



使用U盘升级程序的解决方案

Let`s make your development easier!

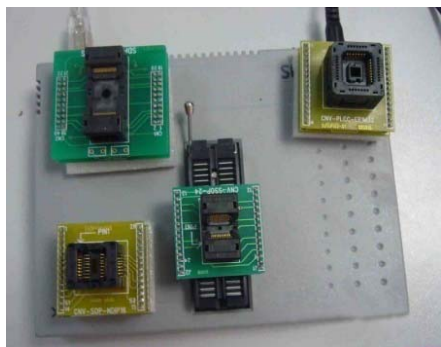
捷恩斯威科技，最专业的TI MCU方案设计商

www.jeansway.cn



传统的软件升级方式

编程器烧录方式:



JTAG下载方式:

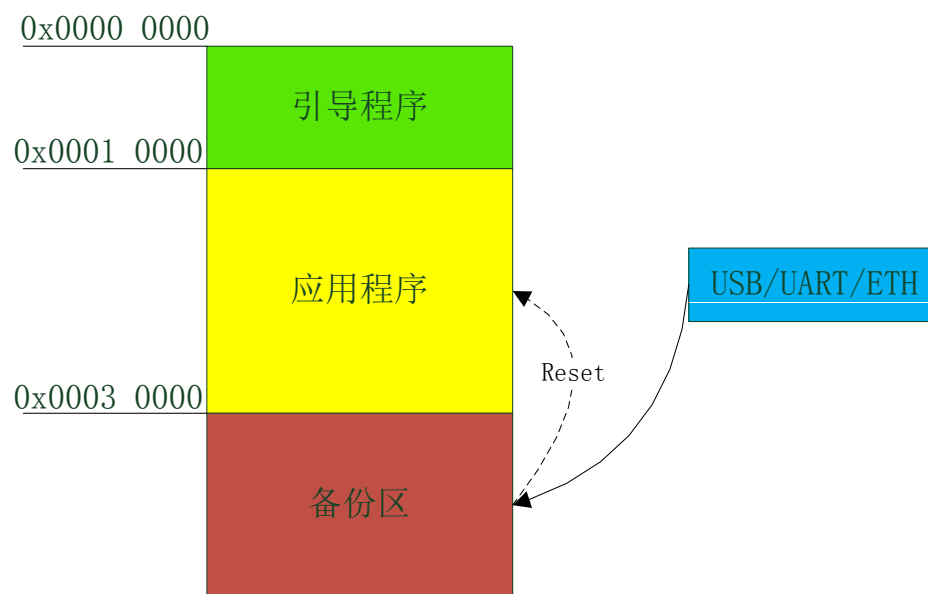


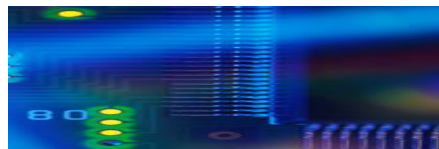
串口下载方式:





通信模式升级方式





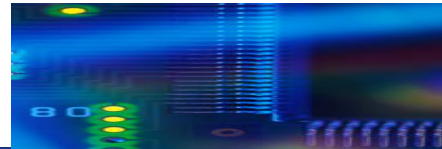
使用U盘升级的优点

- 1 携带方便，利于离线性设备更新程序
- 2 无需上位机软件参与
- 3 避免复杂的通信协议定制
- 4 不用担心更新程序时，掉电等异常情况发生，而导致设备损坏
- 5 可与设备的USB接口共用接口



TI Cortex-M4 目前支持USB系列型号

Cortex-M4	USB类型	备注
LM4F11xXX	N/A	LM4F110/111/112
LM4F12xXX	D	LM4F120/121/122
LM4F13XXX	D/H/OTG	LM4F130/131/132
LM4F23xxx	D/H/OTG	LM4F230/231/232

