

---

---

## 如何在 Cortex<sup>®</sup>-M0+单片机中自定义 ASFv3 SAM-BA 自举程序

---

---

### 简介

---

SAM Boot Assistant (SAM-BA<sup>™</sup>) 允许通过 USB 或 UART 主机进行系统内编程 (In-System Programming, ISP)，无需任何外部编程接口。通常情况下，在基于 ROM 的 SAM MCU 中，内部 ROM 都包含 SAM-BA 监视器固件，而所有基于 Cortex<sup>®</sup>-M0+ 的 MCU 则没有内部 ROM。

可以通过将 SAM-BA 监视器固件加载到闪存中来为基于 Cortex-M0+ 的器件提供 SAM-BA 支持。此 SAM-BA 监视器固件将充当自举程序，可接受从 SAM-BA 主机/GUI 发出的命令。SAM-BA GUI 将 SAM-BA 小程序发送到 SAM-BA 监视器，然后 SAM-BA 监视器将小程序固件加载到 SRAM 中。SAM-BA 小程序是在 SRAM 中运行的一个固件，用于处理 SAM-BA 监视器接收的 SAM-BA 命令。

SAM-BA 自举程序固件在 ASFv3 中作为独立示例提供。

#### 注：

要打开 SAMD21 SAM-BA 自举程序，用户可以执行以下操作从 Atmel Studio 中打开示例项目：依次选择 *File* (文件) > *New* (新建) > *Example project* (示例项目)，在 Device Family (器件系列) 中选择 SAMD21，然后选择 SAM0 SAM-BA 自举程序示例——*SAMD21 Xplained Pro* 项目。

本文档将重点介绍基于 ASFv3 的 SAM-BA 自举程序。

---

## 目录

---

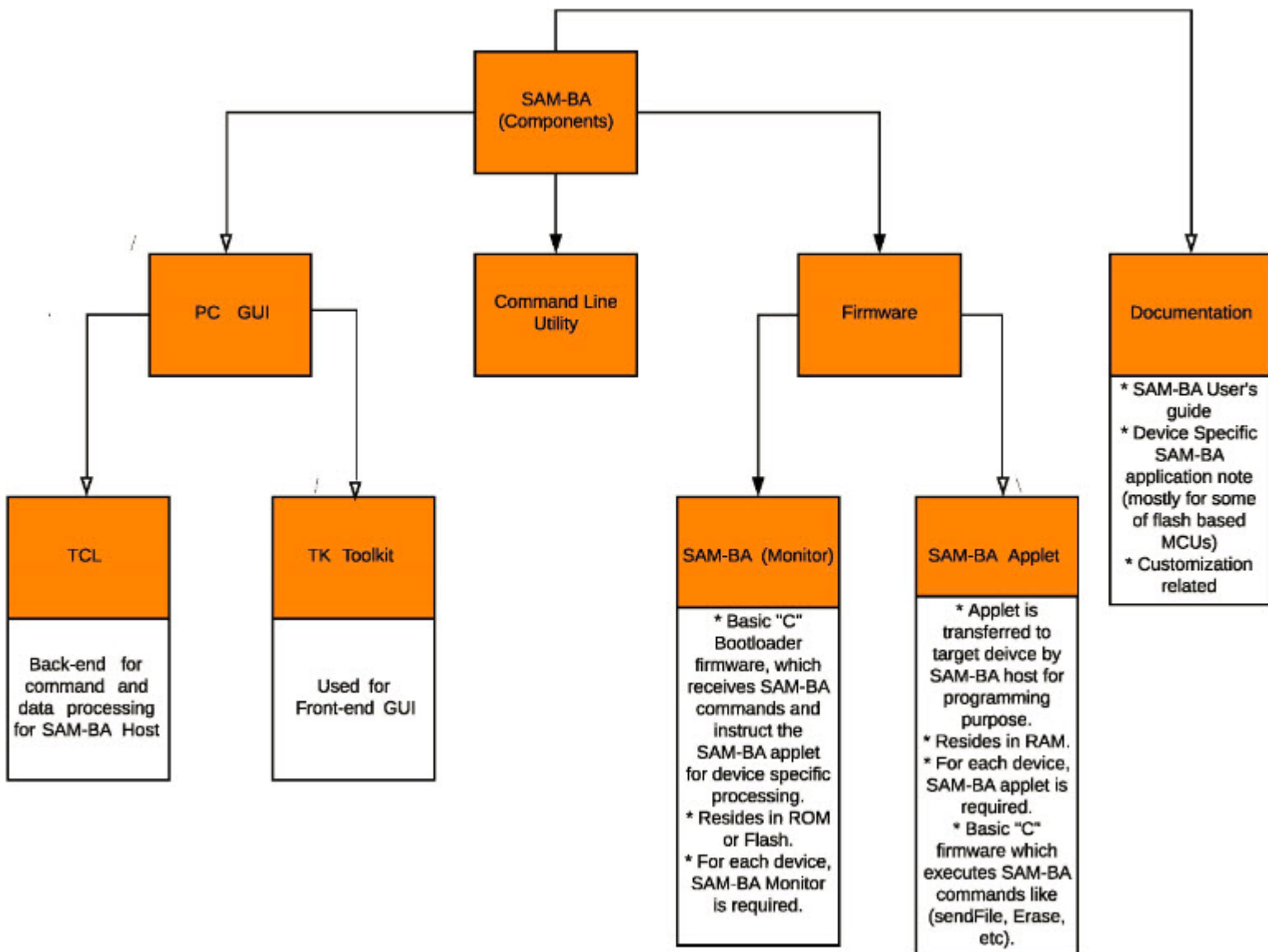
简介.....	1
1. 概念.....	3
2. 解决方案或实现.....	4
3. 其他相关资源.....	9
4. 常见问题.....	10
Microchip 网站.....	11
变更通知客户服务.....	11
客户支持.....	11
Microchip 器件代码保护功能.....	11
法律声明.....	12
商标.....	12
DNV 认证的质量管理体系.....	13
全球销售及服务网点.....	14

