

Kinetis E 系列上的 UART Boot Loader 设计

通过 : Wang Peng

在一个特定的场合，很多应用或产品都需要升级固件，以修复 Bug 或提高性能。大多数人都不愿意使用专用的调试接口，而是使用 UART、USB、I2C 等通信接口。这种情况下，就需要一个串行 Boot Loader 通过其中一个通信接口升级固件，而不需要调试器或特定的程序工具。

本文档将指导您如何使用 UART（通用异步接收器 / 发送器）接口在 Kinetis E 系列上设计 Boot Loader。本文中列出的示例代码是在 IAR 6.50 中开发的。

1 简介

Boot Loader 是一种内置固件，用于通过通信接口将应用代码编程到片上非易失性存储器（Kinetis E 上的 Flash）。本文档将介绍如何在 FRDM-KE02Z 板上设计 UART Boot Loader。

该程序将使用 AN2295 PC 端主机 GUI 工具解码 s19 文件，然后通过 UART 接口将应用代码传输到目标 MCU。然后写入到 Flash。

内容

1. 简介	1
2. 软件体系结构	2
3. 结论	5
4. 参考	5
5. 术语表	5

2 软件体系结构

可以从飞思卡尔网站上下载 `win_hc08sprg.exe` 软件，使用该工具解码 S19 文件并与目标 MCU 通信。该工具兼容 FC 协议。关于详细的信息，请参见 www.freescale.com 上提供的 AN2295。

内置 Flash 上的 Boot Loader 在通电后将开始运行，并确定握手是否成功。如果发生超时，将跳转到应用代码的入口地址。

2.1 Flash 操作

KE02 提供了 flash 操作的高级功能：闪存控制器 (FMC) 和 flash 访问控制，它们不仅能够加速对 flash 的存取，避免在访问 flash 过程中等待，而且还能在 flash 繁忙时避免冲突。下面我们介绍两种 Flash 编程方法：

1. 通过 RAM 中运行的代码编程 Flash
2. 通过 Flash 中运行的代码编程 Flash

2.1.1 RAM 中运行的代码

将 Flash 启动命令的代码复制到 RAM 并禁用中断，使代码可以在 RAM 中运行，不需要从 Flash 中读取指令。此方法在传统的片上 Flash 设备中使用最为广泛。

用于启动 Flash 命令和检查 Flash 状态的代码片段如下：

```
__ramfunc void FTMRH_LaunchCMD(uint8_t bWaitComplete)
{
    DisableInterrupts;
    if(bWaitComplete)
    {
        // Wait till command is completed
        FTMRH_FSTAT = 0x80;
        while (!(FTMRH_FSTAT & FTMRH_FSTAT_CCIF_MASK));
    }
    EnableInterrupts;
}
```

2.1.2 Flash 中运行的代码

KE02 包含 FMC 功能，这是一个极其重要的外围设备，连接于 Flash 和 MCU 之间。KE02 还包含 Flash 访问控制、预取缓冲器、单入口缓冲器和缓存，因此，在 Flash 中运行的代码能够编程和擦写 Flash。如果启用 Flash 访问控制功能，当 Flash 繁忙时，该控制器能够暂时阻止对 Flash 的存取，直到 Flash 变为空闲状态，才允许对 Flash 的访问。用这种方法不需要禁用中断，从而提高代码的效率。

相应的程序片段如下：

```
void FTMRH_LaunchCMD(uint8_t bWaitComplete)
{
    /* enable stalling flash controller when flash is busy */
    MCM_PLACR |= MCM_PLACR_EFSC_MASK;
    FTMRH_FSTAT = 0x80;
    if(bWaitComplete)
    {
        // Wait till command is completed
        while (!(FTMRH_FSTAT & FTMRH_FSTAT_CCIF_MASK));
    }
}
```

2.2 中断向量表重定位

在 Boot Loader 的设计方面，另一个重要事项是如何处理中断向量表。Kinetis E 系列支持中断向量表重定位。默认情况下，向量表的地址为 0x00 到 0xBF。用户可将向量表重定位到其他任何可用地址，例如其他的 Flash 地址或者是 RAM 地址。

Boot Loader 应用会将 Boot Loader 代码和应用代码分配到不同的 Flash 空间。请参见下图来了解 Boot Loader 和应用代码的内存分配。

