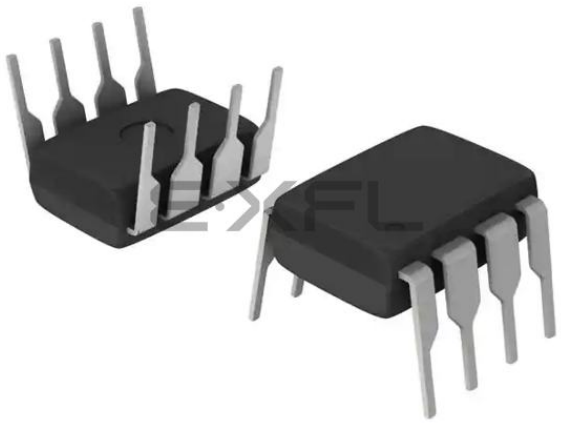


Welcome to E-XFL.COM

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	PIC
Core Size	8-Bit
Speed	4MHz
Connectivity	-
Peripherals	POR, WDT
Number of I/O	5
Program Memory Size	768B (512 x 12)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	25 x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Through Hole
Package / Case	8-DIP (0.300", 7.62mm)
Supplier Device Package	8-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic12c508-04-p

TABLE OF CONTENTS

To Our Valued Customers

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number. e.g., DS30000A is version A of document DS30000.

Errata

An errata sheet may exist for current devices, describing minor operational differences (from the data sheet) and recommended workarounds. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)
- The Microchip Corporate Literature Center; U.S. FAX: (602) 786-7277

When contacting a sales office or the literature center, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Corrections to this Data Sheet

We constantly strive to improve the quality of all our products and documentation. We have spent a great deal of time to ensure that this document is correct. However, we realize that we may have missed a few things. If you find any information that is missing or appears in error, please:

- Fill out and mail in the reader response form in the back of this data sheet.
- E-mail us at webmaster@microchip.com.

We appreciate your assistance in making this a better document.

3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC12C5XX family can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC12C5XX uses a Harvard architecture in which program and data are accessed on separate buses. This improves bandwidth over traditional von Neumann architecture where program and data are fetched on the same bus. Separating program and data memory further allows instructions to be sized differently than the 8-bit wide data word. Instruction opcodes are 12-bits wide making it possible to have all single word instructions. A 12-bit wide program memory access bus fetches a 12-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions. Consequently, all instructions (33) execute in a single cycle (1µs @ 4MHz) except for program branches.

The table below lists program memory (EPROM), data memory (RAM), ROM memory, and non-volatile (EEPROM) for each device.

Device	Memory			
	EPROM Program	ROM Program	RAM Data	EEPROM Data
PIC12C508	512 x 12		25	
PIC12C509	1024 x 12		41	
PIC12C508A	512 x 12		25	
PIC12C509A	1024 x 12		41	
PIC12CR509A		1024 x 12	41	
PIC12CE518	512 x 12		25 x 8	16 x 8
PIC12CE519	1024 x 12		41 x 8	16 x 8

The PIC12C5XX can directly or indirectly address its register files and data memory. All special function registers including the program counter are mapped in the data memory. The PIC12C5XX has a highly orthogonal (symmetrical) instruction set that makes it possible to carry out any operation on any register using any addressing mode. This symmetrical nature and lack of 'special optimal situations' make programming with the PIC12C5XX simple yet efficient. In addition, the learning curve is reduced significantly.

The PIC12C5XX device contains an 8-bit ALU and working register. The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between data in the working register and any register file.

The ALU is 8-bits wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. In two-operand instructions, typically one operand is the W (working) register. The other operand is either a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.

The W register is an 8-bit working register used for ALU operations. It is not an addressable register.

Depending on the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC), and Zero (Z) bits in the STATUS register. The C and DC bits operate as a $\overline{\text{borrow}}$ and $\overline{\text{digit borrow}}$ out bit, respectively, in subtraction. See the SUBWF and ADDWF instructions for examples.

A simplified block diagram is shown in Figure 3-1, with the corresponding device pins described in Table 3-1.

3.1 Clocking Scheme/Instruction Cycle

The clock input (OSC1/CLKIN pin) is internally divided by four to generate four non-overlapping quadrature clocks namely Q1, Q2, Q3 and Q4. Internally, the program counter is incremented every Q1, and the instruction is fetched from program memory and latched into instruction register in Q4. It is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow is shown in Figure 3-2 and Example 3-1.

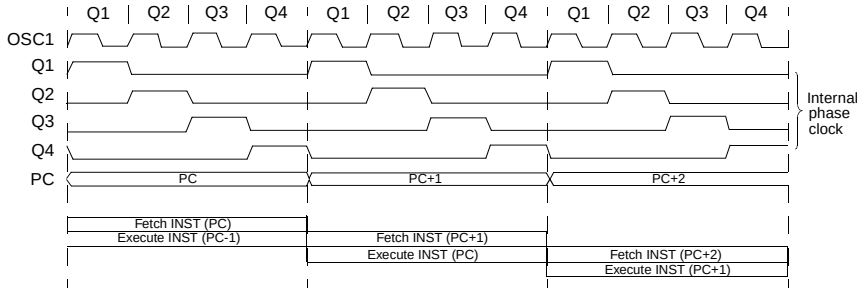
3.2 Instruction Flow/Pipelining

An Instruction Cycle consists of four Q cycles (Q1, Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle while decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g., GOTO) then two cycles are required to complete the instruction (Example 3-1).

A fetch cycle begins with the program counter (PC) incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the Instruction Register (IR) in cycle Q1. This instruction is then decoded and executed during the Q2, Q3, and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

FIGURE 3-2: CLOCK/INSTRUCTION CYCLE



EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW

```
1. MOVLW 03H
2. MOVWF GPIO
3. CALL  SUB_1
4. BSF   GPIO, BIT1
```

Fetch 1	Execute 1					
	Fetch 2	Execute 2				
		Fetch 3	Execute 3			
			Fetch 4	Flush		
				Fetch SUB_1	Execute SUB_1	

All instructions are single cycle, except for any program branches. These take two cycles since the fetch instruction is “flushed” from the pipeline while the new instruction is being fetched and then executed.

4.3 STATUS Register

This register contains the arithmetic status of the ALU, the RESET status, and the page preselect bit for program memories larger than 512 words.

The STATUS register can be the destination for any instruction, as with any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper three bits and set the Z bit. This leaves the STATUS register as `000u u1uu` (where u = unchanged).

