
PIC18F到PIC24F软件移植指南

简介

本文档总结了从PIC18F移植到PIC24F时最常见的兼容性问题。本文档适用于需要将固件移植到PIC24F的PIC18F开发人员。本软件移植指南旨在针对移植代码时经常遇到的兼容性问题提供与软件和工具相关的其他移植注意事项。有关针对硬件和外设的移植的更多信息，请参见《PIC18F到PIC24F以及PIC24F到SAMD2x的移植和性能增强指南》(DS00002991C_CN)。

本文档假定固件是通过采用出厂设置的MPLAB® XC8 v2.05、MPLAB XC16 v1.36B和MPLAB X IDE v5.20以C语言编写。这些工具的未来版本可能会更改与本文档中所述问题有关的默认行为。

有关代码兼容性的完整优质信息，请参见编译器用户指南。强烈建议查阅以下用户指南：

《MPLAB® XC16 C编译器用户指南》
(DS50002071E_CN)

- 第2章“通用C接口”
- 附录B“嵌入式编译器兼容模式”

“MPLAB® XC8 C Compiler User's Guide for PIC® MCU”
(DS50002737)

欲了解更多移植资源，请访问www.microchip.com/migration

使用MPLAB®代码配置器

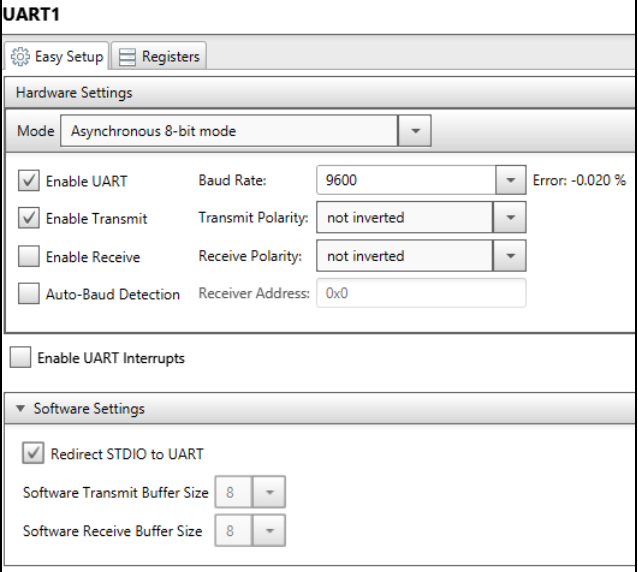
MPLAB代码配置器(MPLAB Code Configurator, MCC)是一款免费的图形编程环境，可生成简单易懂的C语言代码，将其无缝插入项目中。MCC采用直观界面，可针对应用来使能和配置众多外设及功能。MCC集成到MPLAB X IDE中，提供了一个强大易用的开发平台。尽管PIC18F和PIC24F的底层硬件外设有所不同，但为两个系列器件生成的许多外设驱动程序共用通用的驱动程序接口。通用编程接口有助于简化应用程序代码的移植。即使现有的PIC18F固件未使用MCC生成的代码，但利用MCC来生成PIC24F代码仍然大有裨益，因为这有助于缩短硬件学习曲线。此外，通用中间件库(例如USB和文件系统)也作为MCC的一部分提供。图1给出了PIC18F和PIC24F之间关于MCC对UART支持的比较。欲了解更多信息，请访问www.microchip.com/mcc。

PIC18F 到 PIC24F 系列

图 1: PIC18F 和 PIC24F 之间关于 MCC 对 UART 支持的比较

MCC 用户界面
Easy Setup (轻松设置) 视图简单直观, 便于配置各种外设。在本示例中, 开发人员无需学习如何计算波特率寄存器值以获得 9600 波特率, 因为该值是由 MCC 自动计算的。