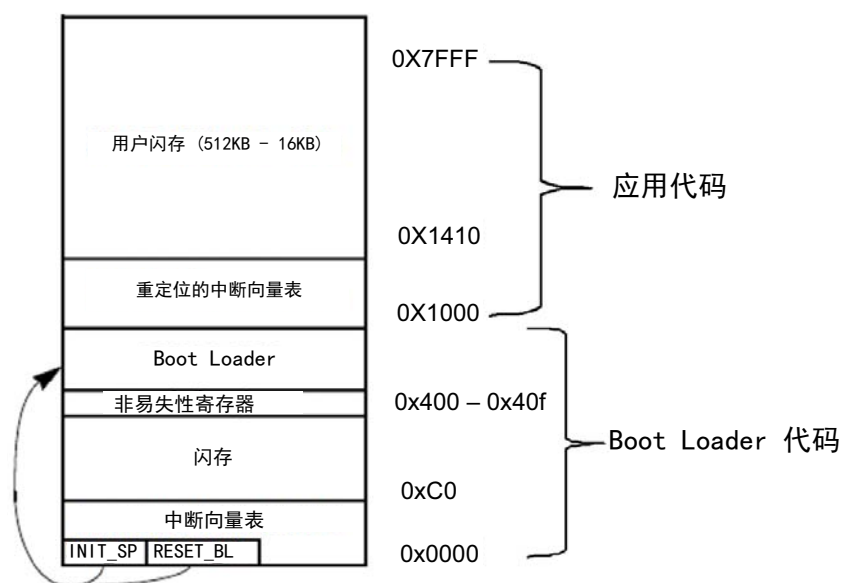


图 2-1. 内存分配

要重定位中断向量表，可以简单地用下面的方法写重定位地址到 SCB_VTOR 寄存器：

```
SCB_VTOR = RELOCATION_ADDRESS;
```

在用 boot loader 下载用户程序时，PC 工具会解码 S19 文件（应用代码），并将地址空间 (0x00 - 0x3FF) 的内容写入重定位地址（0x1000 到 0x13FF）。这可以确保在复位后，Boot Loader 首先开始运行，然后再跳转到应用入口地址。以下章节介绍了复位后的软件流程。

2.3 软件流程图

MCU 上电后，软件将先运行 Boot Loader，以检查握手是否成功。如果发生超时，则会初始化 SP，然后把用户中断向量地址写入 SCB_VTOR 寄存器。然后跳转到用户代码。

