

KE06 CAN引导加载程序设计

作者: Jonson Chen

1 概述

很多应用或产品都需要现场升级固件，以便修复某些漏洞，或者用来提高性能。大多数应用和产品的固件升级并不使用专用的调试接口，而是使用UART、USB和I²C等通信接口。这种情况下，就需要一个串行引导加载程序，以便在不使用调试器或专用编程工具的情况下，通过某个通信接口执行固件升级。

本应用笔记将指导您使用CAN接口在KE06 MCU上设计引导加载程序。

目录

1	概述	1
2	简介	2
3	软件架构	2
3.1	转接板	2
3.2	目标板	4
4	存储器分配	10
5	结语	11
6	参考	11
7	首字母缩略词和缩写词	11
8	修订历史记录	11

2 简介

引导加载程序是一种内置固件，用于通过通信接口将应用程序代码编程到闪存中。本应用笔记说明了使用FRDM-KE06Z板将UART数据从PC终端转换到CAN总线的过程。此外，还解释了与目标板FRDM-KE06Z通信以实现目标应用程序代码更新的过程。

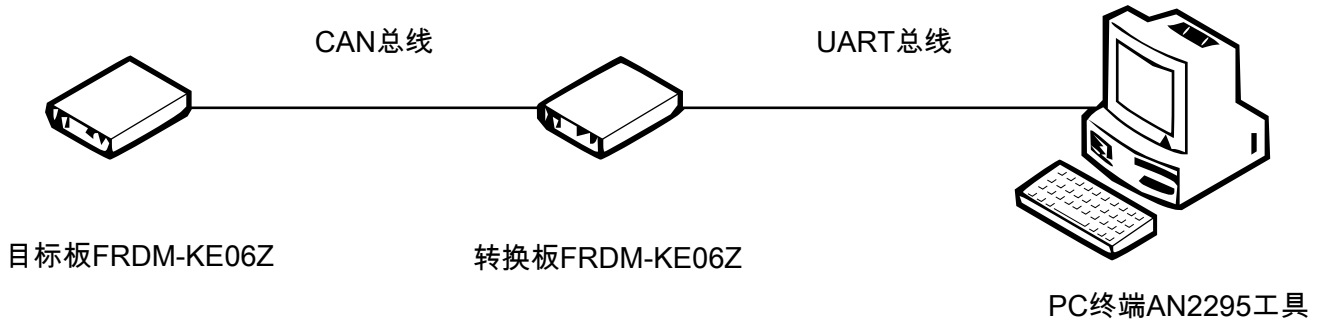


图1. 顶层视图

引导加载程序利用AN2295SW_Rev1软件工具（广泛应用于所有Kinetis器件）的功能，通过UART接口更新应用程序代码。

转接板使用Freedom FRDM-KE06Z板将UART总线转换为CAN总线，将数据重新打包并传输到目标板。目标板将应用程序代码编程到闪存中。

CAN引导加载程序的示例代码可以直接在FRDM-KE06Z板上运行，并将该代码下载到目标板、“Bridge_UARTToCAN”用于转接板，项目“RTC_demo”用于生成S19文件，该文件可使用PC软件下载。

3 软件架构

Win_hc08sprg.exe软件用来解码S19文件，并通过FC协议与转接板进行通信。

3.1 转接板

PC不能通过CAN直接与目标板通信，因此我们需要使用转接板将UART信号（PC端）转换为CAN信号（目标板）。因而，转接板通过FC协议与PC终端通信。转接板重新打包数据帧的数据长度和校验和，利用CAN总线与目标板收发数据包。

表1. 数据包内容

数据长度	原始数据帧	校验和
------	-------	-----

然后，从目标板读取数据。接收到的第一个数据用于确定目标板是否就绪。如果就绪（命令|0x80），表示接收器接收到了正确应答。例如，发送到目标板的命令为0x03，则接收到的ACK应为0x03|0x80。

首先，它将向目标板发送FC_CMD_HOOK（0x02），然后从目标板读取状态，检查目标板是工作于引导加载程序模式，还是用户代码模式。如果接收到的状态为FC_CMD_HOOK|0x80，则发送0Xfc和PC终端握手，否则一直检查目标板的状况，直到收到FC_CMD_HOOK|0x80为止。以下为转接板的软件流程图：