## 1. 概念

要使用 SAM-BA GUI 对基于 Cortex-M0+ 的 MCU 进行编程并写入闪存，必须满足以下条件：

- 必须将 SAM-BA 自举程序固件写入基于 Cortex-M0+ 的 MCU 的闪存。此固件将只能获取通过 SAM-BA GUI 发送的请求并处理受支持的命令。
- SAM-BA GUI 及其组件。
- TCL 脚本，用于对所选 MCU 进行基本验证并协助 SAM-BA GUI 来执行小程序命令。
- SAM-BA 小程序，用于执行从 SAM-BA GUI 发出的实际固件请求。

图 1-1. SAM-BA 框图



本文档将通过小程序和 TCL 脚本并结合一个示例来说明可对 SAM-BA 自举程序固件进行哪些自定义操作。

## 2. 解决方案或实现

在基于 Cortex-M0+ 的 SAM-BA 解决方案中，现有解决方案（SAM-BA 自举程序源代码、小程序等）仅适用于 Xplained Pro 板。以下章节将详细说明如何对用户板以及 Xplained Pro 板所用器件之外的其他器件使用 SAM-BA。

### 自定义 SAM-BA 以对 SAMD21E16B 使用（与 SAMD21J18A Xplained Pro）不同的串行通信接口（SERCOM）

在 Atmel Studio 7.0 中按以下步骤操作，以便为使用用户板的 SAMD21E16B 自定义基于 ASF SAMD21 Xplained Pro 的项目：

1. 在将 SAM0 SAM-BA 自举程序示例加载到 Atmel Studio 中后，用户需通过该 ASF 项目的 *Project (项目) > Properties (属性) > Device (器件)* 选项更改器件以选择 SAMD21E16B。
2. 用户需要通过 ASF 项目的 *Project > Properties > Toolchain (工具链) > ARM/GNU C Compiler (ARM/GNU C 编译器) > Symbols (符号)* 选项将现有定义 `__SAMD21J18A__` 替换为 `__SAMD21E16B__`。
3. ASF 可识别 Microchip 开发的评估板（例如 Xplained Pro 板）。ASF 会将使用 SAM 产品开发的评估板视为用户板。用户需要修改此项目的设置，以表明其正在开发基于用户板的项目。此项目的 *Project > Properties > Toolchain > ARM/GNU C Compiler > Symbols* 选项移除 `BOARD` 的现有定义（SAMD21 Xplained Pro 的定义），然后添加一个新定义 `BOARD=USER_BOARD`。
4. 在 `src\ASF\sam0\boards` 文件夹中添加一个名称为 `user_board.h` 的新文件。此文件类似于 `samd21_xplained_pro.h` 文件，其中包含此用户板的所有必备元件和引脚的定义。
5. 将 `board_init.c` 文件从 `ASF\sam0\boards\samd21_xplained_pro` 文件夹复制到 `ASF\sam0\boards` 文件夹。
6. 从此项目中移除 `ASF\sam0\boards\samd21_xplained_pro` 文件夹。
7. 必须更新链接器脚本，以便与 SAMD21E16B 匹配。通常，必须修改链接器脚本中的 RAM、闪存和默认堆栈大小选项，以便与器件匹配。
8. 可以根据项目需要在 ASF 自举程序中选择进行其他自定义，例如添加或移除 LED 等。