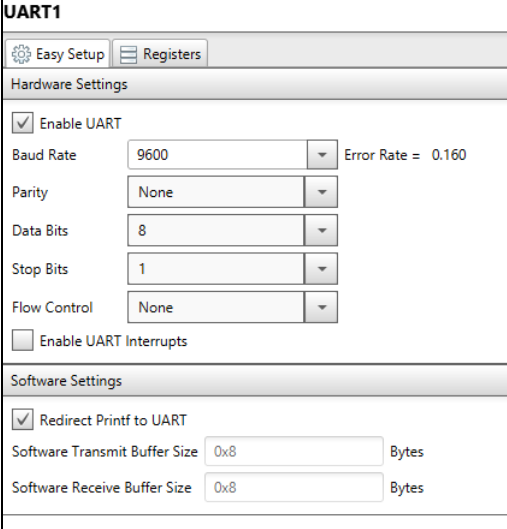
PIC18F



↓

```
void UART1_Initialize(void);
uint8_t UART1_Read(void);
void UART1_Write(uint8_t txData);
```

PIC24F



↓

```
void UART1_Initialize(void);
uint8_t UART1_Read(void);
void UART1_Write(uint8_t txData);
```

MCC 生成的 API

编写中断服务程序

编译器用户指南提供了有关为 PIC18F 和 PIC24F 编写中断服务程序 (Interrupt Service Routine, ISR) 的详细信息。

建议查阅以下内容：

- “MPLAB® XC8 C Compiler User's Guide for PIC® MCU” (DS50002737) 的第 4.9.1 节 “Writing an Interrupt Service Routine”
- 《MPLAB® XC16 C 编译器用户指南》 (DS50002071E_CN)，具体章节如下：
 - 第 2.5.10 节 “中断函数”
 - 第 3.4.6 节 “中断”
 - 第 10.4.2.3 节 “ISR 注意事项”
 - 第 14 章 “中断”

本部分举例说明了三种常用 ISR 定义。了解不同 ISR 定义方式之间的差异对于代码移植很有帮助。

例 1 给出了使用传统中断模式的 PIC18F 的 ISR 定义，此类器件只有高优先级和低优先级中断向量。每个 ISR 通过查看中断允许位和中断标志位来确定中断源。

例 1: PIC18F 传统中断模式 (高优先级和低优先级, 没有独立的中断向量)

```
#include <stdint.h>
int16_t tick_count;

void __interrupt(high_priority) tcInt(void)
{
    if (TMR0IE && TMR0IF) { // any timer 0 interrupts?
        TMR0IF=0;
        ++tick_count;
    }
    if (TMR1IE && TMR1IF) { // any timer 1 interrupts?
        TMR1IF=0;
        tick_count += 100;
    }
    // process other interrupt sources here, if required
    return;
}
```

PIC18F 到 PIC24F 系列

例2给出了使用向量中断模式的PIC18F的ISR定义。请注意，由于每个ISR仅关联一个中断源，中断代码无需确定中断源，因此速度更快。具有向量中断模式的PIC18F器件用编号标识每个中断。如果使能了中断向量表，则可以使用__interrupt()中的irq()参数来指定该编号，开发人员也可以使用等同于该编号的编译器定

义符号。开发人员可通过打开pic_chipinfo.html或pic18_chipinfo.html，然后选择目标器件的方式来查看包含所有中断编号、符号和说明的列表。这两个文件均位于MPLAB XC8编译器安装目录下的docs目录中。

例2: PIC18F 向量中断模式

```
#include <stdint.h>
int16_t tick_count;

void __interrupt(irq(TMR0),high_priority) tc0Int(void)
{
    TMR0IF=0;
    ++tick_count;
    return;
}

void __interrupt(irq(TMR1),high_priority) tc1Int(void)
{
    TMR1IF=0;
    tick_count += 100;
    return;
}
```

