

PIC32 自举程序

作者: *Ganapathi Ramachandra*
Microchip Technology Inc.

简介

PIC32器件的自举程序用于升级目标器件上的固件，无需编程器或调试器。

自举程序包括以下应用程序：

- 自举程序固件，将烧写到目标 PIC32 器件中
- 演示应用程序，可通过自举程序下载到目标 PIC32 器件中
- PC 主机应用程序，可与 PIC32 器件内运行的自举程序固件进行通信。此应用程序用于执行擦除和编程操作。

先决条件

运行自举应用程序的先决条件如下：

- 一台安装了 MPLAB® IDE v8.60 或更高版本的 PC
- PIC32 入门工具包或带 PIC32 接插模块 (PIM) 的 Explorer 16 开发板
- USB 一串行端口转换器 (如果 PC 上没有 COM 端口)
- 传统编程工具，用于最初将自举程序固件写入 PIC32 器件 (如 MPLAB® REAL ICE™ 在线仿真系统或 ICD 3 在线调试器)。入门工具包不需要任何编程工具。

用户应该熟悉以下概念：

- 器件配置寄存器
- 编译和烧写 PIC32 器件
- PIC32 链接器脚本

概念

自举应用程序放置在 PIC32 闪存存储器的专用段，具体位置取决于以下条件：

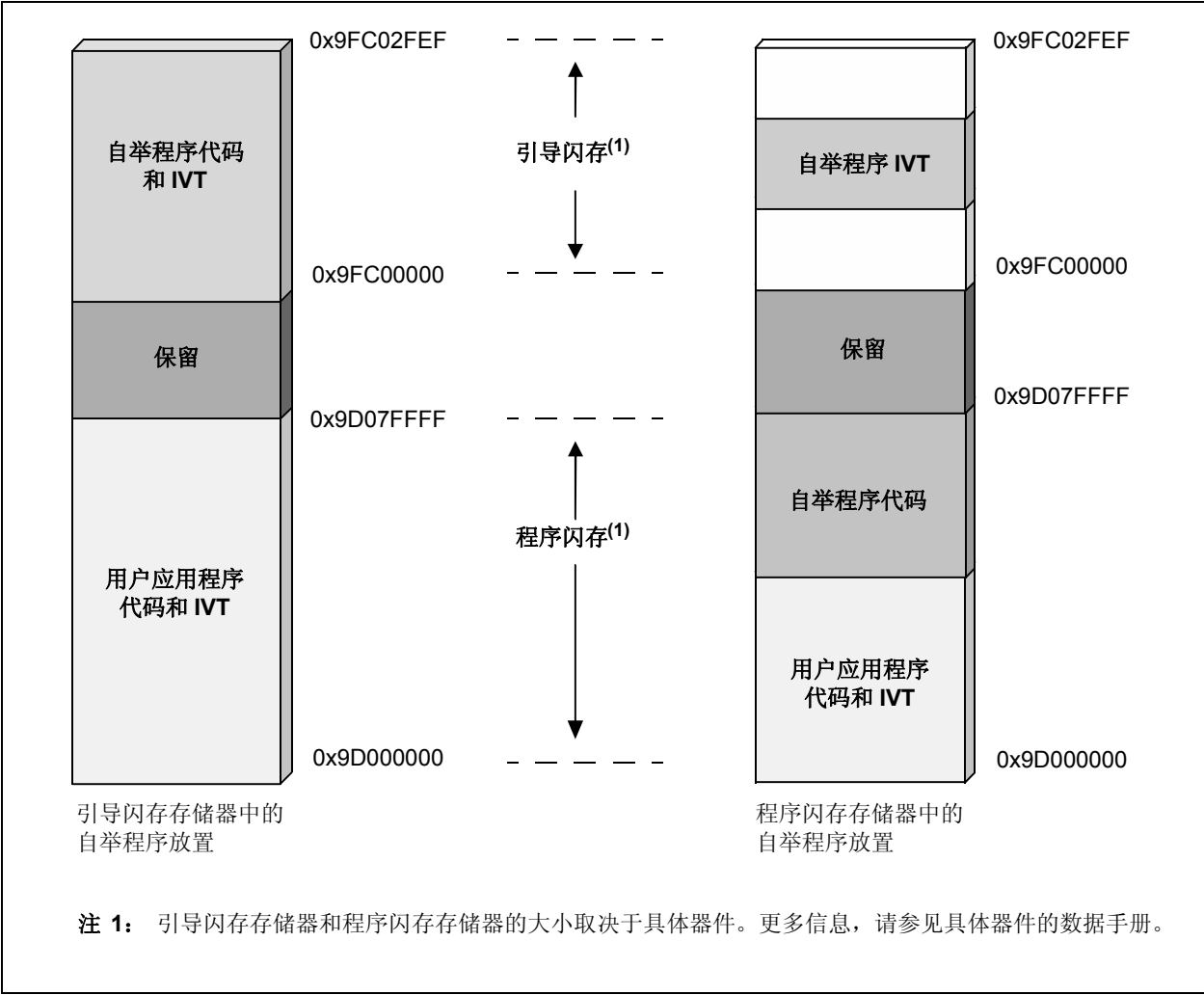
- 如果自举应用程序较小，可放置在 PIC32 的引导闪存存储器中。将自举应用程序放到引导闪存存储器中，可为用户应用程序提供完整的程序闪存存储器。
- 如果自举应用程序较大，则必须放置在程序闪存存储器中。用户应用程序可占用程序闪存存储器的剩余区域。在这种情况下，可利用 PIC32 引导闪存存储器放置自举应用程序的中断向量表 (Interrupt Vector Table, IVT)。图 1 显示了自举程序 and 用户应用程序在 PIC32 闪存存储器中的放置。