#### 注：

用户可以将 SAMD21J18A 的链接器脚本替换为 SAMD21E16B 的链接器脚本。对于已创建的默认项目（即 SAMD21J18A），链接器脚本位于 `src\ASF\sam0\utils\linker_scripts\samd21\gcc` 文件夹中。可以通过在 Atmel Studio 中为 SAMD21E16B 创建一个新项目来获取 SAMD21E16B 的链接器脚本。

### 自定义 ASF SAM-BA 自举程序配置

以下是 ASF SAM-BA 自举程序中的可用配置：

1. 时钟配置——要了解各种不同的时钟配置选项，请参见以下 [Microchip 知识文章](#)。
2. 自举程序配置——要了解各种不同的自举程序专用配置，请参见下表。用户不应修改自举程序配置文件 `conf_bootloader.h` 中的所有其他可用配置。

**表 2-1. SAM-BA 自举程序的可用配置**

序号	配置名称	配置值	备注
1	APP_START_ADDRESS	此值可根据自举程序固件大小进行设置。	例如 0x6000
2	BOOT_LOAD_PIN	用户定义的 GPIO 引脚，用于按钮访问	按下时将进入自举程序模式。 例如 PIN_PA11
3	BOOT_LED	用户定义的 GPIO 引脚，用于 LED 访问	仅在启用了 DEBUG_ENABLE 的情况下使用。 例如 PIN_PA10
4	DEBUG_ENABLE	用于显示通知，指出器件是处于自举程序模式还是应用模式。	例如 false
5	BOOT_USART_MODULE	用于进入自举程序使用的 SERCOM 实例	例如 SERCOM0
6	BOOT_USART_MUX_SETTINGS	用户定义的 SERCOM MUX 设置	例如 USART_RX_1_TX_0_XCK_1
7	BOOT_USART_PADx	用户定义的 SERCOM TX/RX 设置	例如 PINMUX_PA08C_SERCOM0_PAD0
8	BOOT_USART_PADy	用户定义的 SERCOM TX/RX 设置	例如 PINMUX_PA08C_SERCOM0_PAD1
9	BOOT_USART_GCLK_SOURCE	用户为 SAM-BA 自举程序定义的时钟源	例如 GCLK_GENERATOR_0
10	CONF_USBCDC_INTERFACE_SUPPORT	用户定义的自举程序选项	如果需要启用 USB 自举程序选项，则应定义此配置。

**自定义 ASF SAM-BA TCL 脚本**

1. 从 SAM-BA 安装文件夹中选择 `\\tcl_lib\\samd21_xplained_pro\\samd21_xplained_pro.tcl` 文件，然后更新 `devicesList` 变量，以便在受支持的器件列表中加入 SAMD21E16B。
2. 根据选择的器件以及 SAM-BA 自举程序占用的 SRAM，相应地更新小程序 `Addr` 和小程序 `MailboxAddr`。

针对 SAMD21E16B 以及最新版 ASF 自定义 ASF SAM-BA 小程序链接器脚本

1. 从 SAM-BA 安装文件夹中选择 `applets\samd21j18a\sam-ba_applets\linker_script\sram_samba.lds` 文件，然后根据选择的器件以及 SAM-BA 自举程序占用的 SRAM 大小更新相应区段。

针对 SAMD21E16B 以及最新版 ASF 自定义 ASF SAM-BA Applet Makefile

1. 从 SAM-BA 安装文件夹中选择 `applets\samd21j18a\sam-ba_applets\flash\Makefile`。
2. 将 `CHIP_NAME` 更新为新器件的名称，例如 `__SAMD21E16B__`，然后将 `ASF_BRANCH_PATH` 更新为最新版 ASF 的独立目录。