例3给出了PIC24F器件的ISR定义。每个ISR有一个中断源。每个ISR名称都是一个关键字（本例中为_T1Interrupt）。开发人员可通过打开XC16MasterIndex.htm，然后打开“Interrupt Vector Tables Reference”（中断向量表参考）链接的方式来查看包含所有向量关键字的列表。XC16MasterIndex.htm文件位于MPLAB XC16编译器安装目录下的docs目录中。对于所有没有特定处理程

序的中断向量，将安装默认的中断处理程序。默认的中断处理程序由链接器提供，用于复位器件。此外，应用程序也可使用名称_DefaultInterrupt来声明一个中断函数，以此提供默认的中断处理程序。使用interrupt属性时，如果未指定auto_psv或no_auto_psv，则将产生警告并假定为auto_psv。有关使用auto_psv与no_auto_psv的更多详细信息，请参见《MPLAB® XC16 C编译器用户指南》的第10.4.2.3节。

例3: PIC24F

```
void __attribute__((interrupt, no_auto_psv)) _T1Interrupt(void)
{
    IFS0bits.T1IF = false;
}
```

数据对齐

PIC24F 器件架构要求字在偶数字节边界上对齐，并且 MPLAB XC16 默认行为是对齐，以符合缩短代码长度和提高速度的要求。例4中的代码对于 PIC18F 和 MPLAB XC8 将输出“3”，而相同的代码在 PIC24F 和 MPLAB XC16 上则会输出“4”，这是因为需要通过额外填充使成员“x”在偶数字节边界上对齐。在例4中，如果“data”用于与通信无关的用途，则为 PIC18F 编写的原始应用程序可能会在 PIC24F 上正确运行。但是，如果“data”用于与通信相关的功能，则应用程序可能会出现运行时错误，因为 PIC24F 应用程序与其任何通信对象之间的预期字节数都不再匹配。

例4:

```
#include <stdint.h>

struct
{
    uint8_t c;
    uint16_t x;
} data;

printf("%u", sizeof(data));
```

为了解决不匹配问题，常用的移植做法是指定 **packed** 属性。**packed** 属性指定结构体成员应采用尽可能小的对齐方式。在例5中，成员“x”被压缩，它紧跟在成员“c”之后而不进行对齐填充。该代码将在 PIC24F 上输出“3”。

例5:

```
#include <stdint.h>

struct __attribute__((packed))
{
    uint8_t c;
    uint16_t x;
} data;

printf("%u", sizeof(data));
```

采用 **packed** 属性时，成员“x”跨两个字进行分配。编译后的代码会将成员“x”视为两个单独的字节进行访问，因此会导致代码变长、速度变慢。开发人员应事先考虑到这些负面影响，然后再有选择地应用 **packed** 属性。其他指定压缩的方法（例如 `#pragma pack(1)` 和编译器选项 `-fpack-struct`）很少会选择，应谨慎使用。更多的时候会选择使用 `#pragma pack`，如例6所示。

例6:

```
#include <stdint.h>

#pragma pack(push,1)
struct
{
    uint8_t c;
    uint16_t x;
} data;
#pragma pack(pop)

printf("%u", sizeof(data));
```

