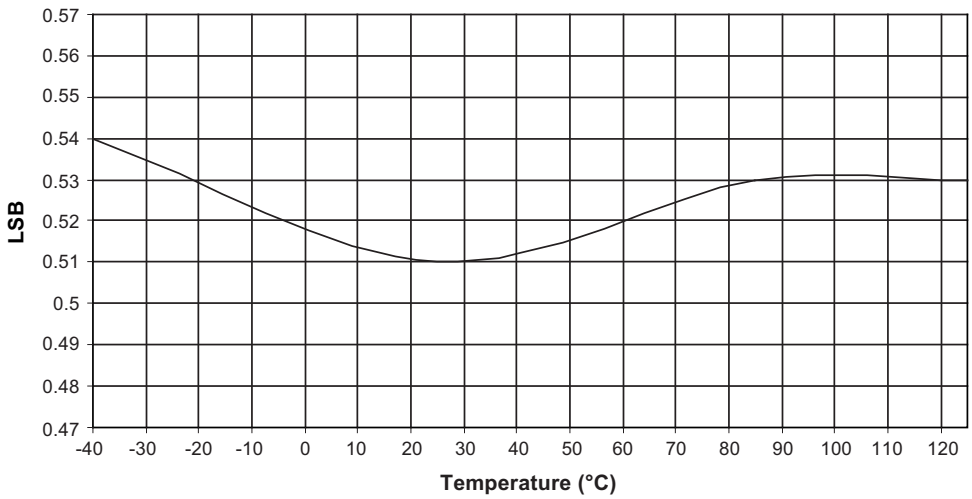


Figure 22-50. Analog to Digital Converter Differential Mode INL versus V_{CC}

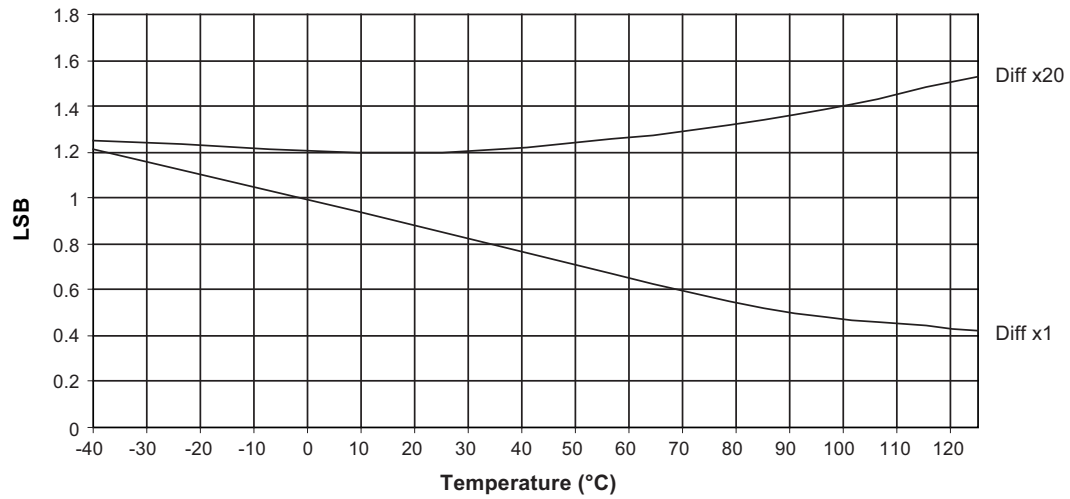


Figure 22-51. Analog to Digital Converter Single Ended Mode INL versus V_{CC}