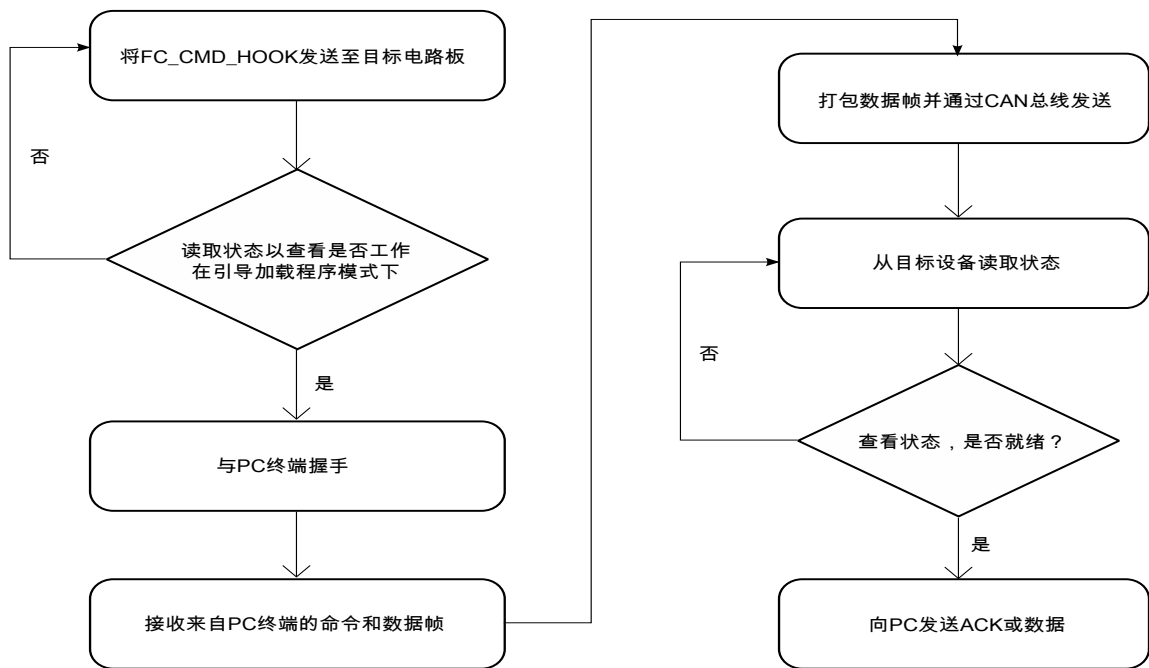


图2. 转接板的软件流程图

转接板充当PC终端与目标板之间的桥梁，使用转接板可以通过PC将S19文件下载到目标板。

3.2 目标板

目标板包含内置引导加载程序代码。启动后，目标板首先检查引导加载程序代码的工作模式，即引导模式或用户代码模式。可以通过多种方式进行该检查。例如，检查外部GPIO的电平，如果GPIO引脚为低电平，则目标板将进入引导模式来运行引导加载程序；如果GPIO引脚为高电平，则进入用户代码模式来运行应用程序代码。

由于MCU上的GPIO资源有限，可能没有用于引导加载程序检测模式的GPIO引脚。这里介绍另一种方式，通过握手命令确定工作模式。如果发生超时且握手失败，则目标板进入用户模式；如果成功，则目标板进入引导加载程序模式。