下图显示了在 ASF 3.36.2 或更高版本的 ASFv3 中需要对 Makefile 进行哪些更改才能使用 SAMD21E16B。

注：对于其他版本的 ASF 同样适用。

图 2-1. 在 ASF 3.36.2 中使用 SAMD21E16B 所需的 Makefile 更改

```

ifeq (samd21j18a,$(CHIP))
BOARD_DIR += samd21_xplained_pro
CHIP_NAME += __SAMD21E16B__
endif
MEMORIES += sram
# Trace level used for compilation
# (can be overridden by adding TRACE_LEVEL=number to the command-line)

LD += $(CROSS_COMPILE)ld
SIZE += $(CROSS_COMPILE)size
OBJCOPY += $(CROSS_COMPILE)objcopy
OBJDUMP += $(CROSS_COMPILE)objdump

# Flags
ASF_BRANCH_PATH += D:\TEMP_ASF\asf-standalone-archive-3.36.2.65\jdk-asf-3.36.2
GCC_LIB_PATH += $(ASF_BRANCH_PATH)/thirdparty/CMSIS/Lib/GCC
#INCLUDES += -I$(PATH_ATML_LIB_BOARD)
#INCLUDES += -I$(PATH_ATML_LIB_BOARD)/include
#INCLUDES += -I$(PATH_ATML_LIB_BOARD)/source
#INCLUDES += -I$(PATH_ATML_LIB_CHIP)
#INCLUDES += -I$(PATH_ATML_LIB_CHIP)/include

INCLUDES += -I$(ASF_BRANCH_PATH)/sam0/drivers/system/interrupt
INCLUDES += -I$(ASF_BRANCH_PATH)/sam0/drivers/nvm
INCLUDES += -I$(ASF_BRANCH_PATH)/thirdparty/CMSIS
INCLUDES += -I$(ASF_BRANCH_PATH)/thirdparty/CMSIS/Include
#INCLUDES += -I$(ASF_BRANCH_PATH)/thirdparty/CMSIS/Lib
#INCLUDES += -I$(ASF_BRANCH_PATH)/thirdparty/CMSIS/Lib/GCC
INCLUDES += -I$(ASF_BRANCH_PATH)/sam0/drivers/system/reset/reset_sam_d_r_h
INCLUDES += -I$(ASF_BRANCH_PATH)/sam0/drivers/system/interrupt/system_interrupt_samd21
INCLUDES += -I$(ASF_BRANCH_PATH)/sam0/drivers/system/clock/clock_samd21_r21_da_ha1
INCLUDES += -I$(ASF_BRANCH_PATH)/sam0/drivers/system/power/power_sam_d_r_h

```

注：可以根据所用的 ASF 版本修改包含路径依赖关系。

自定义 ASF SAM-BA 小程序源以使用 SERCOM0

1. 在 `applet_main method>sam-ba installation >\applets\samd21j18a\sam-ba_applets\flash\flash_app_main.c` 中，修改相应代码以使用 `SERCOM0`，而不是 `SERCOM3`。

