

---

---

## 第 27 章 器件配置位

---

---

### 目录

本章包括以下一些主要内容：

27.1 简介 .....	27-2
27.2 配置字位 .....	27-4
27.3 编程校验 / 代码保护 .....	27-8
27.4 识别码 ID 的位置 .....	27-9
27.5 设计技巧 .....	27-10
27.6 相关应用笔记 .....	27-11
27.7 版本历史 .....	27-12

## 27.1 简介

器件的配置位允许用户根据应用需要配置器件。当器件上电时，这些位的状态决定了器件的工作模式。**27.2 “配置字位”** 小节对这些配置位及其所能配置的模式进行了讨论。这些配置位的程序存储器映射位置为 2007h。器件正常运行时，无法对该位置进行存取（只能在编程模式下存取）。

可以通过对配置位编程（读为 '0'）或不编程（读为 '1'）来选择不同的器件配置。对配置位编程后，是否能够更改其设置取决于器件的存储工艺和封装形式。

对于只读存储器 (ROM) 器件，这些配置位在 ROM 代码提交时即被确定，且一旦器件掩膜完成，即无法更改（若要更改，则需新的掩膜代码）。

对一次可编程 (OTP) 器件，一旦这些位被编程（为 '0'），就无法更改了。

对窗口型 EPROM 器件，一旦这些位被编程（为 '0'）后，必须通过紫外线擦除，使器件的配置字返回到已擦除状态。紫外线擦除器件时，也擦除了程序存储器的内容。

对闪存器件，这些位可擦除并重新编程。

<b>注：</b> Microchip 建议不要将窗口型器件进行代码保护。
---------------------------------------

因文档格式的需要，将 27.2 节“配置字位”置于下页。

## 27.2 配置字位

这些配置位规定了器件的部分工作模式，可通过器件编程器或中档器件的在线串行编程 (ICSP™) 功能对其编程。器件无法读取这些位的值，而且当用户在器件编程器中选好器件时，配置位的地址就自动确定了。欲获得更多信息，请参见器件的编程规范说明。

**注 1:** 请始终确保器件编程器上的所选器件就是您编程的器件

**注 2:** Microchip 建议将所需配置位的状态嵌入应用的源代码中。通过 MPASM 汇编器中的 CONFIG 指令，可以轻松做到这一点。请参看 27.2.1 “MPASMTM 的 CONFIG 指令” 小节

### CP1:CP0: 代码保护位

11 = 代码保护关

10 = 参见器件数据手册

01 = 参见器件数据手册

00 = 所有存储器均受代码保护

**注:** 有些器件使用较多或较少的位数来配置代码保护。目前就一些只使用一位 (CP0) 的器件，保护位的说明如下：

1 = 代码保护关

0 = 代码保护开

### DP: 数据 EEPROM 存储器的代码保护位

1 = 代码保护关

0 = 数据 EEPROM 存储器受代码保护

**注:** 对于具有数据 EEPROM 存储器的 ROM 程序存储器器件，使用该位。

### BODEN: 欠压复位 (BOR) 使能位

1 = BOR 使能

0 = BOR 禁止

**注:** 无论  $\overline{\text{PWRTE}}$  位的值为何，使能欠压复位即自动使能了上电定时器 (PWRT)。请确保使能欠压复位时，也使能了上电定时器。

### $\overline{\text{PWRTE}}$ : 上电定时器 (PWRT) 使能位

1 = PWRT 禁止

0 = PWRT 使能

**注 1:** 无论  $\overline{\text{PWRTE}}$  位的值为何，使能欠压复位即自动使能了上电定时器 (PWRT)。请确保使能欠压复位时，也使能了上电定时器。

**注 2:** 在一些早期的 PICmicro® 单片机中，该位的极性被保留。

**注 3:**

### MCLR: $\overline{\text{MCLR}}$ 引脚功能选择位

1 = 引脚功能为  $\overline{\text{MCLR}}$

0 = 引脚功能为数字 I/O。MCLR 在内部连接到 VDD 上。

### WDTE: 看门狗定时器 (WDT) 使能位

1 = WDT 使能

0 = WDT 禁止

## FOSC1:FOSC0: 振荡器选择位

11 = RC 振荡器  
10 = HS 振荡器  
01 = XT 振荡器  
00 = LP 振荡器

## FOSC2:FOSC0: 振荡器选择位

111 = EXTRC 振荡器, 带 CLKOUT 时钟输出  
110 = EXTRC 振荡器  
101 = INTRC 振荡器, 带 CLKOUT 时钟输出  
100 = INTRC 振荡器  
011 = 保留  
010 = HS 振荡器  
001 = XT 振荡器  
000 = LP 振荡器

未用: 读为 '1'

### 图注

R = 可读位

P = 可编程位

U = 未用位, 读为 '0'

- n = POR 复位后的值

u = 编程后状态不变

### 注:

配置位的位置视器件的不同而不同。欲了解配置位的位置, 请参看器件编程规范说明。使用 Microchip 的器件编程器时, 不要求您了解配置位的具体位置。

## 27.2.1 MPASM™ 的 CONFIG 指令

Microchip 的汇编器 MPASM™ 中有一个很实用的功能，即允许您在源代码文件中规定本程序中配置位所选择的状态。这样可保证在对器件进行编程时，所需的配置位也同时被编程。这最大限度地减小了将错误的配置烧录进器件的风险，使您无需为器件为何不能正常工作而苦恼。