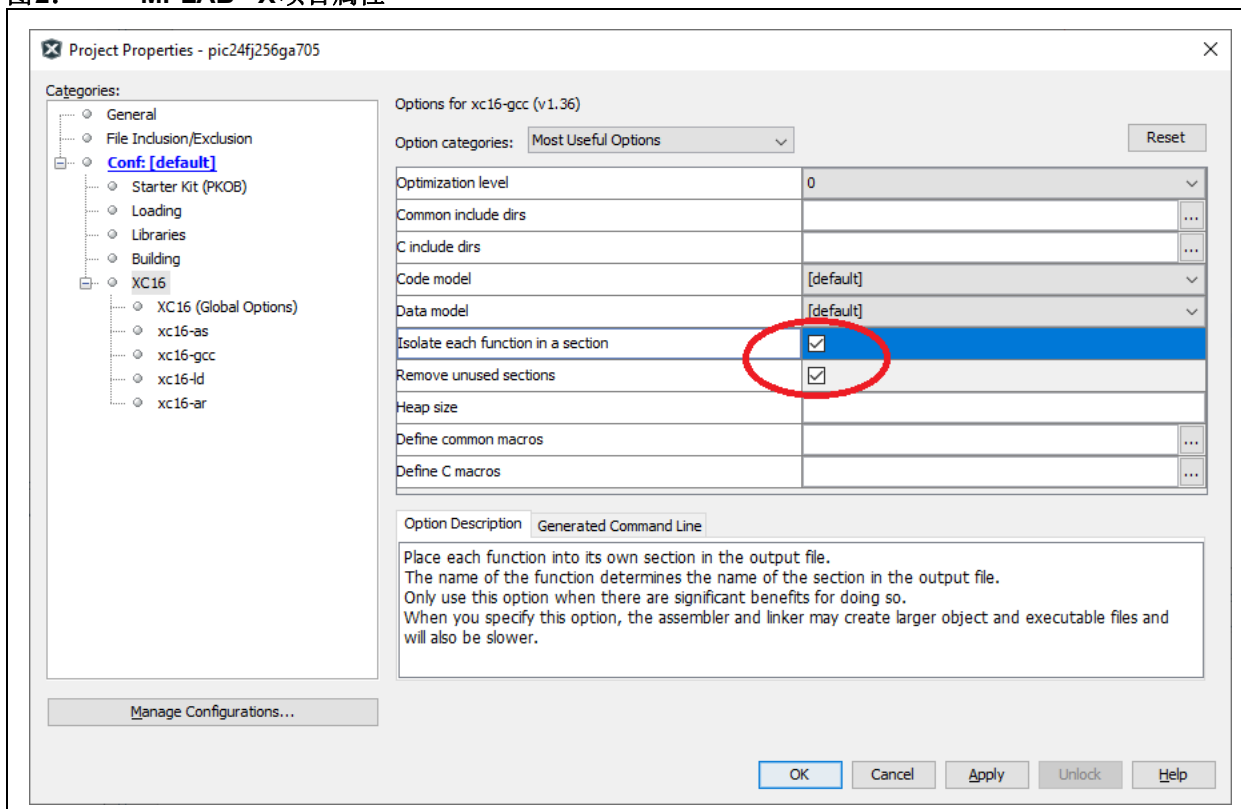
PIC18F 到 PIC24F 系列

未使用函数的管理

默认情况下，MPLAB XC8 会从项目编译中删除未使用的函数，而 MPLAB XC16 不会。在 MPLAB XC16 中，需通过设置 **Isolate each function in a section**（将每个函数单独放在一个段中）和 **Remove unused sections**（删除未使用的段）来实现上述行为。这些设置可通过 MPLAB X Project Properties（项目属性）来使能，如图 2 所示。这些设置在使用编译器优化时同样适用。

如果使用的第三方库以源代码形式提供，并且其函数未分配到不同的编译单元中，则开发人员应考虑使能这些设置。给定应用程序中可能未使用第三方库的所有函数和功能。开发人员应分别使能和禁止这些设置，然后比较哪一种情况在代码长度和运行速度方面更具优势再加以使用。

图 2: MPLAB® X 项目属性



延时宏

MPLAB XC8 和 XC16 均提供基于时间的内置延时宏 `__delay_ms()` 和 `__delay_us()`，但设置有所不同。

MPLAB XC8 需要将 `_XTAL_FREQ` 定义为等于系统频率 (`FOSC`) 并包含 `xc.h`。

MPLAB XC16 需要将 `FCY` 定义为等于指令频率并包含 `libpic30.h`。

从 PIC18F 移植到 PIC24F 时，首要的考虑因素是 PIC24F 内核使用 `Fosc/2` 作为其指令时钟，而 PIC18F 器件则使用 `Fosc/4` 指令时钟。针对 MPLAB XC8 的例 7 假定系统频率为 8 MHz，指令频率为 2 MHz。