自举程序代码在器件复位时开始执行。如果没有有效的应用程序，或者存在外部触发信号，自举应用程序将进入固件升级模式。

如果没有外部触发信号，但是存在有效的用户程序，则自举程序代码会将控制权转移给用户应用程序并开始执行用户应用程序。外部触发器是一个按钮开关，用户只能在器件启动期间按下此开关。自举应用程序通过读取用户应用程序的复位地址内容来测试其有效性。如果发现此内容已擦除，则自举应用程序会保持在固件升级模式。

在固件升级模式下，自举应用程序将持续检查是否有命令来自 PC 主机应用程序。当接收到来自 PC 主机应用程序的命令时，自举程序将执行相应的操作，如擦除或烧写程序闪存存储器等。图 3 中的流程图显示了自举应用程序的操作。

图 1： 自举程序放置



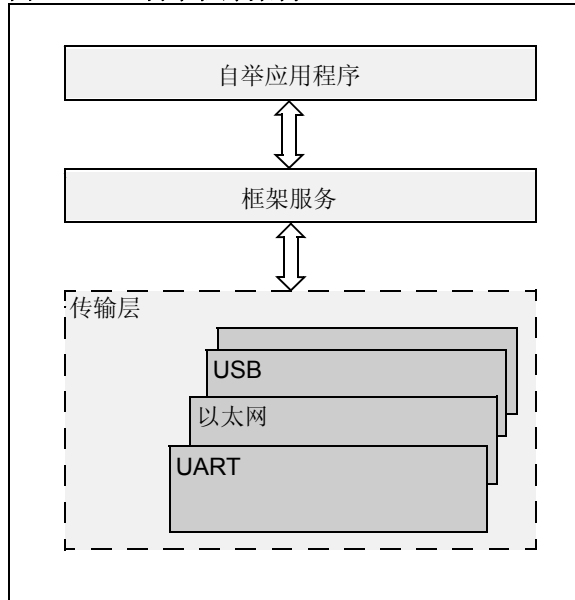
实现概述

自举应用程序使用框架实现。自举程序固件通过预定义的通信协议与 PC 主机应用程序进行通信。自举程序框架提供应用程序编程接口（Application Programming Interface, API）函数处理来自 PC 应用程序的协议相关帧。有关通信协议的更多信息，请参见附录 B：“自举程序通信协议”。

框架

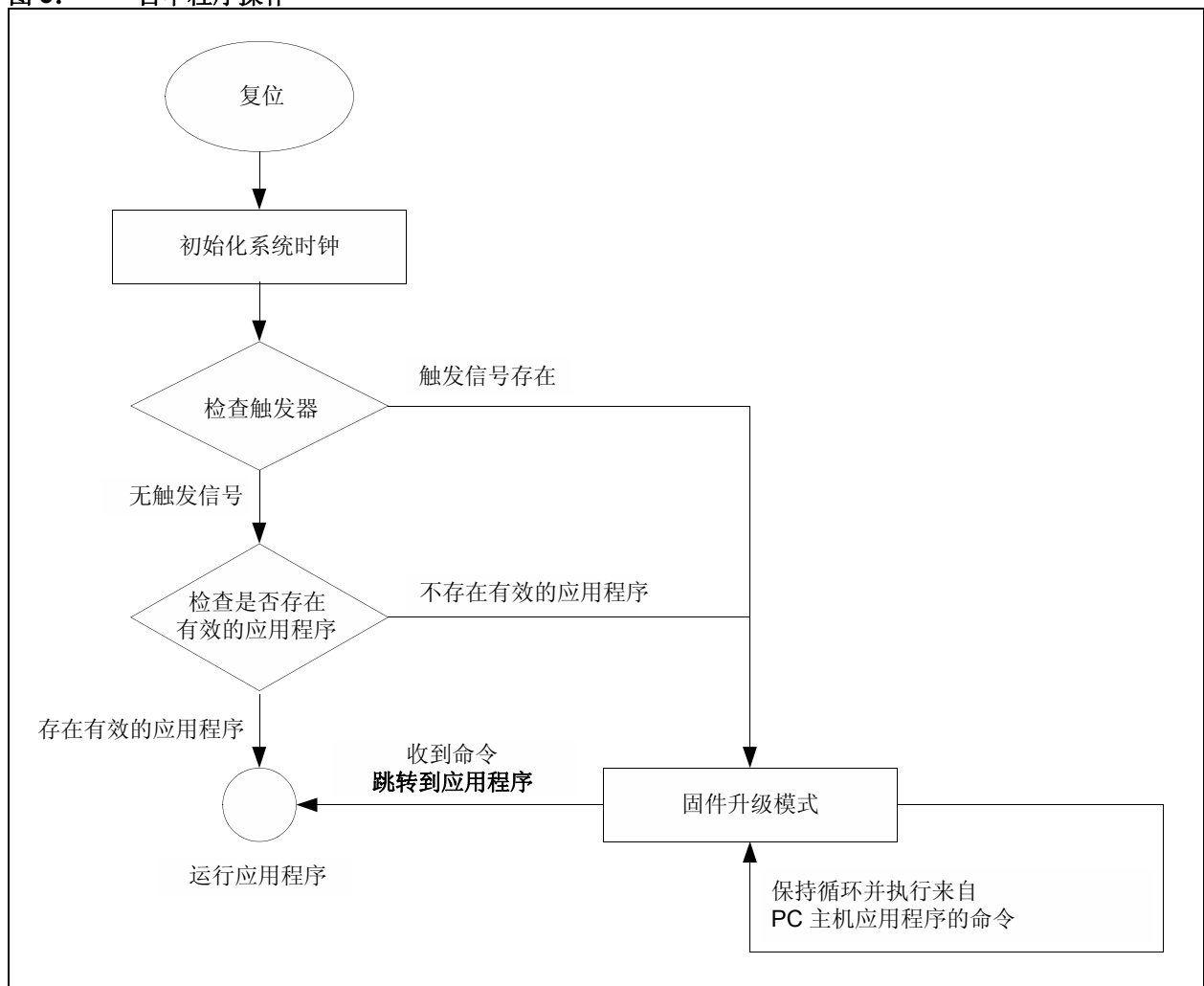
自举程序框架提供多个 API 函数，自举应用程序和传输层可调用这些 API 函数。用户借助自举程序框架可轻松修改自举应用程序以适合不同的要求。自举程序架构如图 2 所示。

