AN2295使用说明

Bootloader是MCU的一个重要功能，很多客户需要产品在出厂后具备在线更新固件的功能，通常情况下，产品作为一个成品是不会将仿真器接口预留出来，只会预留出RS232、RS485、USB或以太网接口，只能通过这些接口对产品进行固件升级。

AN2295是飞思卡尔为用户提供的一个UART Bootloader应用笔记。它包括如下文件：



1. Masters 是PC机应用程序，包括代码以及编译好的应用程序。
2. Src是MCU端代码，支持目前大部分飞思卡尔产品。
3. AN2295.pdf 应用笔记的说明文档，根据不同型号的MCU，分章节介绍了UART升级协议、mcu部分的软件流程、pc端软件流程。
4. MCU Bootloader

这里主要介绍Kinetis系列产品：

1. AN2295\_FRDM\_KL25.h: 我们以这个文件为例，该文件主要定义了板子上的资源，移植时需要修改的有以下几个宏：

#define KINETIS\_MODEL KL25\_48MHz //该宏用于区分产品型号

#define BOOTLOADER\_SHORT\_TRIM 1 //

#define KINETIS\_FLASH FLASH\_128K //该宏用于定义flash大小

#define BOOTLOADER\_FLASH\_PROTECTION 0 //该宏用于定义是否使能保护功能

#define BOOT\_WAITING\_TIMEOUT 100 //等待超时

//以下宏用于定义bootloader通讯所用UART，用户需要根据自己板子来做修改

#define BOOT\_UART\_BASE UART0\_BASE\_PTR

#define BOOT\_UART\_BAUD\_RATE 115200

#define BOOT\_UART\_GPIO\_PORT PORTA\_BASE\_PTR

#define BOOT\_PIN\_UART\_ALTERNATIVE 3

#define BOOT\_PIN\_GPIO\_ALTERNATIVE 1

#define BOOT\_UART\_GPIO\_PIN\_RX 15

#define BOOT\_UART\_GPIO\_PIN\_TX 14

//以下宏用于上电读取某管脚电平，当BOOTLOADER\_PIN\_ENABLE为1时，以此判断是直接进入APP模式还是与上位机通讯模式

#define BOOT\_PIN\_ENABLE\_PORT\_BASE PORTC\_BASE\_PTR

#define BOOT\_PIN\_ENABLE\_GPIO\_BASE PTC\_BASE\_PTR

#define BOOT\_PIN\_ENABLE\_NUM 9

//常用功能

#define BOOTLOADER\_ENABLE\_READ\_CMD 1

#define BOOTLOADER\_INT\_WATCHDOG 0

#define BOOTLOADER\_ENABLE\_VERIFY 0

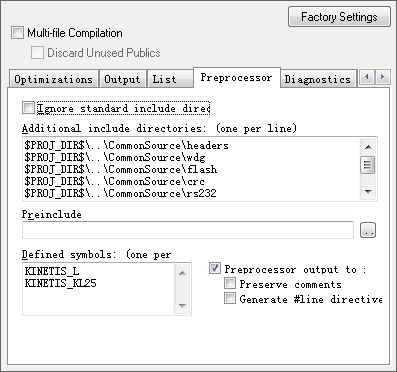
#define BOOTLOADER\_CRC\_ENABLE 0 //是否使能CRC

#define BOOTLOADER\_AUTO\_TRIMMING 1 //是否使能自动TRIM

#define BOOTLOADER\_PIN\_ENABLE 0 //是否读取外部Pin判断进入何种模式。

#define BOOT\_CALIBRATION\_GPIO\_BASE PTA\_BASE\_PTR

1. Bootloader\_cfg.h: 由于Kinetis有多种不同型号，即使相同型号也有不同的工作频率产品，所以该文件用于定义当前工程使用哪个AN2295\_XXX.h文件。
2. Kinetis\_params.h: 这个文件比较重要，它根据一些宏定义，将代码中用到的参数做为区分，这些宏被定义在下图位置：