It is recommended, therefore, that only `BCF`, `BSF` and `MOVWF` instructions be used to alter the STATUS register because these instructions do not affect the Z, DC or C bits from the STATUS register. For other instructions, which do affect STATUS bits, see Instruction Set Summary.

FIGURE 4-4: STATUS REGISTER (ADDRESS:03h)

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
GPWUF	—	PA0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit7	6	5	4	3	2	1	bit0

R = Readable bit
W = Writable bit
- n = Value at POR reset

bit 7: **GPWUF**: GPIO reset bit
1 = Reset due to wake-up from SLEEP on pin change
0 = After power up or other reset

bit 6: **Unimplemented**

bit 5: **PA0**: Program page preselect bits
1 = Page 1 (200h - 3FFh) - PIC12C509, PIC12C509A, PIC12CR509A and PIC12CE519
0 = Page 0 (000h - 1FFh) - PIC12C5XX
Each page is 512 bytes.
Using the PA0 bit as a general purpose read/write bit in devices which do not use it for program page preselect is not recommended since this may affect upward compatibility with future products.

bit 4: **$\overline{TO}$**: Time-out bit
1 = After power-up, `CLRWDT` instruction, or `SLEEP` instruction
0 = A WDT time-out occurred

bit 3: **$\overline{PD}$**: Power-down bit
1 = After power-up or by the `CLRWDT` instruction
0 = By execution of the `SLEEP` instruction

bit 2: **Z**: Zero bit
1 = The result of an arithmetic or logic operation is zero
0 = The result of an arithmetic or logic operation is not zero

bit 1: **DC**: Digit carry/$\overline{borrow}$ bit (for `ADDWF` and `SUBWF` instructions)
ADDWF
1 = A carry from the 4th low order bit of the result occurred
0 = A carry from the 4th low order bit of the result did not occur
SUBWF
1 = A borrow from the 4th low order bit of the result did not occur
0 = A borrow from the 4th low order bit of the result occurred

bit 0: **C**: Carry/$\overline{borrow}$ bit (for `ADDWF`, `SUBWF` and `RRF`, `RLF` instructions)

ADDWF	**SUBWF**	**RRF or RLF**
1 = A carry occurred	1 = A borrow did not occur	Load bit with LSB or MSB, respectively
0 = A carry did not occur	0 = A borrow occurred	

TABLE 5-1: SUMMARY OF PORT REGISTERS

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on All Other Resets
N/A	TRIS	—	—							--11 1111	--11 1111
N/A	OPTION	$\overline{GPWU}$	$\overline{GPPU}$	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
03H	STATUS	GPWUF	—	PAO	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	q00q quuu(1)
06h	GPIO (PIC12C508/ PIC12C509/ PIC12C508A/ PIC12C509A/ PIC12CR509A)	—	—	GP5	GP4	GP3	GP2	GP1	GP0	--xx xxxx	--uu uuuu
06h	GPIO (PIC12CE518/ PIC12CE519)	SCL	SDA	GP5	GP4	GP3	GP2	GP1	GP0	11xx xxxx	11uu uuuu

Legend: Shaded cells not used by Port Registers, read as '0', — = unimplemented, read as '0', x = unknown, u = unchanged, q = see tables in Section 8.7 for possible values.

Note 1: If reset was due to wake-up on change, then bit 7 = 1. All other resets will cause bit 7 = 0.

5.4 I/O Programming Considerations

5.4.1 BI-DIRECTIONAL I/O PORTS

Some instructions operate internally as read followed by write operations. The BCF and BSF instructions, for example, read the entire port into the CPU, execute the bit operation and re-write the result. Caution must be used when these instructions are applied to a port where one or more pins are used as input/outputs. For example, a BSF operation on bit5 of GPIO will cause all eight bits of GPIO to be read into the CPU, bit5 to be set and the GPIO value to be written to the output latches. If another bit of GPIO is used as a bi-directional I/O pin (say bit0) and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and rewritten to the data latch of this particular pin, overwriting the previous content. As long as the pin stays in the input mode, no problem occurs. However, if bit0 is switched into output mode later on, the content of the data latch may now be unknown.

Example 5-1 shows the effect of two sequential read-modify-write instructions (e.g., BCF, BSF, etc.) on an I/O port.

A pin actively outputting a high or a low should not be driven from external devices at the same time in order to change the level on this pin ("wired-or", "wired-and"). The resulting high output currents may damage the chip.

EXAMPLE 5-1: READ-MODIFY-WRITE INSTRUCTIONS ON AN I/O PORT

```
;Initial GPIO Settings
; GPIO<5:3> Inputs
; GPIO<2:0> Outputs
;
;                    GPIO latch  GPIO pins
;                    ----------  ----------
  BCF   GPIO, 5    ;--01 -ppp   --11 pppp
  BCF   GPIO, 4    ;--10 -ppp   --11 pppp
  MOVLW 007h        ;
  TRIS  GPIO       ;--10 -ppp   --11 pppp
;
;Note that the user may have expected the pin
;values to be --00 pppp. The 2nd BCF caused
;GP5 to be latched as the pin value (High).
```

5.4.2 SUCCESSIVE OPERATIONS ON I/O PORTS

The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (Figure 5-2). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should allow the pin voltage to stabilize (load dependent) before the next instruction, which causes that file to be read into the CPU, is executed. Otherwise, the previous state of that pin may be read into the CPU rather than the new state. When in doubt, it is better to separate these instructions with a NOP or another instruction not accessing this I/O port.

7.0 EEPROM PERIPHERAL OPERATION

This section applies to PIC12CE518 and PIC12CE519 only.

The PIC12CE518 and PIC12CE519 each have 16 bytes of EEPROM data memory. The EEPROM memory has an endurance of 1,000,000 erase/write cycles and a data retention of greater than 40 years. The EEPROM data memory supports a bi-directional 2-wire bus and data transmission protocol. These two-wires are serial data (SDA) and serial clock (SCL), that are mapped to bit6 and bit7, respectively, of the GPIO register (SFR 06h). Unlike the GP0-GP5 that are connected to the I/O pins, SDA and SCL are only connected to the internal EEPROM peripheral. For most applications, all that is required is calls to the following functions:

```
; Byte_Write: Byte write routine
;       Inputs: EEPROM Address     EEADDR
;               EEPROM Data        EEDATA
;       Outputs:    Return 01 in W if OK, else
                    return 00 in W
;
; Read_Current: Read EEPROM at address
currently held by EE device.
;       Inputs: NONE
;       Outputs:    EEPROM Data    EEDATA
;                   Return 01 in W if OK, else
                    return 00 in W
;
; Read_Random: Read EEPROM byte at supplied
address
;       Inputs: EEPROM Address     EEADDR
;       Outputs:    EEPROM Data    EEDATA
;                   Return 01 in W if OK,
                    else return 00 in W
```

The code for these functions is available on our website www.microchip.com. The code will be accessed by either including the source code FL51XINC.ASM or by linking FLASH5IX.ASM.