图 2: 自举程序架构



Bootloader.c 文件包含自举应用程序代码。此文件包含自举程序功能，图 3 以流程图的形式说明了此文件。

图 3: 自举程序操作



Framework.c 文件包含框架函数。框架处理通信协议帧并执行接收自 PC 主机应用程序的命令。

传输层文件 Uart.c 和 Usb_HID_tasks.c 包含将原始字节发送到 PC 主机应用程序以及从 PC 主机应用程序接收原始字节的功能。

表 1 列出了框架提供的 API 函数。

表 1: 框架 API 说明

API	说明
void FrameWorkTask(void)	<p>如果存在来自 PC 主机应用程序的有效帧，此函数将执行命令。必须定期在自举应用程序任务中调用该函数。</p> <p>输入参数: 无</p> <p>返回值: 无</p>
void BuildRxFrame(UINT8 *RxData, INT16 RxLen)	<p>当传输层接收到来自 PC 主机应用程序的数据时，将调用此函数。</p> <p>输入参数: *RxData: 指向接收到的数据缓冲区的指针 RxLen: 接收到的数据字节的长度</p> <p>返回值: 无</p>
UINT GetTransmitFrame(UINT8* TxData)	<p>如果存在来自框架的有效响应帧，将返回一个非零值。传输层调用此函数以检查是否存在要发送到 PC 主机应用程序的帧。</p> <p>输入参数: *TxData: 指向将承载响应帧的数据缓冲区的指针</p> <p>返回值: 响应帧的长度。零表示无响应帧</p>
BOOL ExitFirmwareUpgradeMode(void)	<p>框架指示自举应用程序退出固件升级模式并执行用户应用程序。当 PC 主机应用程序发出运行应用程序命令时，可能出现这种情况。</p> <p>输入参数: 无</p> <p>返回值: 如果值为真，则退出固件升级模式</p>

基本设置

如果选择 UART 模块作为通信端口，自举应用程序将提供可更改波特率的选项。默认情况下，自举程序波特率设置为 115200。要更改波特率，请将 UART.h 文件中的 DEFAULT_BAUDRATE 宏的值更改为所需值。

默认情况下，自举程序软件版本设置为 1.0。

Bootloader.h 文件中定义的 MAJOR_VERSION 和 MINOR_VERSION 宏可用于更改自举应用程序的版本。用户必须确保在执行这些更改后重新编译整个项目。PC 应用程序可读取自举程序版本，用户可使用 PC 应用程序对自举程序固件进行版本控制。

器件配置位

PIC32 器件系列具有可编程非易失性器件配置寄存器，该寄存器用于定义器件的行为。有关器件配置寄存器的更多信息，请参见具体器件的数据手册。

注： 器件配置寄存器位于引导闪存存储器段；因此，这些寄存器必须与自举程序代码一起编程。对应用程序进行编程时，自举程序不会修改器件配置寄存器。因此，建议应用程序和自举程序代码使用相同的配置设置。

可通过 #pragma config 编译器指令配置器件配置寄存器。图 4 显示了 bootloader.c 文件中用于配置器件配置寄存器的代码段。

图 4： 代码中的配置设置

```

BootLoader.c
*****/
#include "Include\GenericTypeDefs.h"
#include "Include\HardwareProfile\HardwareProfile.h"
#include "Include\Bootloader.h"
#include "Include\FrameWork\Framework.h"
#include <stdlib.h>
#include <plib.h>

// Configuring the Device Configuration Registers
// 80Mhz Core/Periph, Pri Osc w/PLL, Write protect Boot Flash
#pragma config UPLLEN = ON // USB PLL Enabled
#pragma config UPLLIDIV = DIV_2 // USB PLL Input Divider
#pragma config FPLLMUL = MUL_20, FPLLIDIV = DIV_2, FPLLODIV = DIV_1, FWDTEN = OFF
#pragma config POSCMOD = HS, FNOSC = PRIPLL, FPBDIV = DIV_1
#pragma config ICSEL = ICS_PGx2, BWP = OFF