图 2-2. 修改代码

```

exit:
    /* Acknowledge the end of command */
    TRACE_INFO("\tEnd of Applet %x %x.\n\r",
        (uint32_t)pMailbox->command,
        (uint32_t)pMailbox->status);

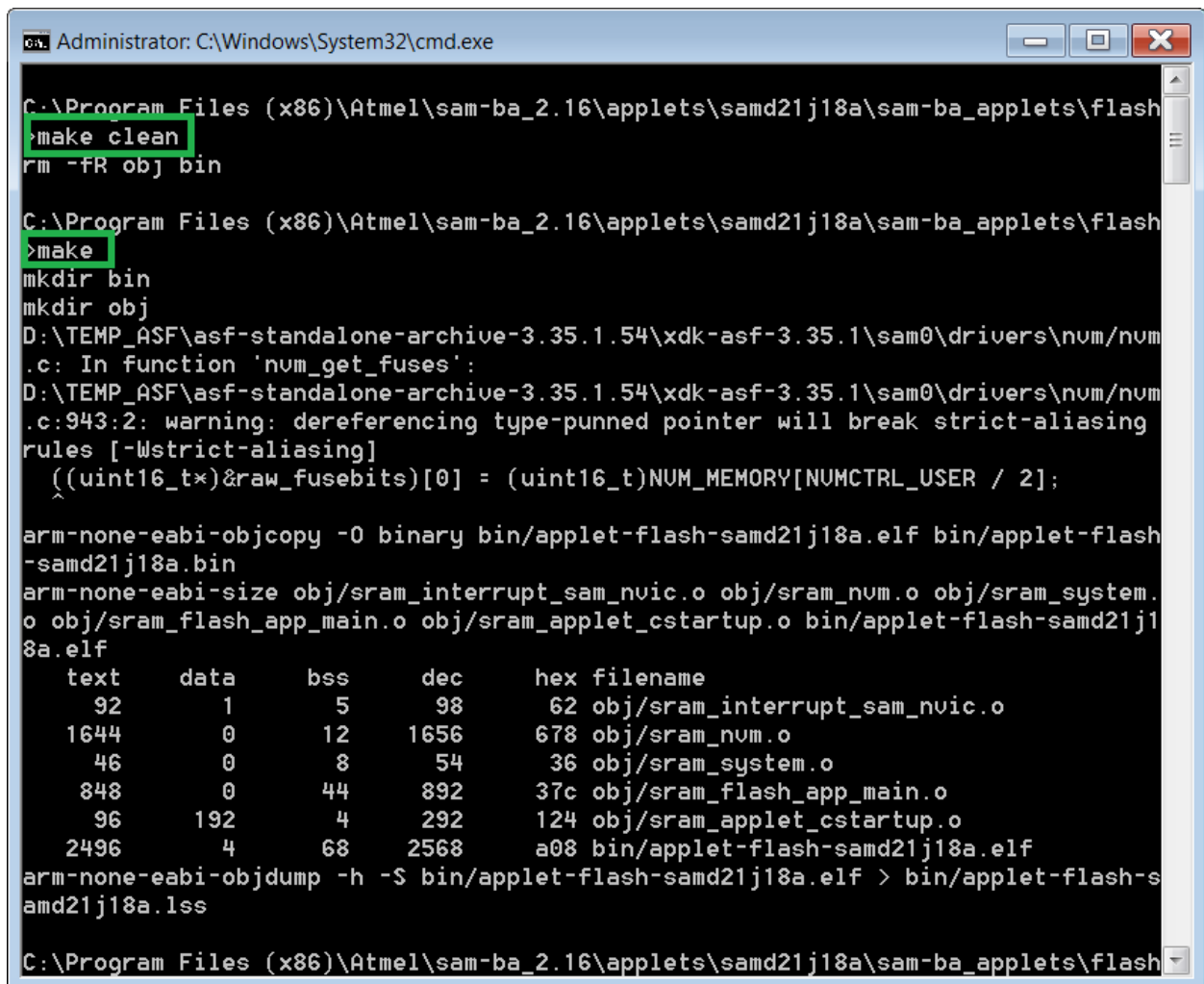
    /* Notify the host application of the end of the command processing */
    pMailbox->command = ~(pMailbox->command);

    SERCOM0->USART.DATA.reg = 0x6;
    return 0;
}

```

2. 打开 Atmel Studio 7.0 命令提示符，切换到 `sam-ba installation >\applets\samd21j18a\sam-ba_applets\flash` 文件夹位置，然后执行下图中用绿色框突出显示的命令。

图 2-3. 打开 Atmel Studio



已知问题:

- 用于 Cortex-M0+ 的 ASF SAM-BA 自举程序未针对 SAM Cortex-M0+ 系列的存储器约束变量进行优化，因此可能不适用于小于 8 KB 的 SRAM。
- 在 SAM-BA v2.18 中，某些器件（例如 SAMD21E16B）无法使用 SAM-BA 小程序、TCL 脚本和监视器固件。
- 如果下载的文件较大（即超过 32 KB），SAM-BA GUI 可能会看起来处于挂起状态，但实际上它正在后台处理请求。
- 某些版本的 GCC 编译器（例如 GCC 6.3.1.508）可能会存在优化问题，从而导致 SAM-BA GUI 挂起。

**提示与技巧:**

- **USB 转串行转换器:** Xplained Pro 板上的 EDBG 芯片是一个全速 USB 转串行桥接器，因此要想获得类似的性能，用户需使用全速 USB 转串行转换器。