It is very important to check the return codes when using these calls, and retry the operation if unsuccessful. Unsuccessful return codes occur when the EE data memory is busy with the previous write, which can take up to 4 mS.

7.0.1 SERIAL DATA

SDA is a bi-directional pin used to transfer addresses and data into and data out of the device.

For normal data transfer SDA is allowed to change only during SCL low. Changes during SCL high are reserved for indicating the START and STOP conditions.

The EEPROM interface is a 2-wire bus protocol consisting of data (SDA) and a clock (SCL). Although these lines are mapped into the GPIO register, they are not accessible as external pins; only to the internal EEPROM peripheral. SDA and SCL operation is also slightly different than GPO-GP5 as listed below. Namely, to avoid code overhead in modifying the TRIS register, both SDA and SCL are always outputs. To read data from the EEPROM peripheral requires outputting a ‘1’ on SDA placing it in high-Z state, where only the internal 100K pull-up is active on the SDA line.

SDA:
- Built-in 100K (typical) pull-up to VDD
- Open-drain (pull-down only)
- Always an output
- Outputs a ‘1’ on reset

SCL:
- Full CMOS output
- Always an output
- Outputs a ‘1’ on reset

The following example requires:

- Code Space: 77 words
- RAM Space: 5 bytes (4 are overlayable)
- Stack Levels:1 (The call to the function itself. The functions do not call any lower level functions.)
- Timing:
  - WRITE_BYTE takes 328 cycles
  - READ_CURRENT takes 212 cycles
  - READ_RANDOM takes 416 cycles.
- IO Pins: 0 (No external IO pins are used)

This code must reside in the lower half of a page. The code achieves it's small size without additional calls through the use of a sequencing table. The table is a list of procedures that must be called in order. The table uses an ADDWF PCL,F instruction, effectively a computed goto, to sequence to the next procedure. However the ADDWF PCL,F instruction yields an 8 bit address, forcing the code to reside in the first 256 addresses of a page.

FIGURE 7-3: DATA TRANSFER SEQUENCE ON THE SERIAL BUS

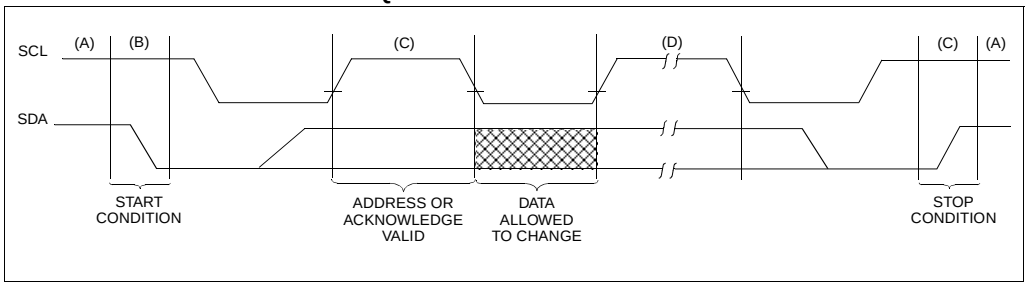
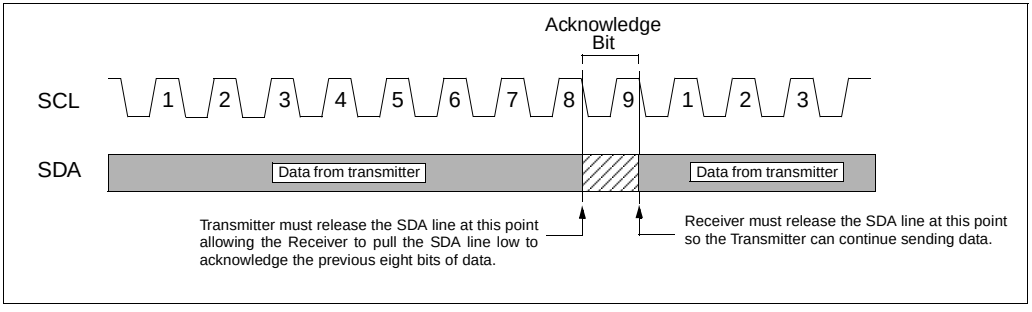


FIGURE 7-4: ACKNOWLEDGE TIMING

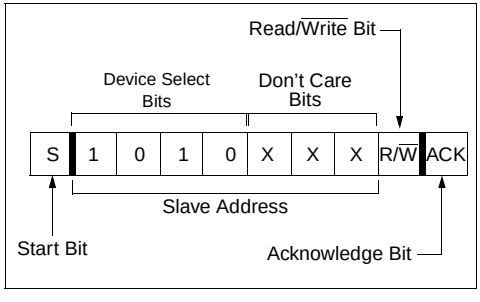


7.2 Device Addressing

After generating a START condition, the bus master transmits a control byte consisting of a slave address and a Read/$\overline{\text{Write}}$ bit that indicates what type of operation is to be performed. The slave address consists of a 4-bit device code (1010) followed by three don't care bits.

The last bit of the control byte determines the operation to be performed. When set to a one a read operation is selected, and when set to a zero a write operation is selected. (Figure 7-5). The bus is monitored for its corresponding slave address all the time. It generates an acknowledge bit if the slave address was true and it is not in a programming mode.

FIGURE 7-5: CONTROL BYTE FORMAT



7.5 READ OPERATIONS

Read operations are initiated in the same way as write operations with the exception that the R/$\overline{W}$ bit of the slave address is set to one. There are three basic types of read operations: current address read, random read, and sequential read.

7.5.1 CURRENT ADDRESS READ

It contains an address counter that maintains the address of the last word accessed, internally incremented by one. Therefore, if the previous read access was to address n, the next current address read operation would access data from address n + 1. Upon receipt of the slave address with the R/$\overline{W}$ bit set to one, the device issues an acknowledge and transmits the eight bit data word. The master will not acknowledge the transfer but does generate a stop condition and the device discontinues transmission (Figure 7-8).