```

表 2 列出了自举应用程序中使用的器件配置位及其设置。此表中未包括的其余器件配置位保留其默认值。

表 2: 自举程序配置设置

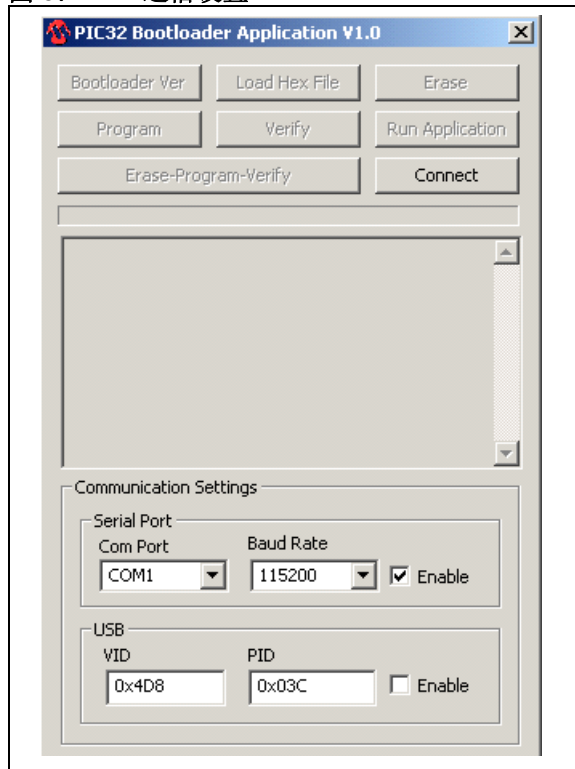
配置位	设置
USB PLL 使能位 (UPLLEN)	使能。
PLL 输入除法器位 (UPLLIDIV)	设置为 2x 除法器。
PLL 乘法器位 (FPLLMUL)	设置为 20x 乘法器。
PLL 输入除法器位 (FPLLIDIV)	设置为 2x 除法器。
PLL 的默认后分频比位 (FPLLIDIV)	PLL 输出除以 1。
看门狗定时器使能位 (FWDTEN)	禁止。
主振荡器配置位 (POSCMOD)	高速振荡器模式。
振荡器选择位 (FNOSC)	具有 PLL 模式的主振荡器。
外设总线时钟除数默认值位 (FPBDIV)	系统时钟除以 1。
在线仿真器/调试器通信通道选择位 (ICESEL)	设置为 1 (在线仿真器使用 EMUC2/EMUD2 引脚; 在线调试器使用 PGC2/PGD2 引脚)。

使用自举应用程序

按照以下步骤使用自举应用程序：

1. 参考相关的 Readme.html 文件以选择适当的工作区，然后在 MPLAB IDE 中选择并打开工作区。工作区位于 Bootloader\Firmware\Bootloader 文件夹中。
2. 从 IDE 菜单中选择相应的器件部件编号，然后编译自举程序项目。使用编程工具（如 MPLAB ICD 3、REAL ICE 或 PIC32 入门工具包）将自举程序烧写到器件中。
3. 断开编程器与 PIC32 器件的连接。
4. 运行 PC 应用程序文件 PIC32UBL.exe，该文件位于 BootLoader\PC Application 文件夹中。
5. 根据所选工作区，在通信设置中选择 COM 端口或 USB，如图 5 所示。

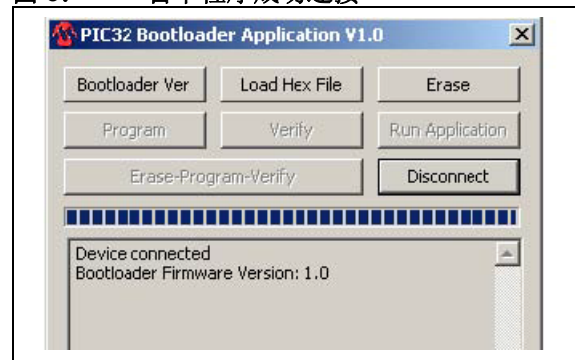
图 5： 通信设置



6. 复位 PIC32 器件并将自举程序设置为固件升级模式。请参考相关的 Readme.html 文件以了解将自举程序设置为固件升级模式的步骤。

7. 如果器件成功连接，将显示确认消息，如图 6 所示。

图 6： 自举程序成功连接



8. 单击 **Load Hex File**（加载十六进制文件）按钮，然后浏览到应用程序映像所在的位置。

应用程序映像必须是 Intel Hex 格式。选择并加载相应的应用程序映像文件。仅当成功加载十六进制文件后，**Program**（编程）和 **Erase-Program-Verify**（擦除 - 编程 - 验证）按钮才会启用。PC 主机应用程序控制台会显示消息“Hex file loaded successfully”（十六进制文件加载成功），指示已成功加载十六进制文件，如图 7 所示。

图 7： PC 应用程序消息



9. 单击 **Erase**（擦除）按钮以擦除程序闪存存储器。控制台窗口中显示消息“Flash Erased”（闪存已擦除），指示已成功擦除闪存，如图 7 所示。

10. 单击 **Program** 按钮对闪存进行烧写。PC 主机应用程序开始将应用程序映像写入器件。此步骤可能需要很长时间，具体取决于所选波特率和所选应用程序映像的大小。完成后，控制台窗口中将显示消息 “Programming completed”（编程已完成），如图 7 所示。
11. 使用 16 位循环冗余校验（Cyclic Redundancy Check, CRC）验证烧写到程序闪存存储器中的数据完整性。PC 主机应用程序计算出的 CRC 与器件计算出的 CRC 必须匹配，才能认为验证通过。CRC 匹配后，控制台窗口中将显示消息 “Verification successful”（验证成功），如图 7 所示。
12. 单击 **Run Application**（运行应用程序）按钮运行用户应用程序固件。完成此步骤之后，自举程序将控制权转移给用户应用程序，然后自举程序停止运行。要再次调用自举程序，请执行步骤 6。