下面介绍几个可能会修改的宏：

#define FLASH\_PROT\_SECTION

这个宏用于定义APP程序的起始地址，K系列由于有比较大的flash，所以默认使用其空间的1/32，K60为0x4000，这样的话0x0000~0x4000都为Bootloader空间，但是对于flash比较小的MCU，比如KL25等，我们将该宏定义为0x1000，主要考虑到Bootloader目前编译后需要占用大于2K空间（Release版本可能小于2K）

#define FLASH\_ERASE\_PAGE

这个宏用于定义flash的page大小，不同型号的MCU，大小不同，可以从RM手册的第三章找到对应大小。

#define BOOTLOADER\_PIN\_ENABLE\_INIT

这个宏用于初始化判断是否进入bootloader模式的Pin管脚，如果硬件电路上该管脚是直接接到地，那么需要内部上拉，则需要添加PORT\_PCR\_PS\_MASK，否则管脚不会读取到高电平。

#define BOOT\_BUS\_CLOCK

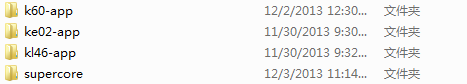
这个宏用于定义时钟频率。

#define INIT\_CLOCKS\_TO\_MODULES

这个宏是用于初始化接口时钟的宏函数

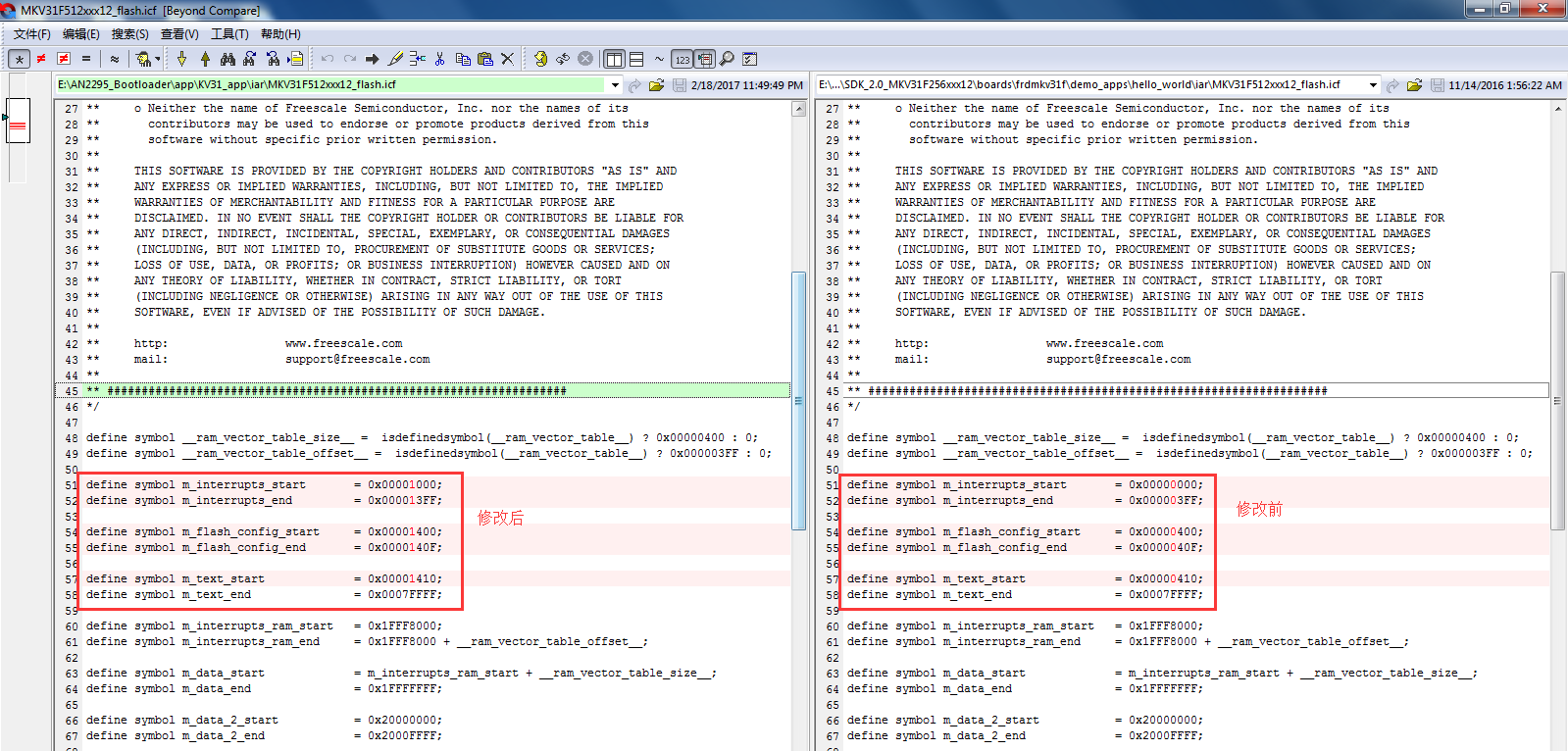
1. 协议简介：AN2295大致流程为，MCU根据PIN或者APP地址是否有工程来判断是否进入bootloader模式，当进入bootloader后，将不断的从指定UART发送0xFC，当PC机软件点击Connect后，并且接收到MCU发送的0xFC后，也会发送0xFC下去，MCU接收到主机发送的0xFC后，会继续发送一个0xFC，并跳转到下一个状态机，去等待上位机发送命令。具体流程，可以参考第四章。
2. 裸板APP应用程序的设计