例 7:

```
#define _XTAL_FREQ 8000000UL
#include <xc.h>

void f(void)
{
    __delay_ms(1);
    __delay_us(1000);
}
```

针对 MPLAB XC16 的例 8 假定系统频率为 8 MHz，指令频率为 4 MHz。

例 8:

```
#define FCY 4000000UL
#include <libpic30.h>

void f(void)
{
    __delay_ms(1);
    __delay_us(1000);
}
```

另外，开发人员也可以使用 MPLAB 代码配置器 Foundation Services Library (基础服务库) (v0.1.33 或更高版本) 的 DELAY 模块，该模块为 PIC18F 和 PIC24F 提供了通用的延时抽象，如例 9 所示。

例 9:

```
#include "mcc_generated_files/delay.h"

void f(void)
{
    DELAY_milliseconds(1);
    DELAY_microseconds(1000);
}
```

非标准数据类型

MPLAB XC8 支持非标准 `__bit` 和 24 位数据类型 (例 10)：

例 10:

```
static bit flag; //C90 mode only
static __bit initialized = 0;

short long x; //C90 mode only
__int24 y;
__uint24 z;
```

PIC18F 开发人员使用这些非标准数据类型的原因是为了减少 RAM 使用量。8 位对象被压缩到存储空间的每个字节中，并且 24 位类型小于 32 位类型。

但是，MPLAB XC16 不支持 `__bit` 或 24 位数据类型。编译预计会失败。常用的移植做法是将 `__bit` 转换为 `bool` 类型，将 24 位转换为 32 位类型 (例 11)：

例 11:

```
#include <stdbool.h>
bool flag;
bool initialized = false;

#include <stdint.h>
int32_t x;
int32_t y;
uint32_t z;
```

对于与这些转换后的变量的大小相关的代码，需加以检查并相应地作出修改。

PIC18F 到 PIC24F 系列

for 循环初始化

自版本 1.xx 起，MPLAB XC8 默认支持 for 循环初始化。
对于版本 2.xx，C90 和 C99 模式下均支持（例 12）：

例 12:

```
#include <stdint.h>

for(uint8_t i = 0; i < 10; i++)
{
    // some code
}
```

目前，MPLAB XC16（v1.36B）默认不支持 for 循环初始化。编译预计会失败，具体错误消息如下：

```
error: 'for' loop initial declarations are only allowed in C99 mode
```

```
note: use option -std=c99 or -std=gnu99 to compile your code
```