### 3. 其他相关资源

有关 SAM-BA 自举程序的更多信息，请参见以下文档，下载位置如下：

- [http://www.microchip.com/avr-support/advanced-software-framework-\(asf\)](http://www.microchip.com/avr-support/advanced-software-framework-(asf))
- [http://asf.atmel.com/docs/latest/sam0.applications.samba\\_bootloader.samc21\\_xplained\\_pro/html/index.html](http://asf.atmel.com/docs/latest/sam0.applications.samba_bootloader.samc21_xplained_pro/html/index.html)
- <http://www.microchip.com/DevelopmentTools/ProductDetails.aspx?PartNO=Atmel%20SAM-BA%20In-system%20Programmer>
- [https://microchip.secure.force.com/microchipknowledge/articles/en\\_US/FAQ/How-to-change-main-clock-source-for-SAMR21-SAMD21-SAMD20-SAMD10-SAMD11-in-ASF-project](https://microchip.secure.force.com/microchipknowledge/articles/en_US/FAQ/How-to-change-main-clock-source-for-SAMR21-SAMD21-SAMD20-SAMD10-SAMD11-in-ASF-project)
- [http://ww1.microchip.com/downloads/en/AppNotes/Atmel-42366-SAM-BA-Bootloader-for-SAM-D21\\_ApplicationNote\\_AT07175.pdf](http://ww1.microchip.com/downloads/en/AppNotes/Atmel-42366-SAM-BA-Bootloader-for-SAM-D21_ApplicationNote_AT07175.pdf)

## 4. 常见问题

### 在构建项目时，如何使用已知数据填充未使用的闪存位置？

有时，需要使用已知数据填充未使用的闪存位置。下列步骤介绍了一种填充方法：

对于 ARM 器件：

1. 在应用程序中创建一个虚拟区段。链接器需要使用此区段来填充数据。

```
/** Creating a dummy section to fill unused flash with known data (0xFF) */
const U8 u8Dummy __attribute__((section(".fill_known_data"))) = 0xFF;
```

2. 通过链接器脚本使用已知数据填充此区段。
  - 使用链接器重新定位已初始化的数据。

```
.relocate :
{
    . = ALIGN(4);
    _srelocate = .;
    *(.ramfunc .ramfunc.*);
    *(.data .data.*);
    . = ALIGN(4);
    _erelocate = .;
} > ram AT > rom
```

在默认脚本中，找到 `.relocate : AT (_etext)`，然后对其进行更改，使其与给定代码匹配。

- 使用 `FF` 填充此区段，直到 `rom` 末尾

```
.Fill_FF :
{
    location */
        _FillStartAddress = .; /* Provides start address of Filled
        KEEP(*(.fill_known_data))
        FILL(0xFF);
        . = ORIGIN(rom) + LENGTH(rom) - _FillStartAddress; /* Fill till
    end of rom */
        _FillEndAddress = .; /* Provides end address of Filled
    location */
} AT > rom
```

在上述示例中，我们使用 `0xFF` 填充未使用的位置。通过将 `0xFF` 更改为另一个值，即可使用已知数据填充未使用的位置。

### 为何我的应用程序会使用这么多的 RAM？

在启动时，链接器脚本会为堆栈分配存储器。此链接器脚本中定义了分配的大小。默认情况下，该值非常大。这样做的目的是为了确保用户不会轻易遇到堆栈溢出问题，因为此类问题很难调试。每个应用程序都应配置该值，以最大限度地利用存储器。

---

## Microchip 网站

---

Microchip 网站 <http://www.microchip.com/> 为客户提供在线支持。客户可通过该网站方便地获取文件和信息。只要使用常用的互联网浏览器即可访问，网站提供以下信息：

- **产品支持**——数据手册和勘误表、应用笔记和示例程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及归档软件
- **一般技术支持**——常见问题（FAQ）、技术支持请求、在线讨论组以及 Microchip 顾问计划成员名单
- **Microchip 业务**——产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

---

## 变更通知客户服务

---

Microchip 的变更通知客户服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时，收到电子邮件通知。

欲注册，请登录 Microchip 网站 <http://www.microchip.com/>。在“支持”（Support）下，点击“变更通知客户”（Customer Change Notification）服务后按照注册说明完成注册。