成功完成所有这些步骤之后，PC 应用程序控制台窗口应类似于图 7。通过单击 **Erase-Program-Verify** 按钮可一步执行擦除、编程和验证。

结论

本应用笔记介绍了 PIC32 自举程序、自举程序存储器映射、自举程序框架 API 调用的概念，以及自举程序 PC 应用程序的使用方法。

附录 B：“自举程序通信协议” 将介绍自举程序使用的通信协议和命令。

附录 C：“移动应用程序映像时的注意事项” 将介绍将应用程序映射到程序闪存内不同区域的分步方法。

参考资料

可从 Microchip Technology Inc. 获取以下资料

以下文档提供了有关链接器脚本文件中使用的术语的信息：

- 《用于 PIC32 MCU 的 MPLAB® 汇编器、链接器和实用程序用户指南》（DS51833A_CN）
- 《MPLAB® C32 C 编译器用户指南》（DS51686A_CN）

以下文档提供了有关 PIC32 器件及其外设的信息：

- 《PIC32MX3XX/4XX 系列数据手册》（DS61143G_CN）
- 《PIC32MX5XX/6XX/7XX 系列数据手册》（DS61156F_CN）

附录 A： 源代码

软件许可协议

Microchip Technology Incorporated（以下简称“本公司”）在此提供的软件旨在向本公司客户提供专门用于本公司生产的产品的软件。

本软件为本公司和/或其供应商所有，并受到适用的版权法保护。版权所有。使用时违反前述约束的用户可能会依法受到刑事制裁，并可能由于违背本许可的条款和条件而承担民事责任。

本软件是按“现状”提供的。不附有任何形式的保证，无论是明示的、暗示的或法定的，包括（但不限于）有关适销性和特定用途的暗示保证。对于在任何情况下，因任何原因造成的特殊的、偶然的或间接的损害，本公司概不负责。

本应用笔记中提及的所有软件都以单个 WinZip 归档文件的形式提供。可从 Microchip 公司网站下载此文件：

www.microchip.com

附录 B： 自举程序通信协议

PC 主机应用程序使用通信协议与自举程序固件进行交互。PC 主机应用程序作为主机，向自举程序固件发送命令以执行特定操作。

帧格式

通信协议遵循例 1 所示的帧格式。帧格式双向（即，从主机应用程序到自举程序以及从自举程序到主机应用程序）相同。

例 1： 帧格式

[<SOH>...]<SOH>[<DATA>...]<CRCL><CRCH><EOT>
其中：
<...>——表示一个字节
[...]——表示可选或可变数量的字节

帧以一个控制字符（帧头开始（Start of Header, SOH））开始，以另一个控制字符（传输结束（End of Transmission, EOT））结束。帧的完整性由两个 CRC-16 字节保护，分别表示为 CRCL（低字节）和 CRCH（高字节）。

控制字符

表 3： 控制字符说明

控制	十六进制值	说明
<SOH>	0x01	标识帧开始
<EOT>	0x04	标识帧结束
<DLE>	0x10	数据链路转义符

数据字段中的某些字节可能与控制字符 SOH 和 EOT 相似。数据链路转义（Data Link Escape, DLE）字符用于转义此类会被解释为控制字符的字节。自举程序总是接受 <DLE> 之后的字节作为数据，并且总是在发送任何控制字符之前先发送一个 <DLE>。

命令

PC 主机应用程序可将表 4 中列出的命令发送到自举程序。数据字段中的第一个字节用于承载命令。

表 4： 命令说明

十六进制的命令值	说明
0x01	读取自举程序版本信息
0x02	擦除闪存
0x03	编程闪存
0x04	读取 CRC
0x05	跳转到应用程序

读取自举程序版本信息

PC 主机应用程序向自举程序请求版本信息，如例 2 所示。

例 2： 请求

[<SOH>...]<SOH>[<0x01>]<CRCL><CRCH><EOT>
--

自举程序用两个字节响应 PC 对版本信息的请求，如例 3 所示。

例 3： 响应

[<SOH>...]<SOH><0x01><MAJOR_VER><MINOR_VER><CRCL><CRCH><EOT>
其中：
MAJOR_VER——自举程序的主要版本
MINOR_VER——自举程序的次要版本

擦除闪存

接收到来自 PC 主机应用程序的擦除闪存命令后，自举程序将擦除分配给用户应用程序的那部分程序闪存。

PC 主机应用程序发送到自举程序的请求帧如例 4 所示。

例 4： 请求

[<SOH>...]<SOH><0x02><CRCL><CRCH><EOT>
--

自举程序发送到 PC 主机应用程序的响应帧如例 5 所示。

例 5： 响应

[<SOH>...]<SOH><0x02><CRCL><CRCH><EOT>
--

编程闪存

PC 主机应用程序发送一个或多个 Intel Hex 格式的十六进制记录以及编程闪存命令。MPLAB C32 编译器将生成 Intel Hex 格式的映像。Intel 十六进制文件中的每一行都表示一条十六进制记录。每条十六进制记录以一个冒号 (:) 开头并采用 ASCII 格式。PC 主机应用程序会丢弃冒号并将其余数据从 ASCII 格式转换为十六进制格式，然后将数据发送到自举程序。自举程序从十六进制记录中提取目标地址和数据，并将数据写入程序闪存。

PC 主机应用程序发送到自举程序的请求帧如例 6 所示。

例 6: 请求

```
[<SOH>...]<SOH><0x03>[<HEX_RECORD>...]<CRCL>
<CRCH><EOT>
```

其中:

HEX_RECORD——十六进制格式的 Intel Hex 记录

自举程序发送到 PC 主机应用程序的响应如例 7 所示。

例 7: 响应

```
[<SOH>...]<SOH><0x03><CRCL><CRCH><EOT>
```

读取 CRC

读取 CRC 命令用于在编程后校验程序闪存的内容。

PC 主机应用程序发送到自举程序的请求帧如例 8 所示。

例 8: 请求

```
[<SOH>...]<SOH><0x04><ADRS_LB><ADRS_HB>
<ADRS_UB><ADRS_MB><NUMBYTES_LB>
<NUMBYTES_HB><NUMBYTES_UB><NUMBYTES_MB>
<CRCL><CRCH><EOT>
```

自举程序发送到 PC 主机应用程序的响应如例 9 所示。

例 9: 响应

```
[<SOH>...]<SOH><0x04><FLASH_CRCL><FLASH_CRCH>
<CRCL><CRCH><EOT>
```

ADRS_LB、ADRS_HB、ADRS_UB 和 ADRS_MB (如例 8 所示) 表示 CRC 计算开始处的 32 位闪存地址。

NUMBYTES_LB、NUMBYTES_HB、NUMBYTES_UB 和 NUMBYTES_MB (如例 8 所示) 表示需要计算其 CRC 的字节总数 (32 位格式)。

跳转到应用程序

PC 主机应用程序的**跳转到应用程序**命令用于命令自举程序执行应用程序。PC 主机应用程序发送到自举程序的请求帧如例 10 所示。

例 10: 请求

```
[<SOH>...]<SOH><0x05><CRCL><CRCH><EOT>
```

自举程序不会响应此命令，因为自举程序会立即退出固件升级模式并开始执行应用程序。

附录 C： 移动应用程序映像时的 注意事项

自举程序代码有三个宏，指定了用户应用程序的位置和复位地址。这些宏可在 `bootloader.h` 文件中找到。

`APP_FLASH_BASE_ADDRESS` 和

`APP_FLASH_END_ADDRESS` 宏指定了预留给用户应用程序的程序闪存的起始地址和结束地址。仅当目标地址处于上述地址范围内时，自举程序才执行擦除或编程操作。`USER_APP_RESET_ADDRESS` 宏指定了用户应用程序的复位地址，应用程序从该地址开始执行。当必须运行用户应用程序时，自举程序会转移到此地址。必须修改应用程序项目的链接器脚本文件以将应用程序映像移动到程序闪存内的不同段。

应用程序映像包含三部分：**C** 启动代码、IVT 和程序（文本和数据）。链接器脚本文件将应用程序的这些部分分别映射到 `kseg0_boot_mem`、`exception_mem` 和 `kseg0_program_mem` 存储器段。要更改应用程序映射，用户只需要在应用程序项目的链接器脚本文件中修改这三个存储器段。有关这些存储器段的更多信息，请参见《用于 PIC32 MCU 的 MPLAB[®] 汇编器、链接器和实用程序用户指南》（DS51833A_CN）和《MPLAB[®] C32 C 编译器用户指南》（DS51686A_CN）。

修改应用程序映射的步骤及示例如下：

1. 在程序闪存中选择存储区放置用户应用程序。用户应用程序的存储区不得与预留给自举程序的存储区重叠。用户应用程序的基址必须与 **4K** 闪存段的起始位置对齐。结束地址必须与 **4K** 闪存段的边界对齐。在此示例中，分别选择 `0x9D005000` 和 `0x9D07FFFF` 作为用户应用程序的基址和结束地址。
2. 将预留给应用程序的程序闪存地址范围通知给自举程序。对于此示例，可通过将 `APP_FLASH_BASE_ADDRESS` 和 `APP_FLASH_END_ADDRESS` 的值分别设置为 `0x9D005000` 和 `0x9D07FFFF` 来实现相同操作。

3. **查找链接器脚本文件：**从 MPLAB C32 编译器的安装路径选择相应的 `procdefs.ld` 链接器头文件。例如，对于 PIC32MX795F512L 器件，从以下路径复制链接器头文件：`C:\Program Files\Microchip\MPLABC32\pic32mx\lib\proc\32MX795F512L`（或者从 MPLAB C32 编译器的安装路径复制链接器头文件）。将链接器文件复制到用户应用程序项目文件夹的根目录下。
4. **映射 IVT：**IVT 的基址必须与 **4K** 闪存段的起始位置对齐。如果 IVT 未与 **4K** 闪存段的起始位置对齐，PIC32 上的中断可能无法起作用。在此示例中，IVT 将放置到预留给应用程序的程序闪存的最后一个 **4K** 闪存段。即，IVT 将映射到 `0x9D07F000` 至 `0x9D07FFFF` 的范围。在 `procdefs.ld` 文件中将 `exception_mem` 的原始值更改为 `0x9D07F000` 以修改 IVT 映射。例 11 显示了 IVT 映射。

例 11： 设置异常存储器

```
exception_mem : ORIGIN = 0x9D07F000, LENGTH
                = 0x1000
```

`procdefs.ld` 文件中要更改的下一行是 `_ebase_address` 的值。`_ebase_address` 的值必须与 `exception_mem` 的原始值相同。因此，请按照例 12 所示更改该行。

例 12： 设置擦除寄存器

```
_ebase_address = 0x9D07F000
```

5. **映射 C 启动代码：**将 `kseg0` 引导存储器放在异常存储区的前面。`kseg0` 引导存储器包含 **C** 启动代码；因此，建议保持引导存储器长度的原始值。计算 `kseg0` 引导存储器的原始值，方法为从异常存储器的原始值（`0x9D07F000`）减去 `kseg0` 引导存储器的长度（`0x970`）。定义 `kseg0_boot_mem` 的行随即发生改变，如例 13 所示。