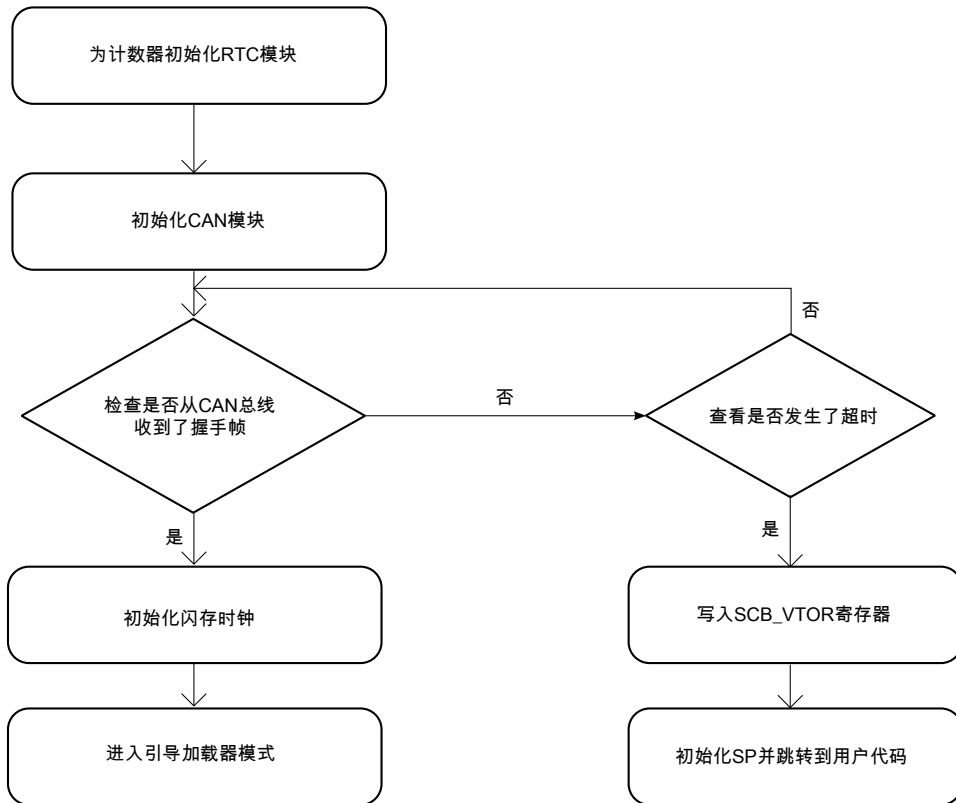


图3. 检查标志流程图

3.2.1 CAN节点驱动程序

目标板将CAN配置为CAN节点。在CAN中断服务程序中接收和发送数据。关于CAN中断流程的更多信息，请参见KE06Z参考手册。以下为CAN驱动程序的示例代码段，用于CAN通信处理：

```
void MSCAN_RxProcessing( void )
{
    u32RxInterruptCounter ++;
    if(CAN_IsRxBuffFull(MSCAN))
    {
        if(u8RxFrameBufferFreeLength!=0)
        {
            CAN_ReadOneFrameFromBuff(MSCAN,&sRxFrame[u8RxFrameBufferIndex++]);
            if(u8RxFrameBufferIndex>=CAN_BUFFER_LENGTH)
            {
                u8RxFrameBufferIndex = 0;
            }
            u8RxFrameBufferFreeLength--;
        }
        else
        {
            //receive frame buffer is full, clear buffer in MSCAN
            // clear receiver full flag
            CAN_ClearRXF_Flag(MSCAN);
        }
    }
}

void MSCAN_TxProcessing( void )
{
    if(CAN_IsOverRunFlag(MSCAN))
    {
        // overrun error occur
        CAN_ClearOVRIF_Flag(MSCAN);
    }
    if(CAN_IsWakeUpIntFlag(MSCAN))
    {
        CAN_ClearWUPIF_Flag(MSCAN);
    }
    if(CAN_IsStatusChangeFlag(MSCAN))
    {
        CAN_ClearCSCIF_Flag(MSCAN);
        // Get currently status
        CAN_GetReceiverStatus(MSCAN);
        CAN_GetReceiveErrorCount(MSCAN);
    }
    if(!CAN_CheckSendBufferFrame(MSCAN,&sCAN_TxBuff))
    {
        // no data in transmitting buffer,disbale interrupt
        CAN_TransmitterEmptyIntDisable(MSCAN);
    }
}
```

CAN在接收到数据帧后，会将gbI2CRecFrame Flag标志置1，以便应用程序代码可以继续处理数据帧。

3.2.2 命令说明

应始终在引导加载程序循环中检查gbI2CRecFrameFlag标志，当标志为1时，引导加载程序开始处理接收到的数据帧。首先，使用校验和来验证接收到的数据帧是否正确。验证后，解包数据帧并执行适当命令。

表2. 命令包

总数据长度 (4字节)	命令 (1字节)	地址 (4字节)	数据长度 (1字节)	数据	校验和 (1字节)
----------------	-------------	-------------	---------------	----	--------------

