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	20MHz
Connectivity	-
Peripherals	POR, WDT
Number of I/O	11
Program Memory Size	1.5KB (1K x 12)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	72 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	14-SOIC (0.154", 3.90mm Width)
Supplier Device Package	14-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16c505-20-sl

3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC16C505 can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16C505 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 (200ns @ 20MHz) except for program branches.

The Table below lists program memory (EPROM) and data memory (RAM) for the PIC16C505.

Device	Memory	
	Program	Data
PIC16C505	1024 x 12	72 x 8

The PIC16C505 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 PIC16C505 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 PIC16C505 simple yet efficient. In addition, the learning curve is reduced significantly.

The PIC16C505 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, one operand is typically 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 borrow and 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.

4.0 MEMORY ORGANIZATION

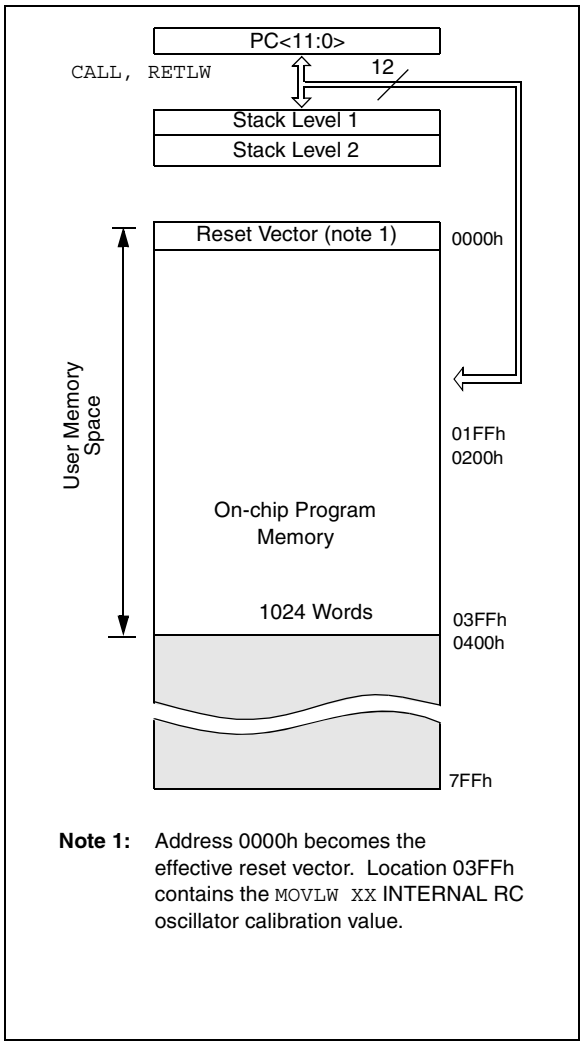
PIC16C505 memory is organized into program memory and data memory. For the PIC16C505, a paging scheme is used. Program memory pages are accessed using one STATUS register bit. Data memory banks are accessed using the File Select Register (FSR).

4.1 Program Memory Organization

The PIC16C505 devices have a 12-bit Program Counter (PC).

The 1K x 12 (0000h-03FFh) for the PIC16C505 are physically implemented. Refer to Figure 4-1. Accessing a location above this boundary will cause a wrap-around within the first 1K x 12 space. The effective reset vector is at 0000h, (see Figure 4-1). Location 03FFh contains the internal clock oscillator calibration value. This value should never be overwritten.

FIGURE 4-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC16C505



Note 1: Address 0000h becomes the effective reset vector. Location 03FFh contains the `MOVLW XX` INTERNAL RC oscillator calibration value.

4.2 Data Memory Organization

Data memory is composed of registers or bytes of RAM. Therefore, data memory for a device is specified by its register file. The register file is divided into two functional groups: Special Function Registers and General Purpose Registers.

The Special Function Registers include the TMR0 register, the Program Counter (PCL), the Status Register, the I/O registers (ports) and the File Select Register (FSR). In addition, Special Function Registers are used to control the I/O port configuration and prescaler options.

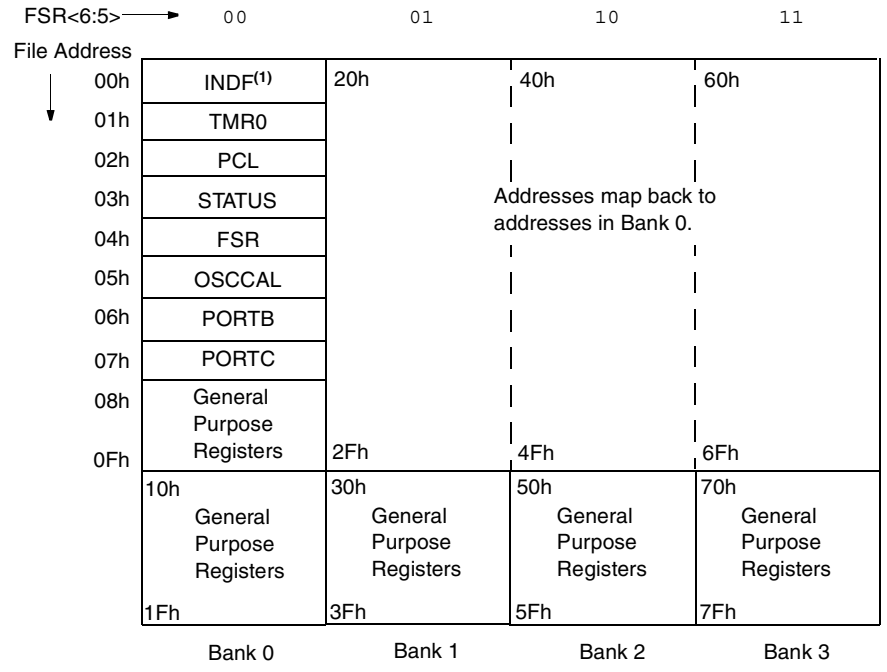
The General Purpose Registers are used for data and control information under command of the instructions.