---

## 客户支持

---

Microchip 产品的用户可通过以下渠道获得帮助：

- 代理商或代表
- 当地销售办事处
- 应用工程师（FAE）
- 技术支持

客户应联系其代理商、代表或应用工程师（FAE）寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过以下网站获得技术支持：<http://www.microchip.com/support>

---

## Microchip 器件代码保护功能

---

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。这样做的人极有可能侵犯了知识产权。
- Microchip 愿意与关心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。

## 法律声明

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，否则在 Microchip 知识产权保护下，不得暗或以其他方式转让任何许可证。

## 商标

Microchip 的名称和徽标组合、Microchip 徽标、AnyRate、AVR、AVR 徽标、AVR Freaks、BitCloud、chipKIT、chipKIT 徽标、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、Heldo、JukeBlox、KeeLoq、Kleer、LANCheck、LINK MD、maXStylus、maXTouch、MediaLB、megaAVR、MOST、MOST 徽标、MPLAB、OptoLyzer、PIC、picoPower、PICSTART、PIC32 徽标、Prochip Designer、QTouch、SAM-BA、SpyNIC、SST、SST 徽标、SuperFlash、tinyAVR、UNI/O 和 XMEGA 是 Microchip Technology Incorporated 在美国和其他国家或地区的注册商标。

ClockWorks、The Embedded Control Solutions Company、EtherSynch、Hyper Speed Control、HyperLight Load、IntelliMOS、mTouch、Precision Edge 和 Quiet-Wire 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、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、motorBench、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.、SQL、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、ViewSpan、WiperLock、Wireless DNA 和 ZENA 为 Microchip Technology Incorporated 在美国和其他国家或地区的商标。

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

Silicon Storage Technology 为 Microchip Technology Inc. 在除美国外的国家或地区的注册商标。

GestIC 是 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。

© 2018, Microchip Technology Incorporated 版权所有。

ISBN: 978-1-5224-3308-8

---

## DNV 认证的质量管理体系

---

### ISO/TS 16949

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<sup>®</sup> MCU 和 dsPIC<sup>®</sup> DSC、KEELOQ<sup>®</sup>跳码器件、串行 EEPROM、单片机外设、非易失性存储器及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。