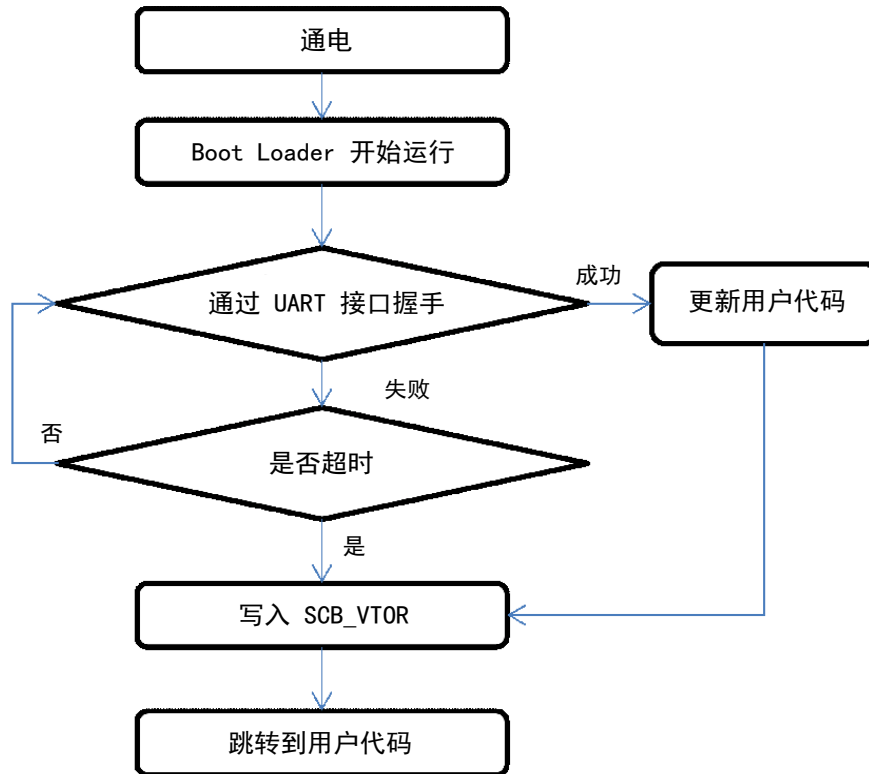


图 2-2. 软件流程图

建议编辑 Flash 保护区，以避免意外擦除 Boot Loader。

2.4 用户代码

用户代码必须定位到指定的位置，然后才能通过 Boot Loader 下载到 Flash。

2.4.1 链接文件

可以使用普通项目将代码起始地址定义为任何可接受的地址。例如，在 IAR 中：

```
define symbol __code_start__ = __ICFEDIT_region_ROM_start__ + 0x410;
```

在本例中，定义的代码起始地址偏移为 0x410。

对于使用 Boot Loader 下载代码的用户代码，需要根据 Boot Loader 里面的“FC_protocol.h”文件里面定义的中断向量表重定义地址来定义代码的开始地址

请参见 FC_protocol.h 中的以下宏定义。

```
#define RELOCATION_VERTOR_ADDR 0x1000
```

代码起始地址应是 RELOCATION_VERTOR_ADDR 加 0x410。地址空间从“RELOCATION_VERTOR_ADDR”到“RELOCATION_VERTOR_ADDR + 0x410”是重定位的中断向量表。

2.4.2 中断向量表

在进入用户代码之前，Boot Loader 必须向 SCB_VTOR 寄存器写入重定位地址。如果用户代码需要将向量表复制到 RAM 中，则请在写入 SCB_VTOR 之前，将 RELOCATION_VECTOR_ADDR 的内容复制到目标 RAM 地址。不应向此寄存器写入任何其他值。

2.4.3 Flash 配置区域

Flash 配置区域的地址为 Boot Loader 区域中的 0x400 到 0x40F。如果该区域受保护，不能对它进行修改。否则，可以擦除或者是编程这部分区域。用户代码首先需要读出地址 0x400 到 0x40F 上包含 Flash 配置字段扇区的所有内容。然后，擦除此扇区，修改缓冲区里面的内容，再写回此扇区。

附注

请不要定义常量在地址 0x400 到 0x40f，因为这样会导致无法通过 Boot Loader 下载代码。

3 结论

本文档介绍了使用 AN2295 PC 端主机软件实施 UART Boot Loader 的方法。在某些特殊应用中，用户无需使用其他编程工具，就能方便地更新用户代码。

4 参考

KE02 子系列参考手册 (MKE02Z64M20SF0RM)

开发人员系列 Boot Loader (AN2295)

Kinetis L 外围模块快速参考用户指南 (KLQRUG)

5 术语表

UART	通用异步接收器 / 发送器
FCCOB	Flash 通用命令对象
WDOG	看门狗
MCG	多用途时钟发生器
FMC	Flash 控制器

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

本文档中的信息仅供系统和软件实施方使用飞思卡尔产品。未包含基于本文档信息设计或加工任何集成电路的任何明确或隐含的版权许可授权。

飞思卡尔保留对此中所有产品进行更改的权利，如有更改，恕不另行通知。飞思卡尔对其产品在任何特定用途方面的适用性不做任何担保、声明或保证，也不承担因为应用或使用产品或电路所产生的任何责任，特别声明，拒绝承担包括但不限于因果性或附带损害在内的所有责任。飞思卡尔数据表和 / 或技术规格中所提供的“典型”参数在不同应用中可能，并且确实不同，实际性能会随时间而有所变化。所有操作参数，包括“典型值”在内，在每个客户应用中必须经由客户的技术专家进行验证。飞思卡尔未转让与其专利权及其他权利相关的许可。飞思卡尔销售产品时遵循以下网址中包含的标准销售条款和条件：freescale.com/SalesTermsandConditions

Freescale, and the Freescale logo, and Kinetis are trademarks of Freescale Semiconductor, Inc., Reg.U.S.Pat.& Tm.Off.All other product or service names are the property of their respective owners.ARM is a registered trademark of ARM Limited.

© 2013 Freescale Semiconductor, Inc.

Document Number: AN4767
Rev. 0
7/2013