7.5.2 RANDOM READ

Random read operations allow the master to access any memory location in a random manner. To perform this type of read operation, first the word address must be set. This is done by sending the word address to the device as part of a write operation. After the word address is sent, the master generates a start condition following the acknowledge. This terminates the write operation, but not before the internal address pointer is set. Then the master issues the control byte again but with the R/$\overline{W}$ bit set to a one. It will then issue an acknowledge and transmits the eight bit data word. The master will not acknowledge the transfer but does generate a stop condition and the device discontinues transmission (Figure 7-9). After this command, the internal address counter will point to the address location following the one that was just read.

7.5.3 SEQUENTIAL READ

Sequential reads are initiated in the same way as a random read except that after the device transmits the first data byte, the master issues an acknowledge as opposed to a stop condition in a random read. This directs the device to transmit the next sequentially addressed 8-bit word (Figure 7-10).

To provide sequential reads, it contains an internal address pointer which is incremented by one at the completion of each read operation. This address pointer allows the entire memory contents to be serially read during one operation.

FIGURE 7-8: CURRENT ADDRESS READ

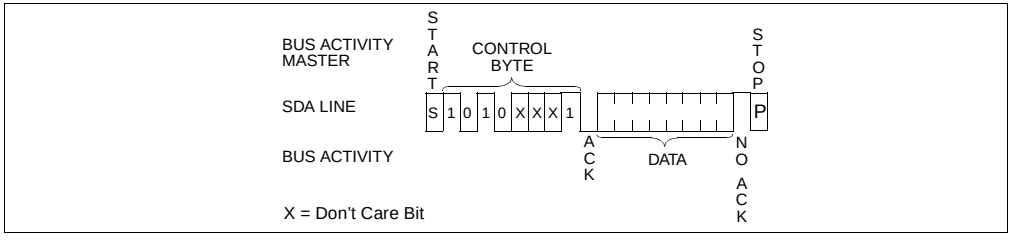


FIGURE 7-9: RANDOM READ

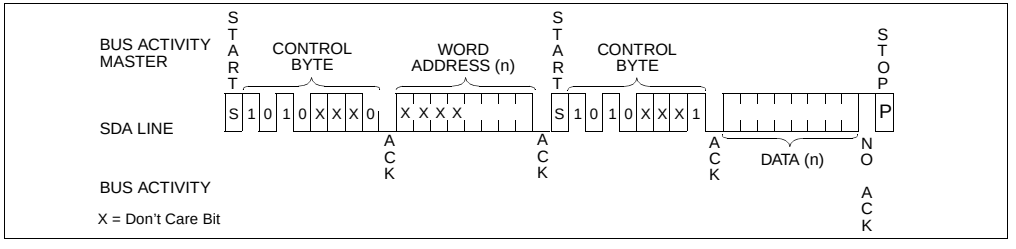
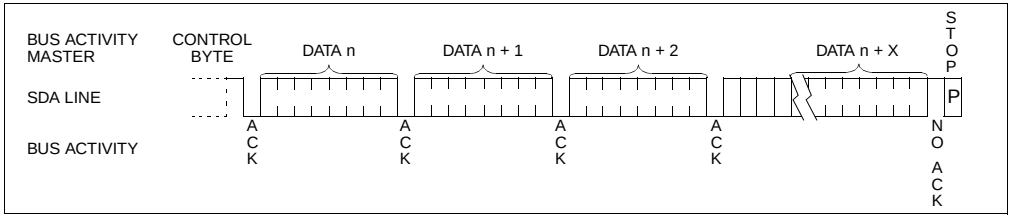


FIGURE 7-10: SEQUENTIAL READ



8.2.5 INTERNAL 4 MHz RC OSCILLATOR

The internal RC oscillator provides a fixed 4 MHz (nominal) system clock at VDD = 5V and 25°C, see "Electrical Specifications" section for information on variation over voltage and temperature.

In addition, a calibration instruction is programmed into the top of memory which contains the calibration value for the internal RC oscillator. This location is never code protected regardless of the code protect settings. This value is programmed as a `MOVLW XX` instruction where XX is the calibration value, and is placed at the reset vector. This will load the W register with the calibration value upon reset and the PC will then roll over to the users program at address 0x000. The user then has the option of writing the value to the OSCCAL Register (05h) or ignoring it.

OSCCAL, when written to with the calibration value, will "trim" the internal oscillator to remove process variation from the oscillator frequency. .

Note:	Please note that erasing the device will also erase the pre-programmed internal calibration value for the internal oscillator. The calibration value must be read prior to erasing the part. so it can be reprogrammed correctly later.

For the PIC12C508A, PIC12C509A, PIC12CE518, PIC12CE519, and PIC12CR509A, bits <7:2>, CAL5-CAL0 are used for calibration. Adjusting CAL5-0 from 000000 to 111111 yields a higher clock speed. Note that bits 1 and 0 of OSCCAL are unimplemented and should be written as 0 when modifying OSCCAL for compatibility with future devices.

For the PIC12C508 and PIC12C509, the upper 4 bits of the register are used. Writing a larger value in this location yields a higher clock speed.

8.3 RESET

The device differentiates between various kinds of reset:

a) Power on reset (POR)

b) $\overline{\text{MCLR}}$ reset during normal operation

c) $\overline{\text{MCLR}}$ reset during SLEEP

d) WDT time-out reset during normal operation

e) WDT time-out reset during SLEEP

f) Wake-up from SLEEP on pin change

Some registers are not reset in any way; they are unknown on POR and unchanged in any other reset. Most other registers are reset to "reset state" on power-on reset (POR), $\overline{\text{MCLR}}$, WDT or wake-up on pin change reset during normal operation. They are not affected by a WDT reset during SLEEP or $\overline{\text{MCLR}}$ reset during SLEEP, since these resets are viewed as resumption of normal operation. The exceptions to this are $\overline{\text{TO}}$, $\overline{\text{PD}}$, and GPWUF bits. They are set or cleared differently in different reset situations. These bits are used in software to determine the nature of reset. See Table 8-3 for a full description of reset states of all registers.

SLEEP **Enter SLEEP Mode**

Syntax: [*label*] SLEEP

Operands: None

Operation: 00h → WDT;
0 → WDT prescaler;
1 → $\overline{TO}$;
0 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$, GPWUF

Encoding:

0000	0000	0011

Description: Time-out status bit ($\overline{TO}$) is set. The power down status bit ($\overline{PD}$) is cleared.
GPWUF is unaffected.
The WDT and its prescaler are cleared.
The processor is put into SLEEP mode with the oscillator stopped. See section on SLEEP for more details.