For the PIC16C505, the register file is composed of 8 Special Function Registers, 24 General Purpose Registers and 48 General Purpose Registers that may be addressed using a banking scheme (Figure 4-2).

4.2.1 GENERAL PURPOSE REGISTER FILE

The General Purpose Register file is accessed, either directly or indirectly, through the File Select Register FSR (Section 4.8).

FIGURE 4-2: PIC16C505 REGISTER FILE MAP



Note 1: Not a physical register.

6.2 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module or as a postscaler for the Watchdog Timer (WDT), respectively (Section 7.6). For simplicity, this counter is being referred to as “prescaler” throughout this data sheet. Note that the prescaler may be used by either the Timer0 module or the WDT, but not both. Thus, a prescaler assignment for the Timer0 module means that there is no prescaler for the WDT, and vice-versa.

The PSA and PS<2:0> bits (OPTION<3:0>) determine prescaler assignment and prescale ratio.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., `CLRF 1`, `MOVWF 1`, `BSF 1,x`, etc.) will clear the prescaler. When assigned to WDT, a `CLRWDT` instruction will clear the prescaler along with the WDT. The prescaler is neither readable nor writable. On a RESET, the prescaler contains all '0's.

6.2.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control (i.e., it can be changed “on-the-fly” during program execution). To avoid an unintended device RESET, the following instruction sequence (Example 6-1) must be executed when changing the prescaler assignment from Timer0 to the WDT.

EXAMPLE 6-1: CHANGING PRESCALER (TIMER0→WDT)

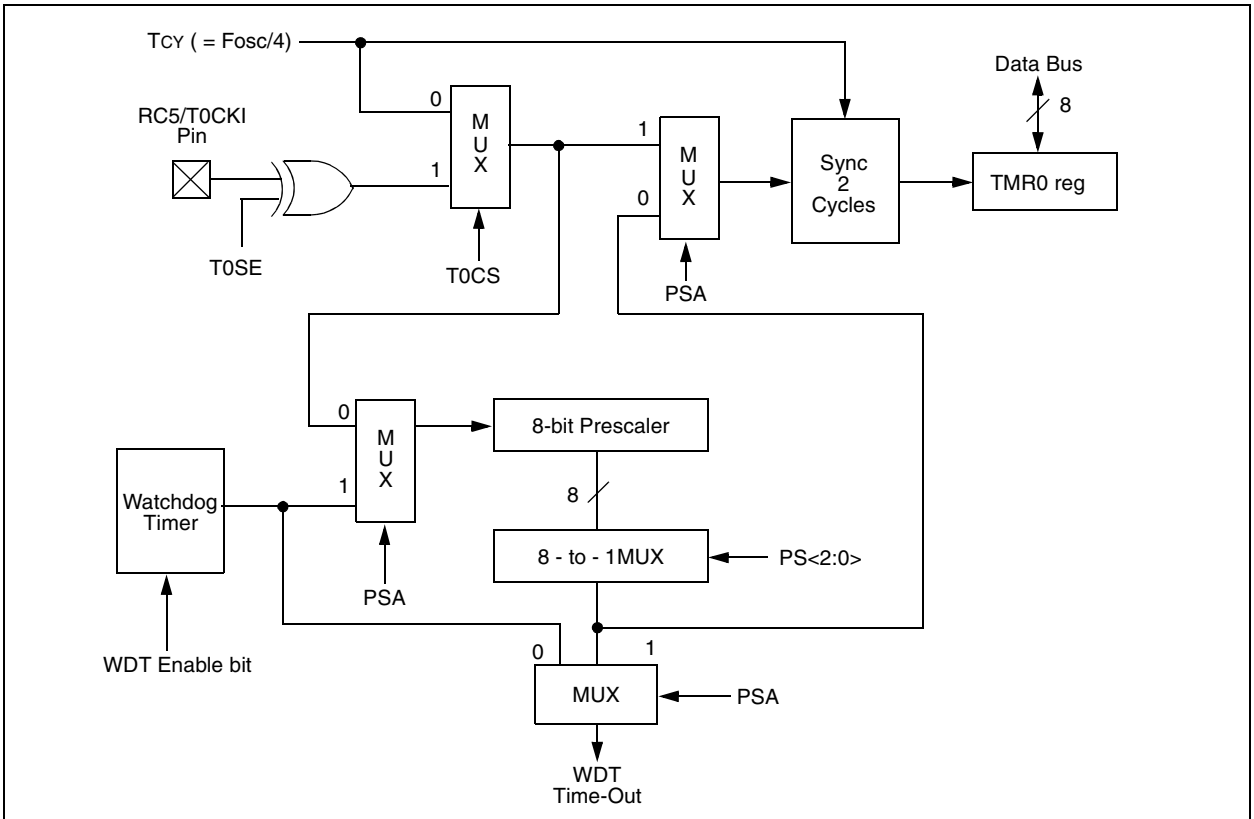
```
1.CLRWDT              ;Clear WDT
2.CLRF   TMR0         ;Clear TMR0 & Prescaler
3.MOVLW  '00xx1111'b  ;These 3 lines (5, 6, 7)
4.OPTION              ; are required only if
                      ; desired
5.CLRWDT              ;PS<2:0> are 000 or 001
6.MOVLW  '00xx1xxx'b  ;Set Postscaler to
7.OPTION              ; desired WDT rate
```

To change prescaler from the WDT to the Timer0 module, use the sequence shown in Example 6-2. This sequence must be used even if the WDT is disabled. A `CLRWDT` instruction should be executed before switching the prescaler.

EXAMPLE 6-2: CHANGING PRESCALER (WDT→TIMER0)

```
CLRWDT                ;Clear WDT and
                      ;prescaler
MOVLW   'xxxx0xxx'    ;Select TMR0, new
                      ;prescale value and
                      ;clock source
OPTION
```

FIGURE 6-5: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER



Note: T0CS, T0SE, PSA, PS<2:0> are bits in the OPTION register.