开发人员应修改代码以在 for 循环外初始化索引（例 13）：

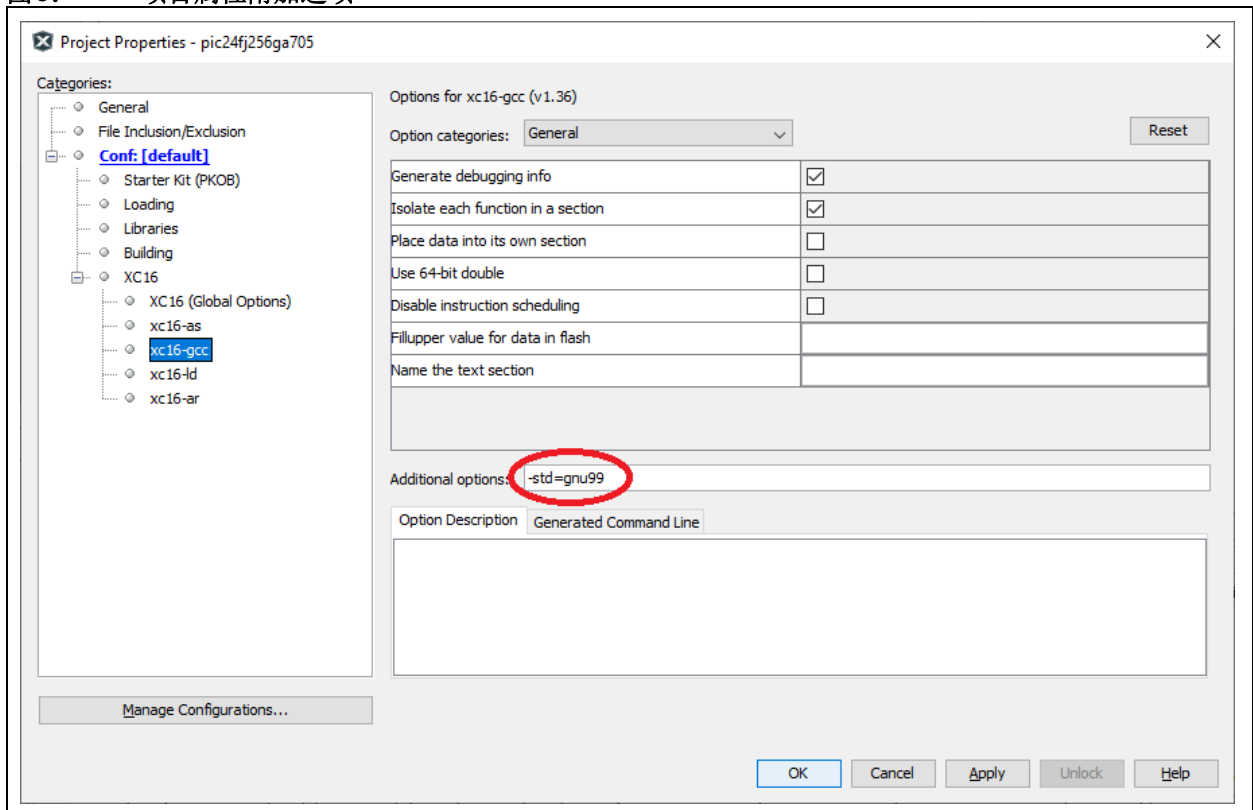
例 13:

```
#include <stdint.h>

uint8_t i;
for(i = 0; i < 10; i++)
{
    // some code
}
```

开发人员还可遵循编译日志中关于使用选项 `-std=c99` 或 `-std=gnu99`（可在 MPLAB X Project Properties 中指定，如图 3 所示）的建议。关于 `c99` 和 `gnu99` 模式之间的差异不在本文档的讨论范围之内。这里我们举例说明二者之间的其中一项差异：这两种模式均可识别关键字 `asm`，但 `c99` 模式下的解析器不会将“asm”识别为关键字。

图 3: 项目属性附加选项



总结

本文介绍了将PIC18F应用程序移植到PIC24F器件时的软件和工具注意事项。列出所有可能的软件移植问题是不切实际的，本文档讨论了一些常见的差异。

PIC18F 到 PIC24F 系列

注:

请注意以下有关 Microchip 器件代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信：在正常使用的情况下，Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前，仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知，所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿与那些注重代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。

代码保护功能处于持续发展中。Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了《数字千年版权法案 (Digital Millennium Copyright Act)》。如果这种行为导致他人在未经授权的情况下，能访问您的软件或其他受版权保护的成果，您有权依据该法案提起诉讼，从而制止这种行为。

提供本文档的中文版本仅为了便于理解。请勿忽视文档中包含的英文部分，因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

本出版物中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。Microchip 对因这些信息及使用这些信息而引起的后果不承担任何责任。如果将 Microchip 器件用于生命维持和 / 或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切伤害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任，并加以赔偿。除非另外声明，在 Microchip 知识产权保护下，不得暗或以其他方式转让任何许可证。