例 13： 设置 kseg0 引导存储器

```
kseg0_boot_mem : ORIGIN = 0x9D07E690,
                  LENGTH=0x970
```

6. **映射程序代码：**将其余存储区（从用户应用程序基址到 `kseg0` 引导存储器的起始位置）分配给 `kseg0` 程序存储器。将 `kseg0` 程序存储器起点设置为用户应用程序基址。可通过从 `kseg0` 引导存储器基址值（`0x9D07E690`）减去用户应用程序基址值（`0x9D005000`）来计算长度。定义 `kseg0_program_mem` 的行随即发生改变，如例 14 所示。

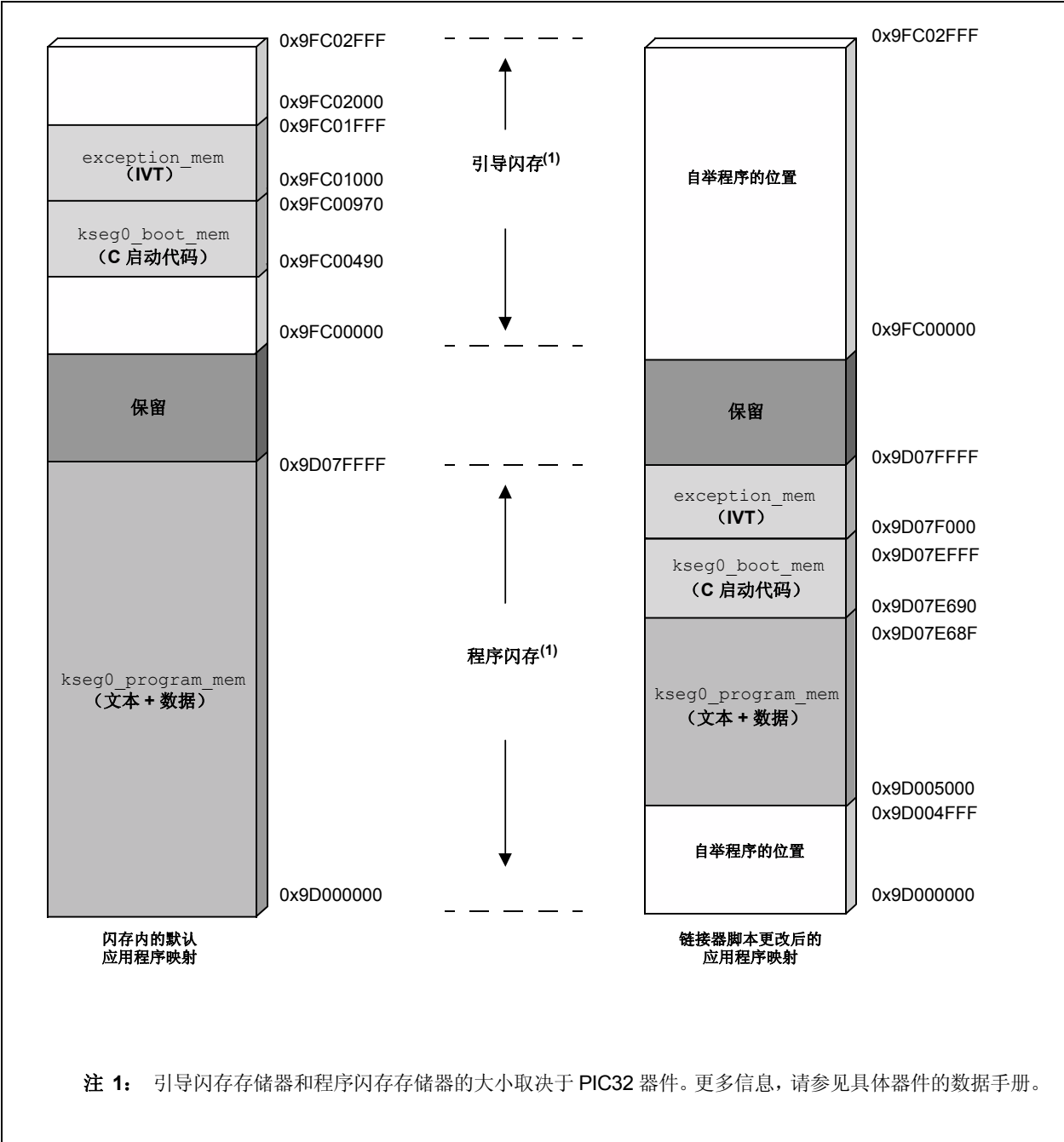
例 14： 设置 `kseg0` 程序存储器

```
kseg0_program_mem (rx) : ORIGIN=0x9D005000,
                          LENGTH=0x79690
```

7. **将应用程序的复位地址通知给自举程序：**存放 C 启动代码的 `kseg0_boot_mem` 的起始位置将成为应用程序的复位地址。通过将 `USER_APP_RESET_ADDRESS` 宏的值设置为 `kseg0_boot_mem` 的原始值，来将应用程序的复位地址通知给自举程序。在此示例中，必须将 `USER_APP_RESET_ADDRESS` 值设置为 `0x9D07E690`。自举程序会将控制权转移到此地址，以执行用户应用程序。
8. 使用所有已修改的宏值重新编译自举程序项目。
9. 使用应用程序项目根目录中包含的已修改链接器头文件重新编译应用程序项目。在链接过程中，**MPLAB IDE** 会自动包含项目根目录中的 `procdefs.ld` 文件。因此，无需将此链接器脚本文件明确添加到工作区中。
从 `elf32pic32mx.x` 文件中选择链接器脚本的主体，该文件与 **MPLAB C32** 工具位于相同位置（`C:\Program Files\Microchip\MPLABC32\pic32mx\lib\ldscripts`）。
`elf32pic32mx.x` 文件将代码段映射到 `procdefs.ld` 文件中定义的存储区。用户不需要对这个主链接器脚本文件进行任何修改。