7.2.5 INTERNAL 4 MHz RC OSCILLATOR

The internal RC oscillator provides a fixed 4 MHz (nominal) system clock at V_{DD} = 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 last address of memory, which contains the calibration value for the internal RC oscillator. This location is always 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 PIC16C505, only bits <7:2> of OSCCAL are implemented.

7.3 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

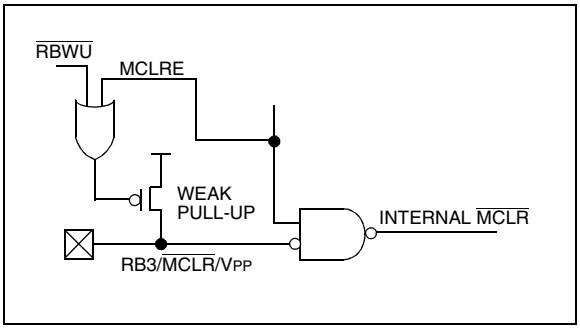
f) Wake-up from SLEEP on pin change

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 RBWUF 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 7-3 for a full description of reset states of all registers.

7.3.1 $\overline{\text{MCLR}}$ ENABLE

This configuration bit when unprogrammed (left in the '1' state) enables the external $\overline{\text{MCLR}}$ function. When programmed, the $\overline{\text{MCLR}}$ function is tied to the internal V_{DD}, and the pin is assigned to be a I/O. See Figure 7-6.

FIGURE 7-6: $\overline{\text{MCLR}}$ SELECT



7.4 Power-On Reset (POR)

The PIC16C505 family incorporates on-chip Power-On Reset (POR) circuitry, which provides an internal chip reset for most power-up situations.

The on chip POR circuit holds the chip in reset until V_{DD} has reached a high enough level for proper operation. To take advantage of the internal POR, program the RB3/$\overline{\text{MCLR}}$/V_{PP} pin as $\overline{\text{MCLR}}$ and tie through a resistor to V_{DD} or program the pin as RB3. An internal weak pull-up resistor is implemented using a transistor. Refer to Table 10-1 for the pull-up resistor ranges. This will eliminate external RC components usually needed to create a Power-on Reset. A maximum rise time for V_{DD} is specified. See Electrical Specifications for details.

When the device starts normal operation (exits the reset condition), device operating parameters (voltage, frequency, temperature, ...) must be met to ensure operation. If these conditions are not met, the device must be held in reset until the operating parameters are met.

A simplified block diagram of the on-chip Power-On Reset circuit is shown in Figure 7-7.

The Power-On Reset circuit and the Device Reset Timer (Section 7.5) circuit are closely related. On power-up, the reset latch is set and the DRT is reset. The DRT timer begins counting once it detects $\overline{\text{MCLR}}$ to be high. After the time-out period, which is typically 18 ms, it will reset the reset latch and thus end the on-chip reset signal.

A power-up example where $\overline{\text{MCLR}}$ is held low is shown in Figure 7-8. V_{DD} is allowed to rise and stabilize before bringing $\overline{\text{MCLR}}$ high. The chip will actually come out of reset T_{DRT} msec after $\overline{\text{MCLR}}$ goes high.

In Figure 7-9, the on-chip Power-On Reset feature is being used ($\overline{\text{MCLR}}$ and V_{DD} are tied together or the pin is programmed to be RB3.). The V_{DD} is stable before the start-up timer times out and there is no problem in getting a proper reset. However, Figure 7-10 depicts a problem situation where V_{DD} rises too slowly. The time between when the DRT senses that $\overline{\text{MCLR}}$ is high and when $\overline{\text{MCLR}}$ and V_{DD} actually reach their full value, is too long. In this situation, when the start-up timer times out, V_{DD} has not reached the V_{DD} (min) value and the chip may not function correctly. For such situations, we recommend that external RC circuits be used to achieve longer POR delay times (Figure 7-9).

Note:	When the device starts normal operation (exits the reset condition), device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not met, the device must be held in reset until the operating conditions are met.

For additional information refer to Application Notes "*Power-Up Considerations*" - AN522 and "*Power-up Trouble Shooting*" - AN607.

7.6.1 WDT PERIOD

The WDT has a nominal time-out period of 18 ms, (with no prescaler). If a longer time-out period is desired, a prescaler with a division ratio of up to 1:128 can be assigned to the WDT (under software control) by writing to the OPTION register. Thus, a time-out period of a nominal 2.3 seconds can be realized. These periods vary with temperature, VDD and part-to-part process variations (see DC specs).

Under worst case conditions (VDD = Min., Temperature = Max., max. WDT prescaler), it may take several seconds before a WDT time-out occurs.

7.6.2 WDT PROGRAMMING CONSIDERATIONS

The CLRWDT instruction clears the WDT and the postscaler, if assigned to the WDT, and prevents it from timing out and generating a device RESET.

The SLEEP instruction resets the WDT and the postscaler, if assigned to the WDT. This gives the maximum SLEEP time before a WDT wake-up reset.