为方便用户学习，这里提供了一些制作好的app应用程序：



这些程序已经在TWR或者FRDM板上通过验证。

应用程序主要需要修改的是lcf文件，让应用程序保留Bootloader的空间：



1. MQX应用程序设计

MQX是飞思卡尔提供的开源实时操作系统，系统本身没有提供Bootloader的功能，但是通过修改链接文件或者源代码，同样可以使用AN2295作为bootloader来引导MQX，具体做法如下：

由于编译后的库文件比较大，如果需要例子可以与我联系。

1. 通讯示例：

KL46通讯截图：

8 [00000018] IRP\_MJ\_READ Length: 0001, Data: FC

9 [00000019] IRP\_MJ\_WRITE Length: 0001, Data: FC

10 [00000019] IRP\_MJ\_READ Length: 0001, Data: FC

11 [00000019] IRP\_MJ\_WRITE Length: 0001, Data: 49

12 [00000019] IRP\_MJ\_READ Length: 0001, Data: C8

13 [00000019] IRP\_MJ\_READ Length: 0001, Data: 05

14 [00000019] IRP\_MJ\_READ Length: 0001, Data: 88

15 [00000019] IRP\_MJ\_READ Length: 0004, Data: 00 00 00 01

16 [00000019] IRP\_MJ\_READ Length: 0004, Data: 00 00 10 00

17 [00000019] IRP\_MJ\_READ Length: 0004, Data: 00 01 FF FF

18 [00000019] IRP\_MJ\_READ Length: 0004, Data: 00 00 10 00

19 [00000019] IRP\_MJ\_READ Length: 0004, Data: 00 00 00 00

20 [00000019] IRP\_MJ\_READ Length: 0004, Data: 00 00 04 00

21 [00000019] IRP\_MJ\_READ Length: 0004, Data: 00 00 00 80

22 [00000019] IRP\_MJ\_READ Length: 0001, Data: 78

23 [00000019] IRP\_MJ\_READ Length: 0001, Data: 78

24 [00000019] IRP\_MJ\_READ Length: 0001, Data: 78

25 [00000019] IRP\_MJ\_READ Length: 0001, Data: 00

26 [00000019] IRP\_MJ\_READ Length: 0002, Data: DB 17

参考文档：

1. AN2295
2. Kinetis RM手册