有关 Microchip 质量管理体系的更多信息，请访问 www.microchip.com/quality。

商标

Microchip 的名称和徽标组合、Microchip 徽标、Adaptec、AnyRate、AVR、AVR 徽标、AVR Freaks、BesTime、BitCloud、chipKIT、chipKIT 徽标、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、maXStylus、maXTouch、MediaLB、megaAVR、Microsemi、Microsemi 徽标、MOST、MOST 徽标、MPLAB、OptoLyzer、PacTime、PIC、picoPower、PICSTART、PIC32 徽标、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SST 徽标、SuperFlash、Symmetricom、SyncServer、Tachyon、TempTrackr、TimeSource、tinyAVR、UNI/O、Vectron 及 XMEGA 均为 Microchip Technology Incorporated 在美国和其他国家或地区的注册商标。

APT、ClockWorks、The Embedded Control Solutions Company、EtherSynch、FlashTec、Hyper Speed Control、HyperLight Load、IntelliMOS、Liberio、motorBench、mTouch、Powermite 3、Precision Edge、ProASIC、ProASIC Plus、ProASIC Plus 徽标、Quiet-Wire、SmartFusion、SyncWorld、Temux、TimeCesium、TimeHub、TimePictra、TimeProvider、Vite、WinPath 和 ZL 均为 Microchip Technology Incorporated 在美国的注册商标。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、BlueSky、BodyCom、CodeGuard、CryptoAuthentication、CryptoAutomotive、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、EtherGREEN、In-Circuit Serial Programming、ICSP、INICnet、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNet 徽标、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICTail、PowerSmart、PureSilicon、QMatrix、REAL ICE、Ripple Blocker、SAM-ICE、Serial Quad I/O、SMART-I.S.、SQI、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、ViewSpan、WiperLock、Wireless DNA 和 ZENA 均为 Microchip Technology Incorporated 在美国和其他国家或地区的商标。

SQTP 为 Microchip Technology Incorporated 在美国的服务标记。

Adaptec 徽标、Frequency on Demand、Silicon Storage Technology 和 Symmcom 均为 Microchip Technology Inc. 在除美国外的国家或地区的注册商标。

GestIC 为 Microchip Technology Inc. 的子公司 Microchip Technology Germany II GmbH & Co. KG 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2020, Microchip Technology Incorporated 版权所有。

ISBN: 978-1-5224-5815-9



全球销售及服务中心

美洲

公司总部 **Corporate Office**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 1-480-792-7200
Fax: 1-480-792-7277