Words: 1

Cycles: 1

Example: SLEEP

SUBWF **Subtract W from f**

Syntax: [*label*] SUBWF f,d

Operands: $0 \le f \le 31$
$d \in [0,1]$

Operation: (f) – (W) → (dest)

Status Affected: C, DC, Z

Encoding:

0000	10df	ffff

Description: Subtract (2's complement method) the W register from register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example 1:

```
SUBWF   REG1, 1
```

Before Instruction
```
REG1    =   3
W       =   2
C       =   ?
```
After Instruction
```
REG1    =   1
W       =   2
C       =   1     ; result is positive
```

Example 2:

Before Instruction
```
REG1    =   2
W       =   2
C       =   ?
```
After Instruction
```
REG1    =   0
W       =   2
C       =   1     ; result is zero
```

Example 3:

Before Instruction
```
REG1    =   1
W       =   2
C       =   ?
```
After Instruction
```
REG1    =   FF
W       =   2
C       =   0     ; result is negative
```

10.6 SIMICE Entry-Level Hardware Simulator

SIMICE is an entry-level hardware development system designed to operate in a PC-based environment with Microchip's simulator MPLAB™-SIM. Both SIMICE and MPLAB-SIM run under Microchip Technology's MPLAB Integrated Development Environment (IDE) software. Specifically, SIMICE provides hardware simulation for Microchip's PIC12C5XX, PIC12CE5XX, and PIC16C5X families of PICmicro® 8-bit microcontrollers. SIMICE works in conjunction with MPLAB-SIM to provide non-real-time I/O port emulation. SIMICE enables a developer to run simulator code for driving the target system. In addition, the target system can provide input to the simulator code. This capability allows for simple and interactive debugging without having to manually generate MPLAB-SIM stimulus files. SIMICE is a valuable debugging tool for entry-level system development.

10.7 PICDEM-1 Low-Cost PICmicro® Demonstration Board

The PICDEM-1 is a simple board which demonstrates the capabilities of several of Microchip's microcontrollers. The microcontrollers supported are: PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C8X, PIC17C42, PIC17C43 and PIC17C44. All necessary hardware and software is included to run basic demo programs. The users can program the sample microcontrollers provided with the PICDEM-1 board, on a PRO MATE II or PICSTART-Plus programmer, and easily test firmware. The user can also connect the PICDEM-1 board to the MPLAB-ICE emulator and download the firmware to the emulator for testing. Additional prototype area is available for the user to build some additional hardware and connect it to the microcontroller socket(s). Some of the features include an RS-232 interface, a potentiometer for simulated analog input, push-button switches and eight LEDs connected to PORTB.

10.8 PICDEM-2 Low-Cost PIC16CXX Demonstration Board

The PICDEM-2 is a simple demonstration board that supports the PIC16C62, PIC16C64, PIC16C65, PIC16C73 and PIC16C74 microcontrollers. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM-2 board, on a PRO MATE II programmer or PICSTART-Plus, and easily test firmware. The MPLAB-ICE emulator may also be used with the PICDEM-2 board to test firmware. Additional prototype area has been provided to the user for adding additional hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push-button switches, a potentiometer for simulated analog input, a Serial EEPROM to demonstrate usage of the I^2C bus and separate headers for connection to an LCD module and a keypad.

10.9 PICDEM-3 Low-Cost PIC16CXXX Demonstration Board

The PICDEM-3 is a simple demonstration board that supports the PIC16C923 and PIC16C924 in the PLCC package. It will also support future 44-pin PLCC microcontrollers with a LCD Module. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM-3 board, on a PRO MATE II programmer or PICSTART Plus with an adapter socket, and easily test firmware. The MPLAB-ICE emulator may also be used with the PICDEM-3 board to test firmware. Additional prototype area has been provided to the user for adding hardware and connecting it to the microcontroller socket(s). Some of the features include an RS-232 interface, push-button switches, a potentiometer for simulated analog input, a thermistor and separate headers for connection to an external LCD module and a keypad. Also provided on the PICDEM-3 board is an LCD panel, with 4 commons and 12 segments, that is capable of displaying time, temperature and day of the week. The PICDEM-3 provides an additional RS-232 interface and Windows 3.1 software for showing the demultiplexed LCD signals on a PC. A simple serial interface allows the user to construct a hardware demultiplexer for the LCD signals.

TABLE 10-1: DEVELOPMENT TOOLS FROM MICROCHIP

		PIC12C5XX	PIC14000	PIC16C5X	PIC16CXXX	PIC16C6X	PIC16C7XX	PIC16C8X	PIC16C9XX	PIC17C4X	PIC17C7XX	24CXX 25CXX 93CXX	HCS200 HCS300 HCS301
Emulator Products	MPLAB™-ICE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
	ICEPIC™ Low-Cost In-Circuit Emulator			✓	✓	✓	✓	✓	✓				
Software Tools	MPLAB™ Integrated Development Environment	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
	MPLAB™ C17* Compiler									✓	✓		
	*fuzzy*TECH®-MP Explorer/Edition Fuzzy Logic Dev. Tool	✓	✓	✓	✓	✓	✓	✓	✓	✓			
	Total Endurance™ Software Model											✓	
Programmers	PICSTART® Plus Low-Cost Universal Dev. Kit	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
	PRO MATE® II Universal Programmer	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	KEELOQ® Programmer												✓
Demo Boards	SEEVAL® Designers Kit											✓	
	SIMICE	✓		✓									
	PICDEM-14A		✓										
	PICDEM-1			✓	✓			✓		✓			
	PICDEM-2					✓	✓						
	PICDEM-3								✓				
	KEELOQ® Evaluation Kit												✓
	KEELOQ Transponder Kit												✓