下表对所有命令进行了说明：

表3. 命令列表

命令功能	命令	加载程序肯定应答	加载程序否定应答
握手	0x02	0x82,0xFC	0x82,0x03
标识	0x49	0xc9, 标识信息	0xc9,0x03
擦除扇区	0x45	0xc5,0xfc	0xc5,0x03
写入	0x57	0xd7,0xfc	0xd7,0x03
读取	0x52	0xd2,data	0xd2,0x03
退出	0x51	无应答	无应答

3.2.2.1 握手

由握手命令（编码为0x02）接收的数据包如下所示：

表4. 握手命令数据包

总数据长度 (4字节)	命令 (1字节)	地址 (4字节)	数据长度 (1字节)	数据	校验和 (1字节)
6	0x02	-	-	-	CS

下表显示了握手命令的命令应答：

表5. 握手命令应答

命令 (1字节)	状态
0x82	0xFC/0x03

如果接收的状态代码为0xFC，则表示目标板工作在引导加载程序模式下，并已准备好与转接板通信。

如果状态代码为0x03，则目标板工作在用户模式下，不能接收任何其他命令。

3.2.2.2 标识命令

由标识命令（编码为0x49）接收的数据包如下所示：

表6. 标识命令数据包

总数据长度 (4字节)	命令 (1字节)	地址 (4字节)	数据长度 (1字节)	数据	校验和 (1字节)
6	0x49	-	-	-	CS

以下为MCU所需的信息：

- 协议版本 — 1字节
- KE06 80LQFP的系统设备标识寄存器（SDID）内容（0x06080000），r（23-20位）反映当前芯片等级的芯片版本号 — 2字节
- 可重新编程存储器区域的数量 — 4字节
- 可重新编程存储器区域的起始地址 — 4字节
- 可重新编程存储器区域的结束地址 — 4字节
- 原始向量表的地址（1 KB） — 4字节
- 新向量表的地址（1 KB） — 4字节
- MCU擦除块的长度 — 4字节
- MCU写入块的长度 — 4字节
- 以零结尾的标识字符串 — n个字节

以下结构体用于识别MCU的信息:

```
typedef uint32_t addrtype;
typedef struct
{
    unsigned char Reserve ;           // reserve bytes for 4 bytes allign
    unsigned char Version;           // version
    uint16_t Sdid;                   // Sd Id */
    addrtype BlocksCnt;              // count of flash blocks
    addrtype FlashStartAddress;      // flash blocks descriptor
    addrtype FlashEndAddress;
    addrtype RelocatedVectors;       // Relocated interrupts vector table
    addrtype InterruptsVectors;      // Interrupts vector table
    addrtype EraseBlockSize;         // Erase Block Size
    addrtype WriteBlockSize;        // Write Block Size
    char IdString[ID_STRING_MAX];    // Id string
}FC_IDENT_INFO;
```

下表显示了标识命令的命令应答:

表7. 标识命令应答

命令 (1字节)	数据
0xc9	标识信息

3.2.2.3 擦除命令

由擦除指令（编码为0x45）接收的数据包如下所示:

表8. 擦除命令数据包

总数据长度 (4字节)	命令 (1字节)	地址 (4字节)	数据长度 (1字节)	数据	校验和 (1字节)
10	0x45	地址	-	-	CS

下表显示了擦除命令的命令应答:

表9. 擦除命令应答

命令 (1字节)	状态
0xc5	0xFC/0x03

3.2.2.4 写命令

由写命令（编码为0x57）接收的数据包如下所示：

表10. 写命令数据包

总数据长度 (4字节)	命令 (1字节)	地址 (4字节)	数据长度 (1字节)	数据	校验和 (1字节)
总长度	0x57	地址	数据长度	数据	CS

下表显示了写命令的命令应答：

表11. 写命令应答

命令 (1字节)	状态
0xd7	0xFC/0x03

3.2.2.5 读命令

由读命令（编码为0x52）接收的数据包如下所示：

表12. 读命令数据包

总数据长度 (4字节)	命令 (1字节)	地址 (4字节)	数据长度 (1字节)	数据	校验和 (1字节)
11	0x52	地址	要读取的数据长度	-	CS

下表显示了读命令的命令应答：

表13. 读命令应答

命令 (1字节)	数据
0xd2	数据

3.2.2.6 退出命令

该命令无需应答。

在接收到退出命令后，目标板将退出引导加载程序模式并进入用户模式。

4 存储器分配

引导加载程序代码占据闪存存储器的第一个区域（最低的存储器地址空间）。这种放置方式使可供用户使用的闪存空间的开始地址发生了变化，所以需要在用户的应用程序中修改链接文件（IAR中的ICF文件，CodeWarrior中的LCF文件）的应用程序开始地址。以下代码段演示了在IAR6.5中修改ICF链接器文件的方法：

```
// default linker file
define symbol __ICFEDIT_region_ROM_start__ = 0x00;
// modified Linker file for KE06Z 128KB flash
define symbol __ICFEDIT_region_ROM_start__ = 0x1000;
```

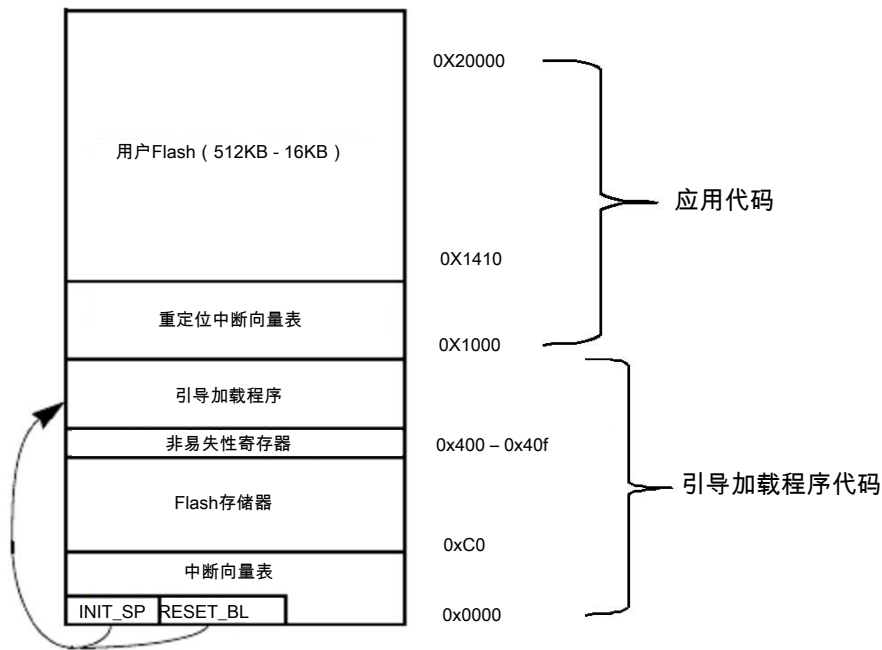


图4. 存储器分配

5 结语

本应用笔记说明了利用桥接板作为转换板、利用其他电路板作为目标板实现CAN引导加载程序的过程。用户也可以在应用软件中自行添加引导加载程序功能。

6 参考

可从freescale.com获取以下参考文档：

开发人员的串行引导加载程序应用笔记（文档AN2295）

7 首字母缩略词和缩写词

表14. 缩略词

术语	含义
CAN	控制器局域网
FCCOB	闪存通用命令对象
MCG	多用途时钟发生器
UART	通用异步接收器/发送器
WDOG	看门狗

8 修订历史记录

表15. 修订历史记录

修订版本号	日期	重要改动
0	2014/03	初始版本

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 Freescale 产品。本文并未明示或者暗示授予利用本文档信息进行设计或者加工集成电路的版权许可。Freescale 保留对此处任何产品进行更改的权利，恕不另行通知。

Freescale 对其产品在任何特定用途方面的适用性不做任何担保、表示或保证，也不承担因为应用程序或者使用产品或电路所产生的任何责任，明确拒绝承担包括但不限于后果性的或附带性的损害在内的所有责任。Freescale 的数据表和 / 或规格中所提供的 "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。所有运行参数，包括 "经典值" 在内，必须经由客户的技术专家对每个客户的应用程序进行验证。Freescale 未转让与其专利权及其他权利相关的许可。Freescale 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, 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 and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

© 2014 Freescale Semiconductor, Inc.

© 2014 飞思卡尔半导体有限公司。