技术支持:
<http://www.microchip.com/support>

网址: www.microchip.com

亚特兰大 **Atlanta**

Duluth, GA
Tel: 1-678-957-9614
Fax: 1-678-957-1455

奥斯汀 **Austin, TX**

Tel: 1-512-257-3370

波士顿 **Boston**

Westborough, MA
Tel: 1-774-760-0087
Fax: 1-774-760-0088

芝加哥 **Chicago**

Itasca, IL
Tel: 1-630-285-0071
Fax: 1-630-285-0075

达拉斯 **Dallas**

Addison, TX
Tel: 1-972-818-7423
Fax: 1-972-818-2924

底特律 **Detroit**

Novi, MI
Tel: 1-248-848-4000

休斯敦 **Houston, TX**

Tel: 1-281-894-5983

印第安纳波利斯 **Indianapolis**

Noblesville, IN
Tel: 1-317-773-8323
Fax: 1-317-773-5453
Tel: 1-317-536-2380

洛杉矶 **Los Angeles**

Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608
Tel: 1-951-273-7800

罗利 **Raleigh, NC**

Tel: 1-919-844-7510

纽约 **New York, NY**

Tel: 1-631-435-6000

圣何塞 **San Jose, CA**

Tel: 1-408-735-9110
Tel: 1-408-436-4270

加拿大多伦多 **Toronto**

Tel: 1-905-695-1980
Fax: 1-905-695-2078

亚太地区

中国 - 北京
Tel: 86-10-8569-7000

中国 - 成都
Tel: 86-28-8665-5511

中国 - 重庆
Tel: 86-23-8980-9588

中国 - 东莞
Tel: 86-769-8702-9880

中国 - 广州
Tel: 86-20-8755-8029

中国 - 杭州
Tel: 86-571-8792-8115

中国 - 南京
Tel: 86-25-8473-2460

中国 - 青岛
Tel: 86-532-8502-7355

中国 - 上海
Tel: 86-21-3326-8000

中国 - 沈阳
Tel: 86-24-2334-2829

中国 - 深圳
Tel: 86-755-8864-2200

中国 - 苏州
Tel: 86-186-6233-1526

中国 - 武汉
Tel: 86-27-5980-5300

中国 - 西安
Tel: 86-29-8833-7252

中国 - 厦门
Tel: 86-592-238-8138

中国 - 香港特别行政区
Tel: 852-2943-5100

中国 - 珠海
Tel: 86-756-321-0040

台湾地区 - 高雄
Tel: 886-7-213-7830

台湾地区 - 台北
Tel: 886-2-2508-8600

台湾地区 - 新竹
Tel: 886-3-577-8366

亚太地区

澳大利亚 **Australia - Sydney**
Tel: 61-2-9868-6733

印度 **India - Bangalore**
Tel: 91-80-3090-4444

印度 **India - New Delhi**
Tel: 91-11-4160-8631

印度 **India - Pune**
Tel: 91-20-4121-0141

日本 **Japan - Osaka**
Tel: 81-6-6152-7160

日本 **Japan - Tokyo**
Tel: 81-3-6880-3770

韩国 **Korea - Daegu**
Tel: 82-53-744-4301

韩国 **Korea - Seoul**
Tel: 82-2-554-7200

马来西亚 **Malaysia - Kuala Lumpur**
Tel: 60-3-7651-7906

马来西亚 **Malaysia - Penang**
Tel: 60-4-227-8870

菲律宾 **Philippines - Manila**
Tel: 63-2-634-9065

新加坡 **Singapore**
Tel: 65-6334-8870

泰国 **Thailand - Bangkok**
Tel: 66-2-694-1351

越南 **Vietnam - Ho Chi Minh**
Tel: 84-28-5448-2100

欧洲

奥地利 **Austria - Wels**
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

丹麦 **Denmark - Copenhagen**
Tel: 45-4485-5910
Fax: 45-4485-2829

芬兰 **Finland - Espoo**
Tel: 358-9-4520-820

法国 **France - Paris**
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

德国 **Germany - Garching**
Tel: 49-8931-9700

德国 **Germany - Haan**
Tel: 49-2129-3766400

德国 **Germany - Heilbronn**
Tel: 49-7131-72400

德国 **Germany - Karlsruhe**
Tel: 49-721-625370

德国 **Germany - Munich**
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

德国 **Germany - Rosenheim**
Tel: 49-8031-354-560

以色列 **Israel - Ra'anana**
Tel: 972-9-744-7705

意大利 **Italy - Milan**
Tel: 39-0331-742611
Fax: 39-0331-466781

意大利 **Italy - Padova**
Tel: 39-049-7625286

荷兰 **Netherlands - Drunen**
Tel: 31-416-690399
Fax: 31-416-690340

挪威 **Norway - Trondheim**
Tel: 47-7288-4388

波兰 **Poland - Warsaw**
Tel: 48-22-3325737

罗马尼亚 **Romania - Bucharest**
Tel: 40-21-407-87-50

西班牙 **Spain - Madrid**
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

瑞典 **Sweden - Gothenberg**
Tel: 46-31-704-60-40

瑞典 **Sweden - Stockholm**
Tel: 46-8-5090-4654

英国 **UK - Wokingham**
Tel: 44-118-921-5800
Fax: 44-118-921-5820