FIGURE 7-11: WATCHDOG TIMER BLOCK DIAGRAM

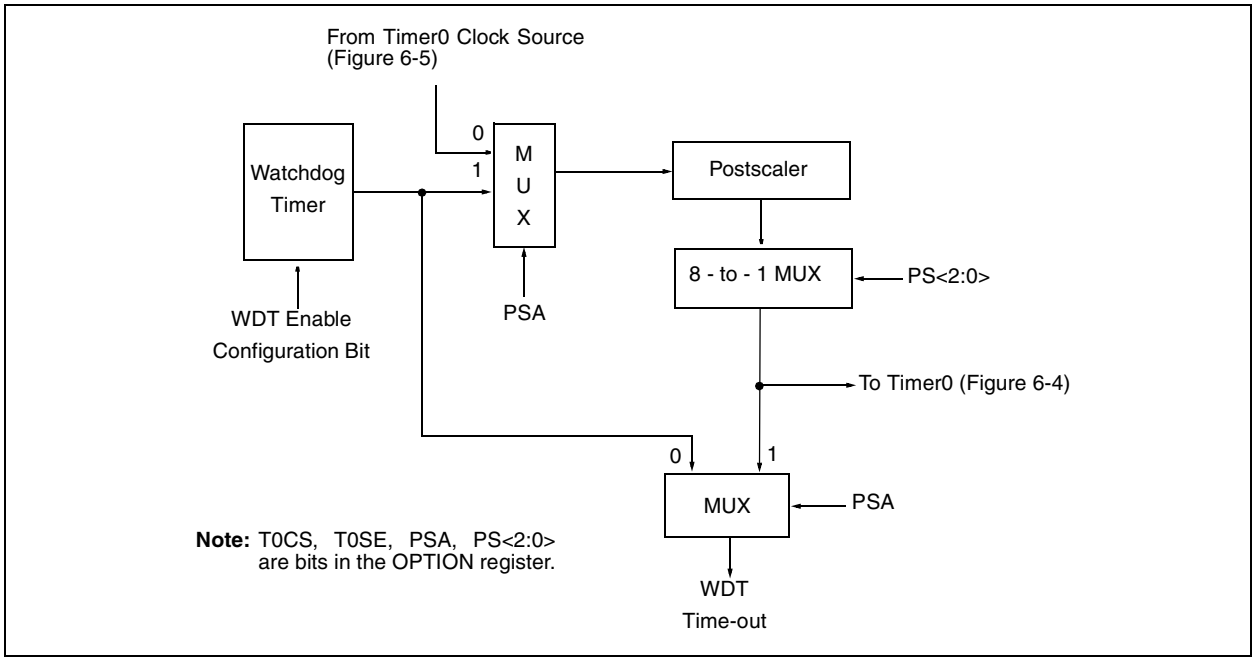


TABLE 7-6: SUMMARY OF REGISTERS ASSOCIATED WITH THE WATCHDOG TIMER

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	OPTION	$\overline{RBWU}$	$\overline{RBPU}$	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: Shaded boxes = Not used by Watchdog Timer, — = unimplemented, read as '0', u = unchanged.

7.7 Time-Out Sequence, Power Down, and Wake-up from SLEEP Status Bits ($\overline{TO}$/$\overline{PD}$/RBWUF)

The $\overline{TO}$, $\overline{PD}$, and RBWUF bits in the STATUS register can be tested to determine if a RESET condition has been caused by a power-up condition, a $\overline{MCLR}$ or Watchdog Timer (WDT) reset.

TABLE 7-7: $\overline{TO}$/$\overline{PD}$/RBWUF STATUS AFTER RESET

RBWUF	$\overline{TO}$	$\overline{PD}$	RESET caused by
0	0	0	WDT wake-up from SLEEP
0	0	u	WDT time-out (not from SLEEP)
0	1	0	$\overline{MCLR}$ wake-up from SLEEP
0	1	1	Power-up
0	u	u	$\overline{MCLR}$ not during SLEEP
1	1	0	Wake-up from SLEEP on pin change

Legend: u = unchanged

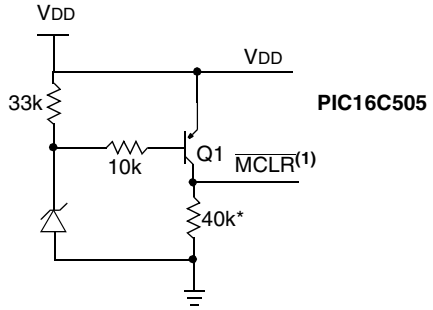
Note 1: The $\overline{TO}$, $\overline{PD}$, and RBWUF bits maintain their status (u) until a reset occurs. A low-pulse on the $\overline{MCLR}$ input does not change the $\overline{TO}$, $\overline{PD}$, and RBWUF status bits.

7.8 Reset on Brown-Out

A brown-out is a condition where device power (V_{DD}) dips below its minimum value, but not to zero, and then recovers. The device should be reset in the event of a brown-out.

To reset PIC16C505 devices when a brown-out occurs, external brown-out protection circuits may be built, as shown in Figure 7-12 and Figure 7-13.

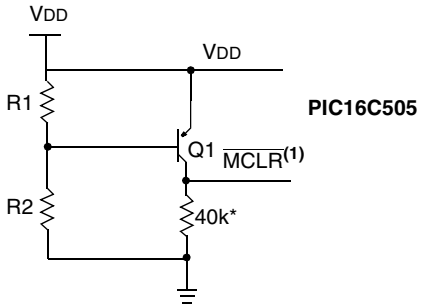
FIGURE 7-12: BROWN-OUT PROTECTION CIRCUIT 1



This circuit will activate reset when V_{DD} goes below Vz + 0.7V (where Vz = Zener voltage).

Note 1: Pin must be confirmed as $\overline{MCLR}$.

FIGURE 7-13: BROWN-OUT PROTECTION CIRCUIT 2

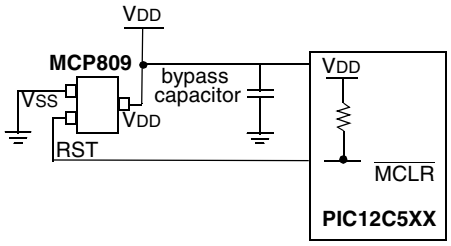


This brown-out circuit is less expensive, although less accurate. Transistor Q1 turns off when V_{DD} is below a certain level such that:

$$V_{DD} \bullet \frac{R1}{R1 + R2} = 0.7V$$

Note 1: Pin must be confirmed as $\overline{MCLR}$.