## 全球销售及服务中心

美洲	亚太地区	亚太地区	欧洲
<b>公司总部</b> 2355 West Chandler Blvd. Chandler, AZ 85224-6199 电话: 1-480-792-7200 传真: 1-480-792-7277 技术支持: <a href="http://www.microchip.com/support">http://www.microchip.com/support</a> 网址: <a href="http://www.microchip.com">www.microchip.com</a>	<b>中国 - 北京</b> 电话: 86-10-8569-7000 <b>中国 - 成都</b> 电话: 86-28-8665-5511 <b>中国 - 重庆</b> 电话: 86-23-8980-9588 <b>中国 - 东莞</b> 电话: 86-769-8702-9880 <b>中国 - 广州</b> 电话: 86-20-8755-8029 <b>中国 - 杭州</b> 电话: 86-571-8792-8115 <b>中国 - 南京</b> 电话: 86-25-8473-2460 <b>中国 - 青岛</b> 电话: 86-532-8502-7355 <b>中国 - 上海</b> 电话: 86-21-3326-8000 <b>中国 - 沈阳</b> 电话: 86-24-2334-2829 <b>中国 - 深圳</b> 电话: 86-755-8864-2200 <b>中国 - 苏州</b> 电话: 86-186-6233-1526 <b>中国 - 武汉</b> 电话: 86-27-5980-5300 <b>中国 - 西安</b> 电话: 86-29-8833-7252 <b>中国 - 厦门</b> 电话: 86-592-2388138 <b>中国 - 香港特别行政区</b> 电话: 852-2943-5100 <b>中国 - 珠海</b> 电话: 86-756-3210040 <b>台湾地区 - 高雄</b> 电话: 886-7-213-7830 <b>台湾地区 - 台北</b> 电话: 886-2-2508-8600 <b>台湾地区 - 新竹</b> 电话: 886-3-577-8366	<b>澳大利亚 - 悉尼</b> 电话: 61-2-9868-6733 <b>印度 - 班加罗尔</b> 电话: 91-80-3090-4444 <b>印度 - 新德里</b> 电话: 91-11-4160-8631 <b>印度 - 浦那</b> 电话: 91-20-4121-0141 <b>日本 - 大阪</b> 电话: 81-6-6152-7160 <b>日本 - 东京</b> 电话: 81-3-6880-3770 <b>韩国 - 大邱</b> 电话: 82-53-744-4301 <b>韩国 - 首尔</b> 电话: 82-2-554-7200 <b>马来西亚 - 吉隆坡</b> 电话: 60-3-7651-7906 <b>马来西亚 - 檳榔嶼</b> 电话: 60-4-227-8870 <b>菲律宾 - 马尼拉</b> 电话: 63-2-634-9065 <b>新加坡</b> 电话: 65-6334-8870 <b>泰国 - 曼谷</b> 电话: 66-2-694-1351 <b>越南 - 胡志明市</b> 电话: 84-28-5448-2100	<b>奥地利 - 韦尔斯</b> 电话: 43-7242-2244-39 传真: 43-7242-2244-393 <b>丹麦 - 哥本哈根</b> 电话: 45-4450-2828 传真: 45-4485-2829 <b>芬兰 - 埃斯波</b> 电话: 358-9-4520-820 <b>法国 - 巴黎</b> 电话: 33-1-69-53-63-20 传真: 33-1-69-30-90-79 <b>德国 - 加兴</b> 电话: 49-8931-9700 <b>德国 - 哈恩</b> 电话: 49-2129-3766400 <b>德国 - 海尔布隆</b> 电话: 49-7131-67-3636 <b>德国 - 卡尔斯鲁厄</b> 电话: 49-721-625370 <b>德国 - 慕尼黑</b> 电话: 49-89-627-144-0 传真: 49-89-627-144-44 <b>德国 - 罗森海姆</b> 电话: 49-8031-354-560 <b>以色列 - 赖阿南纳</b> 电话: 972-9-744-7705 <b>意大利 - 米兰</b> 电话: 39-0331-742611 传真: 39-0331-466781 <b>意大利 - 帕多瓦</b> 电话: 39-049-7625286 <b>荷兰 - 德卢内市</b> 电话: 31-416-690399 传真: 31-416-690340 <b>挪威 - 特隆赫姆</b> 电话: 47-7289-7561 <b>波兰 - 华沙</b> 电话: 48-22-3325737 <b>罗马尼亚 - 布加勒斯特</b> 电话: 40-21-407-87-50 <b>西班牙 - 马德里</b> 电话: 34-91-708-08-90 传真: 34-91-708-08-91 <b>瑞典 - 哥德堡</b> 电话: 46-31-704-60-40 <b>瑞典 - 斯德哥尔摩</b> 电话: 46-8-5090-4654 <b>英国 - 沃金厄姆</b> 电话: 44-118-921-5800 传真: 44-118-921-5820
<b>亚特兰大</b> 德卢斯, 乔治亚州 电话: 1-678-957-9614 传真: 1-678-957-1455 <b>奥斯汀, 德克萨斯州</b> 电话: 1-512-257-3370 <b>波士顿</b> 韦斯特伯鲁, 马萨诸塞州 电话: 1-774-760-0087 传真: 1-774-760-0088 <b>芝加哥</b> 艾塔斯卡, 伊利诺伊州 电话: 1-630-285-0071 传真: 1-630-285-0075 <b>达拉斯</b> 艾迪生, 德克萨斯州 电话: 1-972-818-7423 传真: 1-972-818-2924 <b>底特律</b> 诺维, 密歇根州 电话: 1-248-848-4000 <b>休斯敦, 德克萨斯州</b> 电话: 1-281-894-5983 <b>印第安纳波利斯</b> 诺布尔斯维尔, 印第安纳州 电话: 1-317-773-8323 传真: 1-317-773-5453 电话: 1-317-536-2380 <b>洛杉矶</b> 米申维耶霍, 加利福尼亚州 电话: 1-949-462-9523 传真: 1-949-462-9608 电话: 1-951-273-7800 <b>罗利, 北卡罗来纳州</b> 电话: 1-919-844-7510 <b>纽约, 纽约州</b> 电话: 1-631-435-6000 <b>圣何塞, 加利福尼亚州</b> 电话: 1-408-735-9110 电话: 1-408-436-4270 <b>加拿大 - 多伦多</b> 电话: 1-905-695-1980 传真: 1-905-695-2078			