FIGURE 12-5: IOH vs. VOH, VDD = 2.5 V

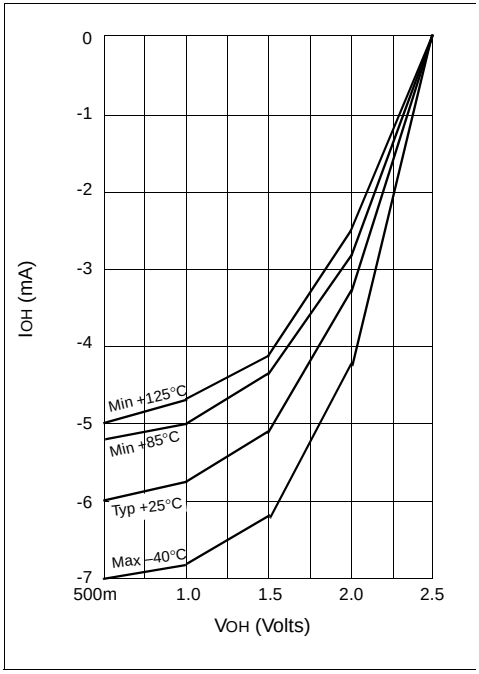


FIGURE 12-6: IOH vs. VOH, VDD = 5.5 V

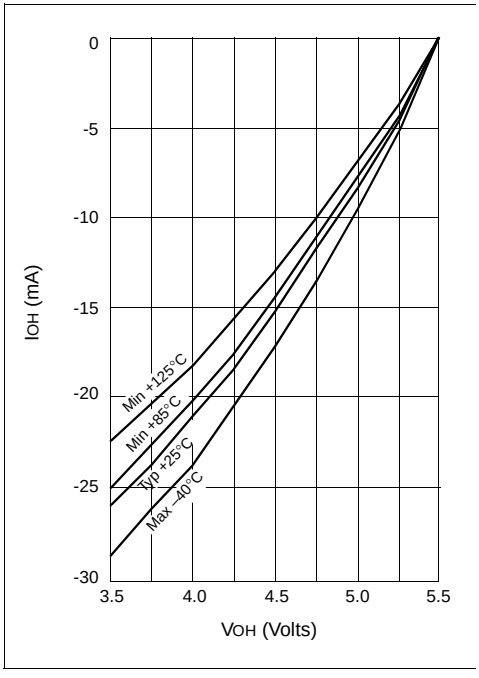


FIGURE 12-7: IOL vs. VOL, VDD = 2.5 V

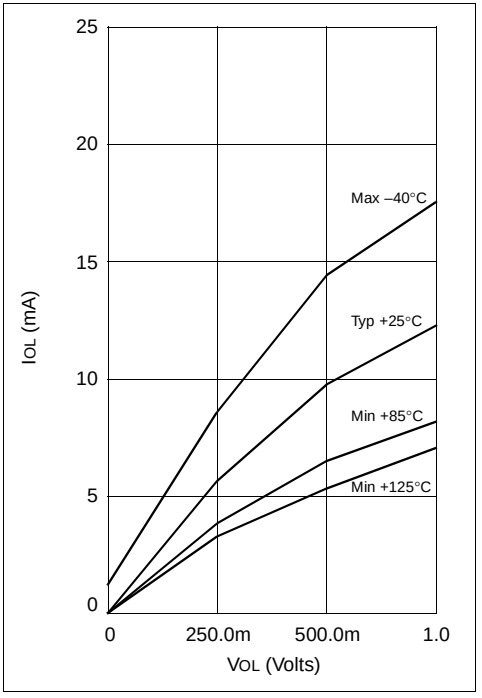
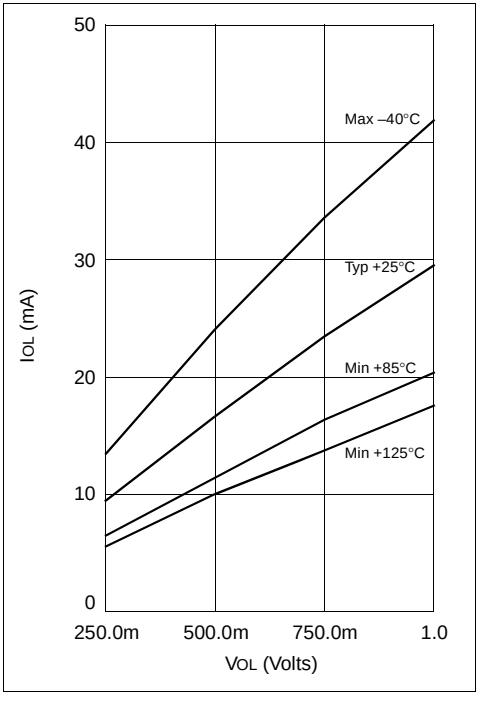


FIGURE 12-8: IOL vs. VOL, VDD = 5.5 V



13.6 Timing Diagrams and Specifications

FIGURE 13-2: EXTERNAL CLOCK TIMING - PIC12C508A, PIC12C509A, PIC12CR509A, PIC12CE518, PIC12CE519, PIC12LC508A, PIC12LC509A, PIC12LCR509A, PIC12LCE518 and PIC12LCE519

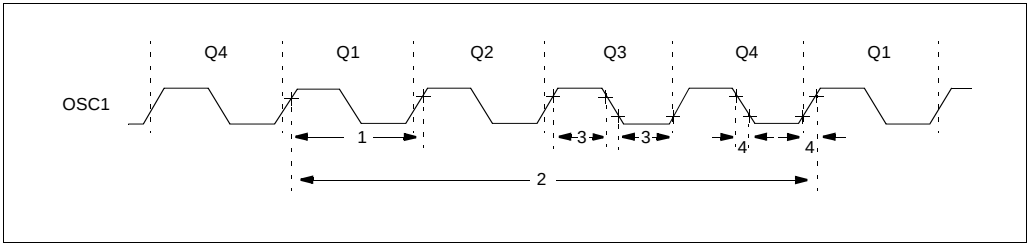


TABLE 13-2: EXTERNAL CLOCK TIMING REQUIREMENTS - PIC12C508A, PIC12C509A, PIC12CE518, PIC12CE519, PIC12LC508A, PIC12LC509A, PIC12LCR509A, PIC12LCE518 and PIC12LCE519

AC Characteristics	Standard Operating Conditions (unless otherwise specified) Operating Temperature 0°C ≤ TA ≤ +70°C (commercial), –40°C ≤ TA ≤ +85°C (industrial), –40°C ≤ TA ≤ +125°C (extended) Operating Voltage VDD range is described in Section 13.1

Parameter No.	Sym	Characteristic	Min	Typ(1)	Max	Units	Conditions
	FOSC	External CLKIN Frequency(2)					
			DC	—	4	MHz	XT osc mode
			DC	—	200	kHz	LP osc mode
		Oscillator Frequency(2)	DC	—	4	MHz	EXTRC osc mode
			0.1	—	4	MHz	XT osc mode
			DC	—	200	kHz	LP osc mode
1	TOSC	External CLKIN Period(2)					
			250	—	—	ns	XT osc mode
			5	—	—	ms	LP osc mode
		Oscillator Period(2)	250	—	—	ns	EXTRC osc mode
			250	—	10,000	ns	XT osc mode
			5	—	—	ms	LP osc mode
2	Tcy	Instruction Cycle Time(3)	—	4/FOSC	—	—	
3	TosL, TosH	Clock in (OSC1) Low or High Time	50*	—	—	ns	XT oscillator
			2*	—	—	ms	LP oscillator
4	TosR, TosF	Clock in (OSC1) Rise or Fall Time	—	—	25*	ns	XT oscillator
			—	—	50*	ns	LP oscillator

* These parameters are characterized but not tested.