FIGURE 7-14: BROWN-OUT PROTECTION CIRCUIT 3



This brown-out protection circuit employs Microchip Technology's MCP809 microcontroller supervisor. There are 7 different trip point selections to accommodate 5V to 3V systems.

7.9 Power-Down Mode (SLEEP)

A device may be powered down (SLEEP) and later powered up (Wake-up from SLEEP).

7.9.1 SLEEP

The Power-Down mode is entered by executing a SLEEP instruction.

If enabled, the Watchdog Timer will be cleared but keeps running, the $\overline{TO}$ bit (STATUS<4>) is set, the $\overline{PD}$ bit (STATUS<3>) is cleared and the oscillator driver is turned off. The I/O ports maintain the status they had before the SLEEP instruction was executed (driving high, driving low or hi-impedance).

It should be noted that a RESET generated by a WDT time-out does not drive the $\overline{MCLR}$ pin low.

For lowest current consumption while powered down, the T0CKI input should be at VDD or VSS and the RB3/$\overline{MCLR}$/VPP pin must be at a logic high level (VIHMC) if $\overline{MCLR}$ is enabled.

7.9.2 WAKE-UP FROM SLEEP

The device can wake-up from SLEEP through one of the following events:

1. An external reset input on RB3/$\overline{MCLR}$/VPP pin, when configured as $\overline{MCLR}$.
2. A Watchdog Timer time-out reset (if WDT was enabled).
3. A change on input pin RB0, RB1, RB3 or RB4 when wake-up on change is enabled.

These events cause a device reset. The $\overline{TO}$, $\overline{PD}$, and RBWUF bits can be used to determine the cause of device reset. The $\overline{TO}$ bit is cleared if a WDT time-out occurred (and caused wake-up). The $\overline{PD}$ bit, which is set on power-up, is cleared when SLEEP is invoked. The RBWUF bit indicates a change in state while in SLEEP at pins RB0, RB1, RB3 or RB4 (since the last file or bit operation on RB port).

Caution:	Right before entering SLEEP, read the input pins. When in SLEEP, wake up occurs when the values at the pins change from the state they were in at the last reading. If a wake-up on change occurs and the pins are not read before reentering SLEEP, a wake-up will occur immediately even if no pins change while in SLEEP mode.

The WDT is cleared when the device wakes from sleep, regardless of the wake-up source.

7.10 Program Verification/Code Protection

If the code protection bit has not been programmed, the on-chip program memory can be read out for verification purposes.

The first 64 locations and the last location (OSCCAL) can be read, regardless of the code protection bit setting.

7.11 ID Locations

Four memory locations are designated as ID locations where the user can store checksum or other code-identification numbers. These locations are not accessible during normal execution, but are readable and writable during program/verify.

Use only the lower 4 bits of the ID locations and always program the upper 8 bits as '0's.

8.0 INSTRUCTION SET SUMMARY

Each PIC16C505 instruction is a 12-bit word divided into an OPCODE, which specifies the instruction type, and one or more operands which further specify the operation of the instruction. The PIC16C505 instruction set summary in Table 8-2 groups the instructions into byte-oriented, bit-oriented, and literal and control operations. Table 8-1 shows the opcode field descriptions.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator is used to specify which one of the 32 file registers is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the number of the file in which the bit is located.

For **literal and control** operations, 'k' represents an 8 or 9-bit constant or literal value.

TABLE 8-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1) The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0 (store result in W) d = 1 (store result in file register 'f') Default is d = 1
label	Label name
TOS	Top of Stack
PC	Program Counter
WDT	Watchdog Timer Counter
$\overline{TO}$	Time-Out bit
$\overline{PD}$	Power-Down bit
dest	Destination, either the W register or the specified register file location
[ ]	Options
( )	Contents
→	Assigned to
< >	Register bit field
∈	In the set of
italics	User defined term (font is courier)

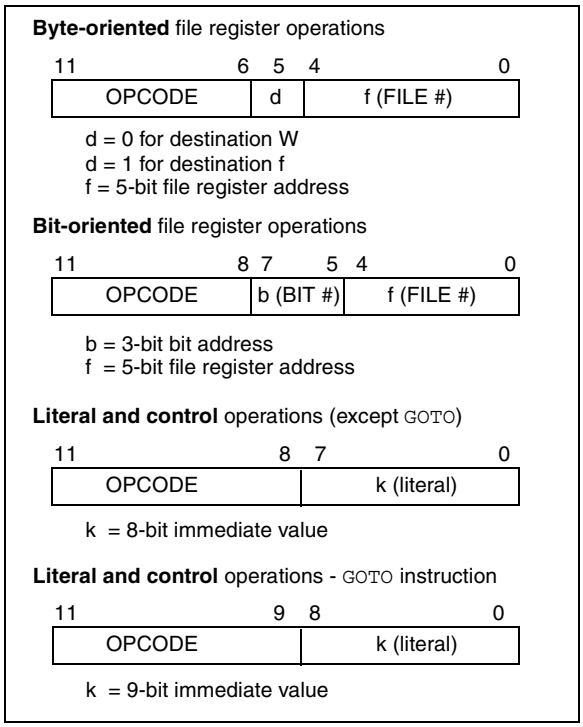
All instructions are executed within a single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 μs. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 μs.

Figure 8-1 shows the three general formats that the instructions can have. All examples in the figure use the following format to represent a hexadecimal number:

0xhhh

where 'h' signifies a hexadecimal digit.

FIGURE 8-1: GENERAL FORMAT FOR INSTRUCTIONS



CALL **Subroutine Call**

Syntax: [*label*] CALL k

Operands: $0 \le k \le 255$

Operation: (PC) + 1→ Top of Stack;
k → PC<7:0>;
(STATUS<6:5>) → PC<10:9>;
0 → PC<8>

Status Affected: None

Encoding:

1001	kkkk	kkkk

Description: Subroutine call. First, return address (PC+1) is pushed onto the stack. The eight bit immediate address is loaded into PC bits <7:0>. The upper bits PC<10:9> are loaded from STATUS<6:5>, PC<8> is cleared. CALL is a two cycle instruction.