例 27-1 给出了一个使用 CONFIG 指令的模板程序。

### 例 27-1: 使用 CONFIG 指令的模板程序

```
LIST    p = p16C77      ; List Directive,
; Revision History
;
#include <P16C77.INC>    ; Microchip Device Header File
;
#include <MY_STD.MAC>    ; File which includes my standard macros
#include <APP.MAC>       ; File which includes macros specific
                        ; to this application
;
; Specify Device Configuration Bits
;
__CONFIG  _XT_OSC & _PWRTE_ON & _BODEN_OFF & _CP_OFF & _WDT_ON
;
org 0x00          ; Start of Program Memory
RESET_ADDR :      ; First instruction to execute after a reset

end
```

目前在 Microchip 头文件中直接使用 CONFIG 指令的符号显示在表 27-1 中。欲了解您所使用的器件具备哪些可用符号，请参见 Microchip 所提供的该器件的头文件。

**注：** 只要 (在 LIST 和 INCLUDE 文件指令中) 正确指定了器件，即可保证所有位的极性。

表 27-1: \_\_CONFIG 指令符 (来自 Microchip 的头文件)

特性	符号
振荡器	__RC_OSC
	__EXTRC_OSC
	__EXTRC_OSC_CLKOUT
	__EXTRC_OSC_NOCLKOUT
	__INTRC_OSC
	__INTRC_OSC_CLKOUT
	__INTRC_OSC_NOCLKOUT
	__LP_OSC
	__XT_OSC
看门狗定时器	__WDT_ON
	__WDT_OFF
上电定时器	__PWRTE_ON
	__PWRTE_OFF
欠压复位	__BODEN_ON
	__BODEN_OFF
总清零使能	__MCLRE_ON
	__MCLRE_OFF
代码保护	__CP_ALL
	__CP_ON
	__CP_75
	__CP_50
代码保护数据 EEPROM	__CP_OFF
	__DP_ON
代码保护校准空间	__DP_OFF
	__CPC_ON
	__CPC_OFF

注 1: 任一器件不一定具备所有上述配置位符号。欲了解某一器件具备哪些符号, 请参见 Microchip 所提供的该器件的头文件。

## 27.3 编程校验 / 代码保护

若代码保护位未被烧录，则可读出片上程序存储器的内容以便校验。

**注：** Microchip 建议不要将窗口型器件进行代码保护。

### 27.3.1 ROM 器件

当一个 ROM 器件也具备数据 EEPROM 存储器时，则可额外实现一个代码保护配置。程序存储器的配置位是作为部分 ROM 代码提交的。数据 EEPROM 存储器的代码保护配置位将成为一个 EEPROM 位。在 ROM 器件测试完毕后，EEPROM 数据存储器的代码保护位将被烧录成与程序存储器代码保护位相同的状态，即当程序存储器代码保护被关闭时，数据 EEPROM 代码保护也被关闭，所有其它选择均将打开数据 EEPROM 的代码保护功能。

有些应用中，如果器件已受代码保护，而在应用交付前尚需对 EEPROM 进行编程，则需先将整个数据 EEPROM 擦除。器件的编程规范中详细说明了擦除步骤。Microchip 的器件编程器贯彻了指定的步骤。完成上述步骤后，EEPROM 存储器的代码保护即被关闭。这样即可将所需数据烧录到 EEPROM 中。对数据 EEPROM 阵列的编程完成后，应按需要将数据 EEPROM 的代码保护配置位编程。

## 27.4 识别码 ID 的位置

识别码 ID 指定了四个位置 (2000h - 2003h)，供用户存储校验和或其它识别代码。器件正常运行时，这些位置是不可存取的，但在编程 / 校验时可以被读写。建议只使用识别码 ID 位置的最低 4 位。

## 27.5 设计技巧

**问 1:** *我有一个 JW 型器件但无法再对它编程（读出的数据是乱码或者全是 '0'）。这是什么问题？*

**答 1:**

没有问题。可能是您对器件进行了代码保护。如果是这样，这个器件就不能再用了。详细内容请参见 [27.3 “编程校验 / 代码保护”](#) 小节。

**问 2:** *从 PIC16C74 转换到 PIC16C74A 后，我的应用就无法正常工作了。*

**答 2:**

1. 您是否在 INCLUDE 文件和 LIST 指令中指定了 PIC16C74A 并重新编译了源程序？我们强烈建议使用 CONFIG 指令。
2. 在器件编程器中您是否指定了 PIC16C74A？配置位是否全都符合要求？

**问 3:** *对器件进行擦除时，程序存储器已擦空，但配置字却没有被擦除。*

**答 3:**

这是设计原因。另请记住 Microchip 建议不要将窗口型器件代码保护。

## 27.6 相关应用笔记

本部分列出了与本章内容相关的应用笔记。这些应用笔记并非都是专门针对中档单片机系列而写的 ( 即有些针对低档系列, 有些针对高档系列 ), 但其概念是相近的, 经过适当修改并受一定限制即可使用。目前与配置字相关的应用笔记有:

标题

应用笔记 #

目前没有相关的应用笔记

## 27.7 版本历史

### 版本 A

这是描述配置字的初始发行版。