Note 1: Data in the Typical ("Typ") column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

2: All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption.
When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

3: Instruction cycle period (TCY) equals four times the input oscillator time base period.

TABLE 13-8: EEPROM MEMORY BUS TIMING REQUIREMENTS - PIC12CE5XX ONLY.

AC Characteristics	Standard Operating Conditions (unless otherwise specified) Operating Temperature 0°C ≤ TA ≤ +70°C, Vcc = 3.0V to 5.5V (commercial) –40°C ≤ TA ≤ +85°C, Vcc = 3.0V to 5.5V (industrial) –40°C ≤ TA ≤ +125°C, Vcc = 4.5V to 5.5V (extended) Operating Voltage VDD range is described in Section 13.1				
Parameter	**Symbol**	**Min**	**Max**	**Units**	**Conditions**
Clock frequency	FCLK	— — —	100 100 400	kHz	4.5V ≤ Vcc ≤ 5.5V (E Temp range) 3.0V ≤ Vcc ≤ 4.5V 4.5V ≤ Vcc ≤ 5.5V
Clock high time	THIGH	4000 4000 600	— — —	ns	4.5V ≤ Vcc ≤ 5.5V (E Temp range) 3.0V ≤ Vcc ≤ 4.5V 4.5V ≤ Vcc ≤ 5.5V
Clock low time	TLOW	4700 4700 1300	— — —	ns	4.5V ≤ Vcc ≤ 5.5V (E Temp range) 3.0V ≤ Vcc ≤ 4.5V 4.5V ≤ Vcc ≤ 5.5V
SDA and SCL rise time (Note 1)	TR	— — —	1000 1000 300	ns	4.5V ≤ Vcc ≤ 5.5V (E Temp range) 3.0V ≤ Vcc ≤ 4.5V 4.5V ≤ Vcc ≤ 5.5V
SDA and SCL fall time	TF	—	300	ns	(Note 1)
START condition hold time	THD:STA	4000 4000 600	— — —	ns	4.5V ≤ Vcc ≤ 5.5V (E Temp range) 3.0V ≤ Vcc ≤ 4.5V 4.5V ≤ Vcc ≤ 5.5V
START condition setup time	TSU:STA	4700 4700 600	— — —	ns	4.5V ≤ Vcc ≤ 5.5V (E Temp range) 3.0V ≤ Vcc ≤ 4.5V 4.5V ≤ Vcc ≤ 5.5V
Data input hold time	THD:DAT	0	—	ns	(Note 2)
Data input setup time	TSU:DAT	250 250 100	— — —	ns	4.5V ≤ Vcc ≤ 5.5V (E Temp range) 3.0V ≤ Vcc ≤ 4.5V 4.5V ≤ Vcc ≤ 5.5V
STOP condition setup time	TSU:STO	4000 4000 600	— — —	ns	4.5V ≤ Vcc ≤ 5.5V (E Temp range) 3.0V ≤ Vcc ≤ 4.5V 4.5V ≤ Vcc ≤ 5.5V
Output valid from clock (Note 2)	TAA	— — —	3500 3500 900	ns	4.5V ≤ Vcc ≤ 5.5V (E Temp range) 3.0V ≤ Vcc ≤ 4.5V 4.5V ≤ Vcc ≤ 5.5V
Bus free time: Time the bus must be free before a new transmission can start	TBUF	4700 4700 1300	— — —	ns	4.5V ≤ Vcc ≤ 5.5V (E Temp range) 3.0V ≤ Vcc ≤ 4.5V 4.5V ≤ Vcc ≤ 5.5V
Output fall time from VIH minimum to VIL maximum	TOF	20+0.1 CB	250	ns	(Note 1), CB ≤ 100 pF
Input filter spike suppression (SDA and SCL pins)	TSP	—	50	ns	(Notes 1, 3)
Write cycle time	TWC	—	4	ms	
Endurance		1M	—	cycles	25°C, VCC = 5.0V, Block Mode (Note 4)

Note 1: Not 100% tested. CB = total capacitance of one bus line in pF.

2: As a transmitter, the device must provide an internal minimum delay time to bridge the undefined region (minimum 300 ns) of the falling edge of SCL to avoid unintended generation of START or STOP conditions.

3: The combined TSP and VHYS specifications are due to new Schmitt trigger inputs which provide improved noise spike suppression. This eliminates the need for a TI specification for standard operation.

4: This parameter is not tested but guaranteed by characterization. For endurance estimates in a specific application, please consult the Total Endurance Model which can be obtained on Microchip's website.

14.0 DC AND AC CHARACTERISTICS - PIC12C508A/PIC12C509A/ PIC12LC508A/PIC12LC509A, PIC12CE518/PIC12CE519/PIC12CR509A/ PIC12LCE518/PIC12LCE519/ PIC12LCR509A

The graphs and tables provided in this section are for design guidance and are not tested. In some graphs or tables the data presented are outside specified operating range (e.g., outside specified VDD range). This is for information only and devices will operate properly only within the specified range.

The data presented in this section is a statistical summary of data collected on units from different lots over a period of time. “Typical” represents the mean of the distribution while “max” or “min” represents (mean + 3σ) and (mean – 3σ) respectively, where σ is standard deviation.