Words: 1

Cycles: 2

Example:
```
HERE    CALL    THERE
```
Before Instruction
PC = address (HERE)
After Instruction
PC = address (THERE)
TOS = address (HERE + 1)

CLRF **Clear f**

Syntax: [*label*] CLRF f

Operands: $0 \le f \le 31$

Operation: 00h → (f);
1 → Z

Status Affected: Z

Encoding:

0000	011f	ffff

Description: The contents of register 'f' are cleared and the Z bit is set.

Words: 1

Cycles: 1

Example:
```
CLRF    FLAG_REG
```
Before Instruction
FLAG_REG = 0x5A
After Instruction
FLAG_REG = 0x00
Z = 1

CLRW **Clear W**

Syntax: [*label*] CLRW

Operands: None

Operation: 00h → (W);
1 → Z

Status Affected: Z

Encoding:

0000	0100	0000

Description: The W register is cleared. Zero bit (Z) is set.

Words: 1

Cycles: 1

Example:
```
CLRW
```
Before Instruction
W = 0x5A
After Instruction
W = 0x00
Z = 1

CLRWDT **Clear Watchdog Timer**

Syntax: [*label*] CLRWDT

Operands: None

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

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

Encoding:

0000	0000	0100

Description: The CLRWDT instruction resets the WDT. It also resets the prescaler, if the prescaler is assigned to the WDT and not Timer0. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

Words: 1

Cycles: 1

Example:
```
CLRWDT
```
Before Instruction
WDT counter = ?
After Instruction
WDT counter = 0x00
WDT prescale = 0
$\overline{TO}$ = 1
$\overline{PD}$ = 1

SLEEP **Enter SLEEP Mode**

Syntax: [*label*] SLEEP

Operands: None

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

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

Encoding:

0000	0000	0011

Description: Time-out status bit ($\overline{TO}$) is set. The power down status bit ($\overline{PD}$) is cleared.

RBWUF 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

9.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers and dsPIC® digital signal controllers are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB C Compiler for Various Device Families
  - HI-TECH C® for Various Device Families
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers
  - MPLAB ICD 3
  - PICkit™ 3 Debug Express
- Device Programmers
  - PICkit™ 2 Programmer
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits, and Starter Kits

9.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16/32-bit microcontroller market. The MPLAB IDE is a Windows® operating system-based application that contains:

- A single graphical interface to all debugging tools
  - Simulator
  - Programmer (sold separately)
  - In-Circuit Emulator (sold separately)
  - In-Circuit Debugger (sold separately)
- A full-featured editor with color-coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- High-level source code debugging
- Mouse over variable inspection
- Drag and drop variables from source to watch windows
- Extensive on-line help
- Integration of select third party tools, such as IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either C or assembly)
- One-touch compile or assemble, and download to emulator and simulator tools (automatically updates all project information)
- Debug using:
  - Source files (C or assembly)
  - Mixed C and assembly
  - Machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.

9.11 PICkit 2 Development Programmer/Debugger and PICkit 2 Debug Express

The PICkit™ 2 Development Programmer/Debugger is a low-cost development tool with an easy to use interface for programming and debugging Microchip's Flash families of microcontrollers. The full featured Windows® programming interface supports baseline (PIC10F, PIC12F5xx, PIC16F5xx), midrange (PIC12F6xx, PIC16F), PIC18F, PIC24, dsPIC30, dsPIC33, and PIC32 families of 8-bit, 16-bit, and 32-bit microcontrollers, and many Microchip Serial EEPROM products. With Microchip's powerful MPLAB Integrated Development Environment (IDE) the PICkit™ 2 enables in-circuit debugging on most PIC® microcontrollers. In-Circuit-Debugging runs, halts and single steps the program while the PIC microcontroller is embedded in the application. When halted at a breakpoint, the file registers can be examined and modified.

The PICkit 2 Debug Express include the PICkit 2, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

9.12 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at V_{DDMIN} and V_{DDMAX} for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages and a modular, detachable socket assembly to support various package types. The ICSP™ cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices and incorporates an MMC card for file storage and data applications.

9.13 Demonstration/Development Boards, Evaluation Kits, and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web page (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

10.2 DC CHARACTERISTICS: PIC16LC505-04 (Commercial, Industrial)

DC Characteristics Power Supply Pins			Standard Operating Conditions (unless otherwise specified) Operating Temperature 0°C ≤ TA ≤ +70°C (commercial) –40°C ≤ TA ≤ +85°C (industrial)				
Parm. No.	**Characteristic**	**Sym**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
D001	Supply Voltage	VDD	2.5	—	5.5	V	See Figure 10-1 through Figure 10-3
D002	RAM Data Retention Voltage(2)	VDR	—	1.5*	—	V	Device in SLEEP mode
D003	VDD Start Voltage to ensure Power-on Reset	VPOR	—	VSS	—	V	See section on Power-on Reset for details
D004	VDD Rise Rate to ensure Power-on Reset	SVDD	0.05*	—	—	V/ms	See section on Power-on Reset for details
D010	Supply Current(3)	IDD	—	0.8	1.4	mA	FOSC = 4MHz, VDD = 5.5V, WDT disabled (Note 4)*
			—	0.4	0.8	mA	FOSC = 4MHz, VDD = 2.5V, WDT disabled (Note 4)
			—	15	23	µA	FOSC = 32kHz, VDD = 2.5V, WDT disabled (Note 6)
D020	Power-Down Current (5)	IPD	—	0.25	3	µA	VDD = 2.5V (Note 6)
			—	0.25	4	µA	VDD = 3.0V * (Note 6)
			—	3	8	µA	VDD = 5.5V Industrial
D022	WDT Current(5)	ΔIWDT	—	2.0	4	µA	VDD = 2.5V (Note 6)
1A	LP Oscillator Operating Frequency	FOSC	0	—	200	kHz	All temperatures
	RC Oscillator Operating Frequency		0	—	4	MHz	All temperatures
	XT Oscillator Operating Frequency		0	—	4	MHz	All temperatures
	HS Oscillator Operating Frequency		0	—	4	MHz	All temperatures

* These parameters are characterized but not tested.