图 8 显示了 PIC32 闪存存储器中应用程序映像的默认和已修改映射。

图 8： 链接器脚本更改之前和之后的应用程序映射



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《数字器件千年版权法案（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。在 Microchip 知识产权保护下，不得暗中以其他方式转让任何许可证。

商标

Microchip 的名称和徽标组合、Microchip 徽标、dsPIC、KEELOQ、KEELOQ 徽标、MPLAB、PIC、PICmicro、PICSTART、PIC³² 徽标、rfPIC 和 UNI/O 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

FilterLab、Hampshire、HI-TECH C、Linear Active Thermistor、MXDEV、MXLAB、SEEVAL 和 The Embedded Control Solutions Company 均为 Microchip Technology Inc. 在美国的注册商标。

Analog-for-the-Digital Age、Application Maestro、chipKIT、chipKIT 徽标、CodeGuard、dsPICDEM、dsPICDEM.net、dsPICworks、dsSPEAK、ECAN、ECONOMONITOR、FanSense、HI-TIDE、In-Circuit Serial Programming、ICSP、Mindi、MiWi、MPASM、MPLAB Certified 徽标、MPLIB、MPLINK、mTouch、Omniscient Code Generation、PICC、PICC-18、PICDEM、PICDEM.net、PICkit、PICtail、REAL ICE、rfLAB、Select Mode、Total Endurance、TSHARC、UniWinDriver、WiperLock 和 ZENA 均为 Microchip Technology Inc. 在美国和其他国家或地区的商标。

SQTP 是 Microchip Technology Inc. 在美国的服务标记。

在此提及的所有其他商标均为各持有公司所有。

© 2011, Microchip Technology Inc. 版权所有。

ISBN: 978-1-61341-725-6

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949:2002 ==

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2009 认证。Microchip 的 PIC[®] MCU 与 dsPIC[®] DSC、KEELOQ[®] 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品严格遵守公司的质量体系流程。此外，Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

全球销售及服务网点

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](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

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

克里夫兰 **Cleveland**

Independence, OH
Tel: 1-216-447-0464
Fax: 1-216-447-0643

达拉斯 **Dallas**

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

底特律 **Detroit**

Farmington Hills, MI
Tel: 1-248-538-2250
Fax: 1-248-538-2260

印第安纳波利斯

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

洛杉矶 **Los Angeles**

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

圣克拉拉 **Santa Clara**

Santa Clara, CA
Tel: 1-408-961-6444
Fax: 1-408-961-6445

加拿大多伦多 **Toronto**

Mississauga, Ontario,
Canada
Tel: 1-905-673-0699
Fax: 1-905-673-6509

亚太地区

亚太总部 **Asia Pacific Office**

Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

中国 - 北京

Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

中国 - 成都

Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

中国 - 重庆

Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

中国 - 杭州

Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

中国 - 香港特别行政区

Tel: 852-2401-1200
Fax: 852-2401-3431

中国 - 南京

Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

中国 - 青岛

Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

中国 - 上海

Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

中国 - 沈阳

Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

中国 - 深圳

Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

中国 - 武汉

Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

中国 - 西安

Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

中国 - 厦门

Tel: 86-592-238-8138
Fax: 86-592-238-8130

中国 - 珠海

Tel: 86-756-321-0040
Fax: 86-756-321-0049

亚太地区

台湾地区 - 高雄

Tel: 886-7-536-4818
Fax: 886-7-330-9305

台湾地区 - 台北

Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

台湾地区 - 新竹

Tel: 886-3-5778-366
Fax: 886-3-5770-955

澳大利亚 **Australia - Sydney**

Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

印度 **India - Bangalore**

Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

印度 **India - New Delhi**

Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

印度 **India - Pune**

Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

日本 **Japan - Yokohama**

Tel: 81-45-471- 6166
Fax: 81-45-471-6122

韩国 **Korea - Daegu**

Tel: 82-53-744-4301
Fax: 82-53-744-4302

韩国 **Korea - Seoul**

Tel: 82-2-554-7200
Fax: 82-2-558-5932 或
82-2-558-5934

马来西亚 **Malaysia - Kuala Lumpur**

Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

马来西亚 **Malaysia - Penang**

Tel: 60-4-227-8870
Fax: 60-4-227-4068

菲律宾 **Philippines - Manila**

Tel: 63-2-634-9065
Fax: 63-2-634-9069

新加坡 **Singapore**

Tel: 65-6334-8870
Fax: 65-6334-8850

泰国 **Thailand - Bangkok**

Tel: 66-2-694-1351
Fax: 66-2-694-1350

欧洲

奥地利 **Austria - Wels**

Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

丹麦 **Denmark-Copenhagen**

Tel: 45-4450-2828
Fax: 45-4485-2829

法国 **France - Paris**

Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

德国 **Germany - Munich**

Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

意大利 **Italy - Milan**

Tel: 39-0331-742611
Fax: 39-0331-466781

荷兰 **Netherlands - Druenen**

Tel: 31-416-690399
Fax: 31-416-690340

西班牙 **Spain - Madrid**

Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

英国 **UK - Wokingham**

Tel: 44-118-921-5869
Fax: 44-118-921-5820