FIGURE 14-1: CALIBRATED INTERNAL RC FREQUENCY RANGE VS. TEMPERATURE (VDD = 5.0V) (INTERNAL RC IS CALIBRATED TO 25°C, 5.0V)

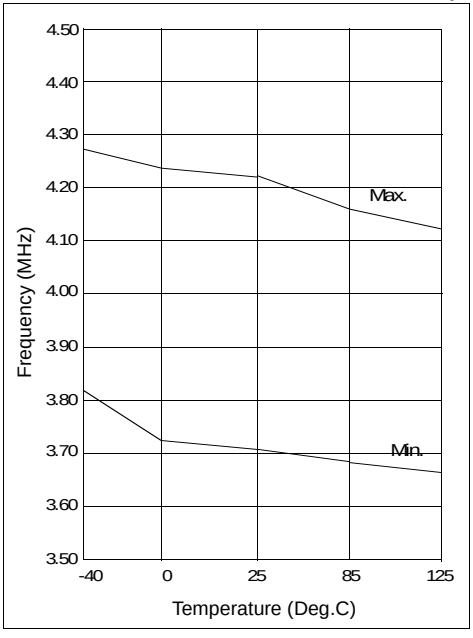
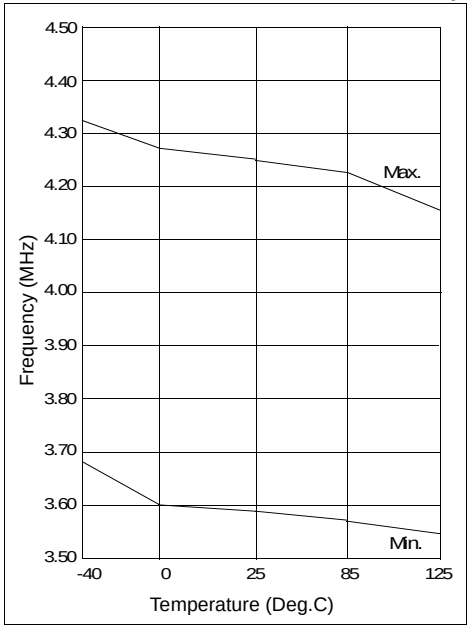
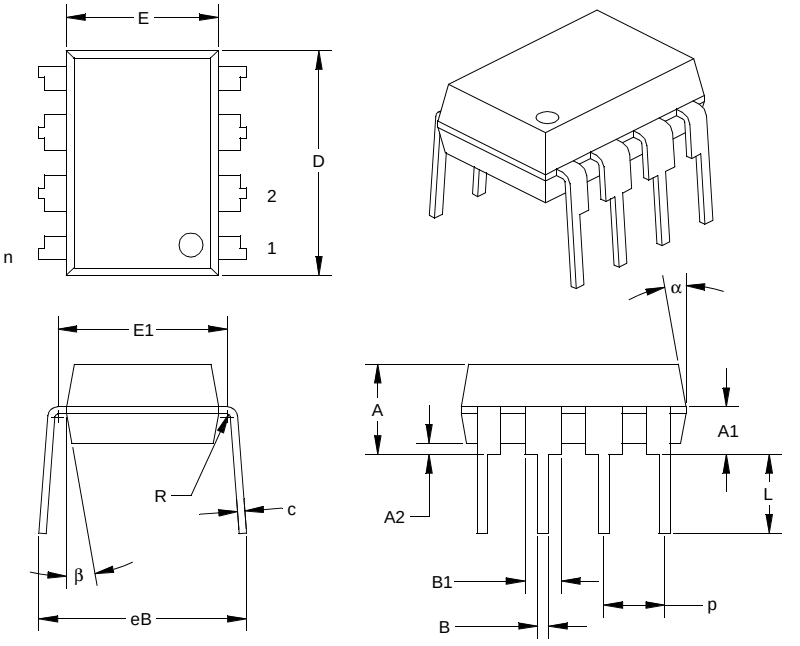


FIGURE 14-2: CALIBRATED INTERNAL RC FREQUENCY RANGE VS. TEMPERATURE (VDD = 2.5V) (INTERNAL RC IS CALIBRATED TO 25°C, 5.0V)



Package Type: K04-018 8-Lead Plastic Dual In-line (P) – 300 mil



Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
PCB Row Spacing			0.300			7.62	
Number of Pins	n		8			8	
Pitch	p		0.100			2.54	
Lower Lead Width	B	0.014	0.018	0.022	0.36	0.46	0.56
Upper Lead Width	B1†	0.055	0.060	0.065	1.40	1.52	1.65
Shoulder Radius	R	0.000	0.005	0.010	0.00	0.13	0.25
Lead Thickness	c	0.006	0.012	0.015	0.20	0.29	0.38
Top to Seating Plane	A	0.140	0.150	0.160	3.56	3.81	4.06
Top of Lead to Seating Plane	A1	0.060	0.080	0.100	1.52	2.03	2.54
Base to Seating Plane	A2	0.005	0.020	0.035	0.13	0.51	0.89
Tip to Seating Plane	L	0.120	0.130	0.140	3.05	3.30	3.56
Package Length	D‡	0.355	0.370	0.385	9.02	9.40	9.78
Molded Package Width	E‡	0.245	0.250	0.260	6.22	6.35	6.60
Radius to Radius Width	E1	0.267	0.280	0.292	6.78	7.10	7.42
Overall Row Spacing	eB	0.310	0.342	0.380	7.87	8.67	9.65
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

* Controlling Parameter.

† Dimension “B1” does not include dam-bar protrusions. Dam-bar protrusions shall not exceed 0.003” (0.076 mm) per side or 0.006” (0.152 mm) more than dimension “B1.”

‡ Dimensions “D” and “E” do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.010” (0.254 mm) per side or 0.020” (0.508 mm) more than dimensions “D” or “E.”

INDEX

ON-LINE SUPPORT

Microchip provides on-line support on the Microchip World Wide Web (WWW) site.

The web site is used by Microchip as a means to make files and information easily available to customers. To view the site, the user must have access to the Internet and a web browser, such as Netscape or Microsoft Explorer. Files are also available for FTP download from our FTP site.

Connecting to the Microchip Internet Web Site

The Microchip web site is available by using your favorite Internet browser to attach to:

www.microchip.com

The file transfer site is available by using an FTP service to connect to:

ftp://ftp.microchip.com

The web site and file transfer site provide a variety of services. Users may download files for the latest Development Tools, Data Sheets, Application Notes, User's Guides, Articles and Sample Programs. A variety of Microchip specific business information is also available, including listings of Microchip sales offices, distributors and factory representatives. Other data available for consideration is:

- Latest Microchip Press Releases
- Technical Support Section with Frequently Asked Questions
- Design Tips
- Device Errata
- Job Postings
- Microchip Consultant Program Member Listing
- Links to other useful web sites related to Microchip Products
- Conferences for products, Development Systems, technical information and more
- Listing of seminars and events

Systems Information and Upgrade Hot Line

The Systems Information and Upgrade Line provides system users a listing of the latest versions of all of Microchip's development systems software products. Plus, this line provides information on how customers can receive any currently available upgrade kits.The Hot Line Numbers are:

1-800-755-2345 for U.S. and most of Canada, and

1-602-786-7302 for the rest of the world.

981103