Note 1: Data in the Typical ("Typ") column is based on characterization results at 25°C. This data is for design guidance only and is not tested.

2: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

3: The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, oscillator type, bus rate, internal code execution pattern and temperature also have an impact on the current consumption.

a) The test conditions for all IDD measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tristated, pulled to VSS, T0CKI = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

b) For standby current measurements, the conditions are the same, except that the device is in SLEEP mode.

4: Does not include current through Rext. The current through the resistor can be estimated by the formula: IR = VDD/2Rext (mA) with Rext in kOhm.

5: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD or VSS.

6: Commercial temperature range only.

10.5 Timing Diagrams and Specifications

FIGURE 10-5: EXTERNAL CLOCK TIMING - PIC16C505

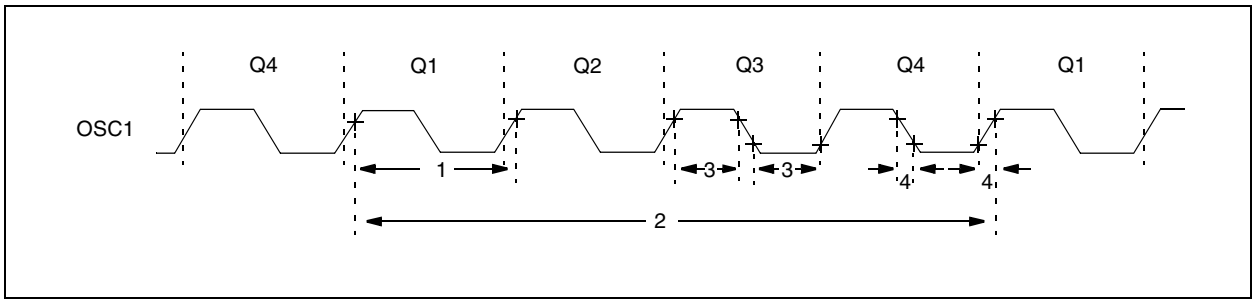


TABLE 10-2: EXTERNAL CLOCK TIMING REQUIREMENTS - PIC16C505

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 10.1						
Parameter No.	**Sym**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
1A	FOSC	External CLKIN Frequency(2)	DC	—	4	MHz	XT osc mode
			DC	—	4	MHz	HS osc mode (PIC16C505-04)
			DC	—	20	MHz	HS osc mode (PIC16C505-20)
			DC	—	200	kHz	LP osc mode
		Oscillator Frequency(2)	DC	—	4	MHz	EXTRC osc mode
			0.1	—	4	MHz	XT osc mode
			4	—	4	MHz	HS osc mode (PIC16C505-04)
			DC	—	200	kHz	LP osc mode
1	TOSC	External CLKIN Period(2)	250	—	—	ns	XT osc mode
			50	—	—	ns	HS osc mode (PIC16C505-20)
				—	—	µs	LP osc mode
		Oscillator Period(2)	250	—	—	ns	EXTRC osc mode
			250	—	10,000	ns	XT osc mode
			250	—	250	ns	HS ocs mode (PIC16C505-04)
			50	—	250	ns	HS ocs mode (PIC16C505-20)
			5	—	—	µs	LP osc mode
2	TCY	Instruction Cycle Time	—	4/FOSC	DC	ns	
			200	—		ns	

* 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.

11.0 DC AND AC CHARACTERISTICS - PIC16C505

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 11-1: CALIBRATED INTERNAL RC FREQUENCY RANGE VS. TEMPERATURE (VDD = 5.0V) (INTERNAL RC IS CALIBRATED TO 25°C, 5.0V)

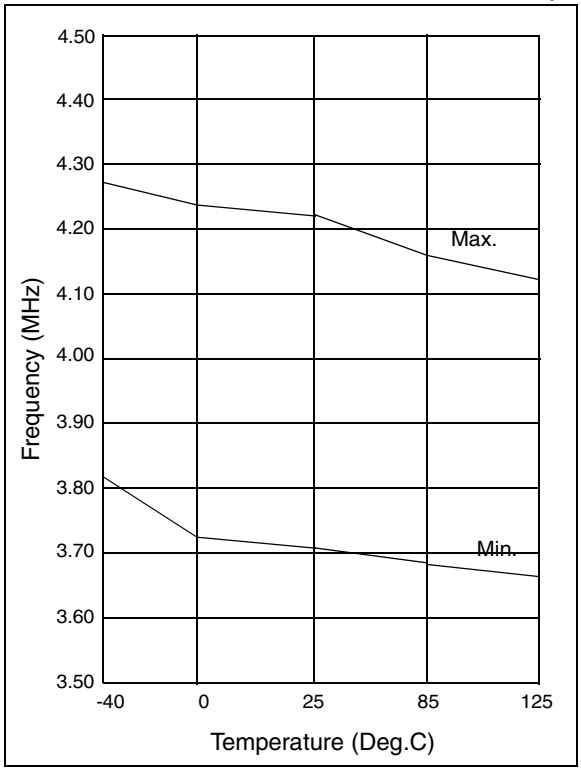


FIGURE 11-2: CALIBRATED INTERNAL RC FREQUENCY RANGE VS. TEMPERATURE (VDD = 2.5V) (INTERNAL RC IS CALIBRATED TO 25°C, 5.0V)

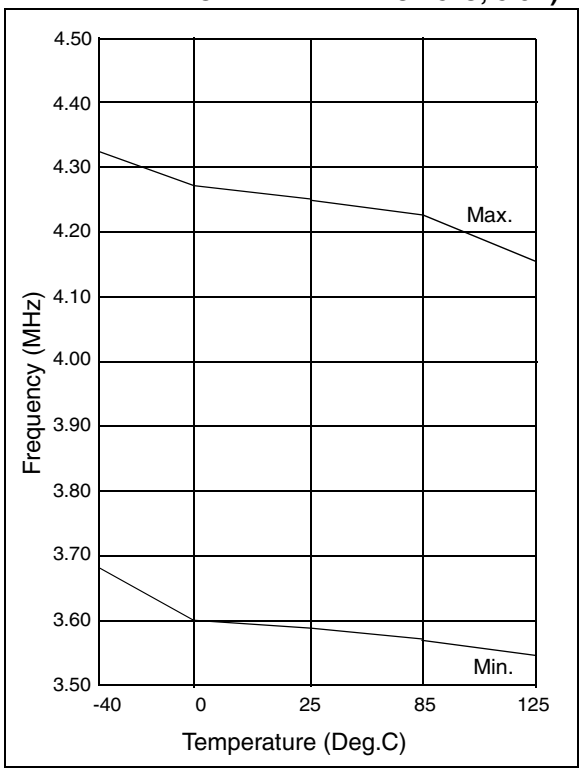


TABLE 11-1: DYNAMIC IDD (TYPICAL) - WDT ENABLED, 25°C

Oscillator	Frequency	VDD = 3.0V(1)	VDD = 5.5V
External RC	4 MHz	240 µA(2)	800 µA(2)
Internal RC	4 MHz	320 µA	800 µA
XT	4 MHz	300 µA	800 µA
LP	32 kHz	19 µA	50 µA
HS	20 MHz	N/A	4.5 mA

Note 1: LP oscillator based on VDD = 2.5V

2: Does not include current through external R&C.

FIGURE 11-3: WDT TIMER TIME-OUT PERIOD vs. VDD

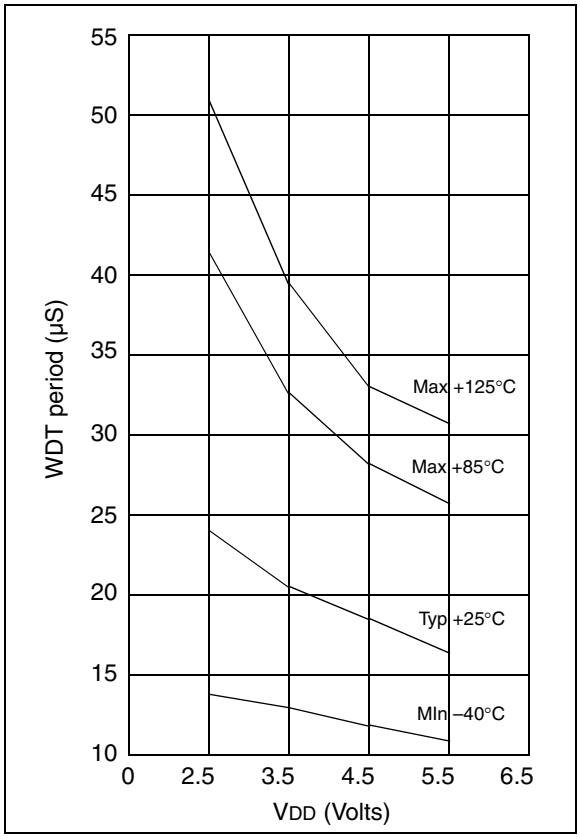
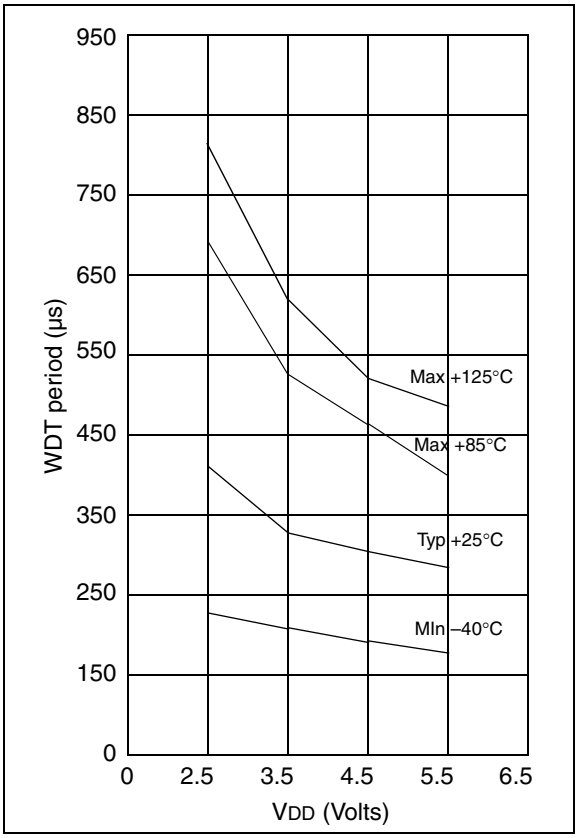


FIGURE 11-4: SHORT DRT PERIOD VS. VDD



APPENDIX A: REVISION HISTORY

Revision D (June 2012)

- **Section 12.0 “Packaging Information”** was updated with current package outline drawings.
- Removed Section 2.1 “UV Erasable Devices” section.

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

TO: Technical Publications Manager Total Pages Sent _____

RE: Reader Response

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

Telephone: (_____) _____ - _____ FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? ___ Y ___ N

Device: PIC16C505 Literature Number: DS40192D

Questions:

1. What are the best features of this document?

2. How does this document meet your hardware and software development needs?

3. Do you find the organization of this document easy to follow? If not, why?

4. What additions to the document do you think would enhance the structure and subject?

5. What deletions from the document could be made without affecting the overall usefulness?

6. Is there any incorrect or misleading information (what and where)?

7. How would you improve this document?