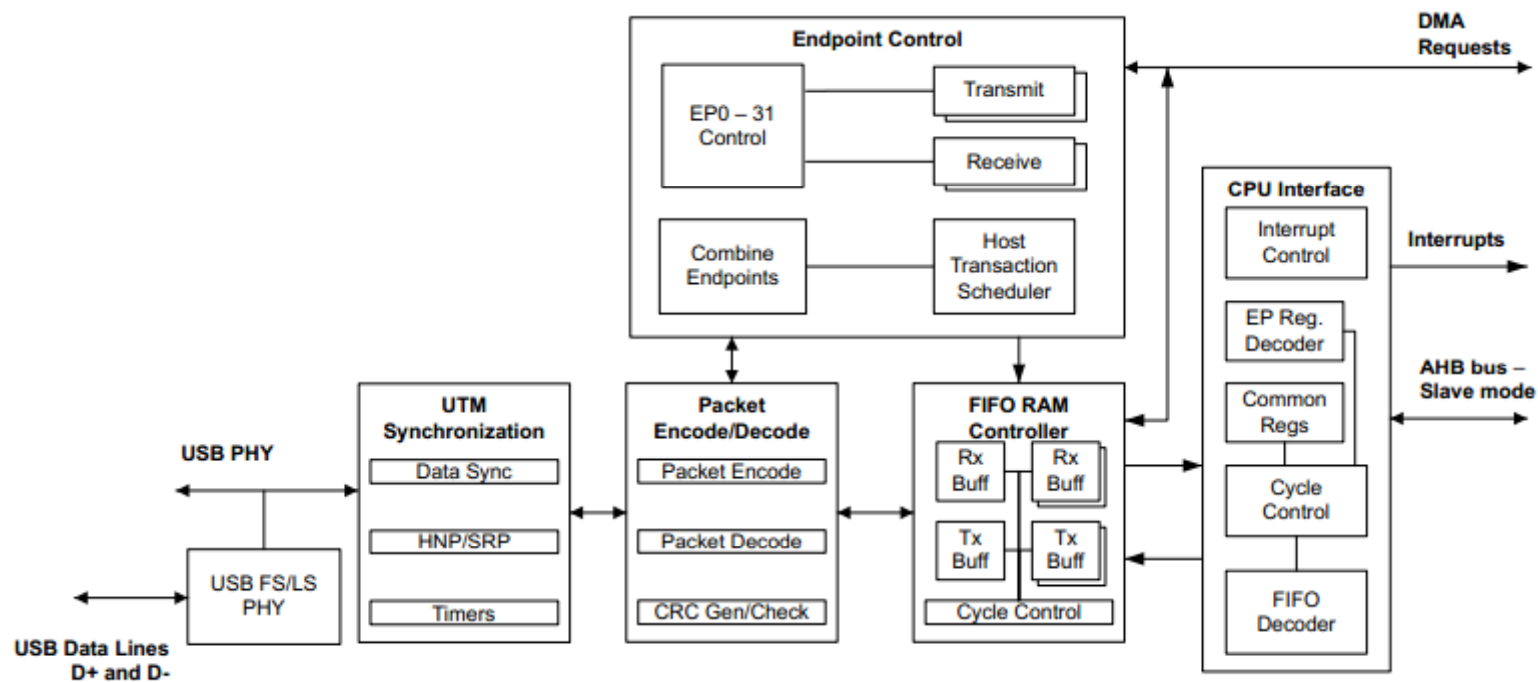


TI Cortex-M4 USB硬件特性

- ◆ 支持USB 2.0全速(12 Mbps)和低速(1.5 Mbps)
- ◆ 4类传输模式: 控制传输, 中断传输, 批量传输, 等时传输
- ◆ 16个端点:
 - 1个专用的输入控制端点和1个专用输出控制端点
 - 7个可配置的输入端点和7个可配置的输出端点
- ◆ 2KB专用端点内存空间:可支持双缓冲的1023字节最大包的等时传输
- ◆ 支持VBUS 电压浮动和有效ID检测, 并产生中断信号.
- ◆ 高效传输 μ DMA :3个输入端点和3个输出端点独立发送和接收, 当FIFO中包含需要的大量数据时, 触发通道请求



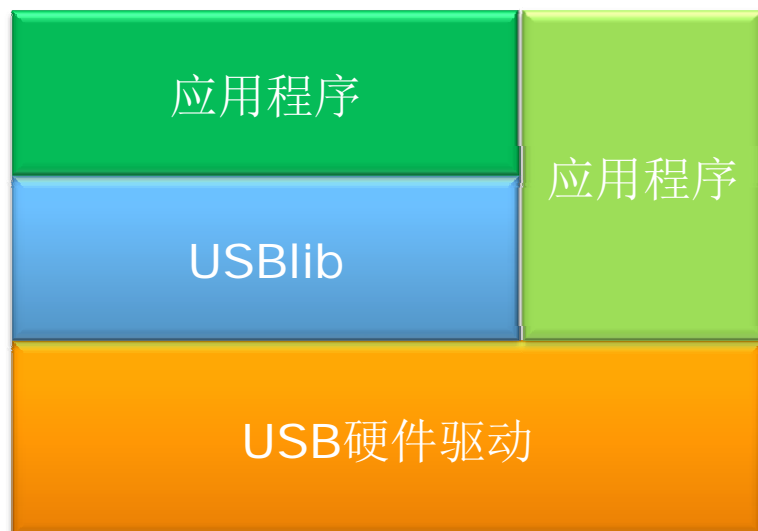
TI Cortex-M4 USB 模块图



系统时钟必须工作在20MHz以上
管脚USB0RBIAS和地之间需要接一个1%精度的9.1 kΩ电阻.



UsbLib软件库，助你快速完成U盘升级功能

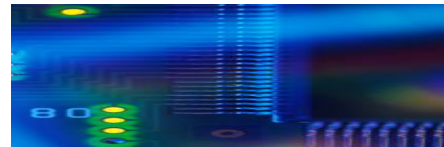


❖ TI USB软件库

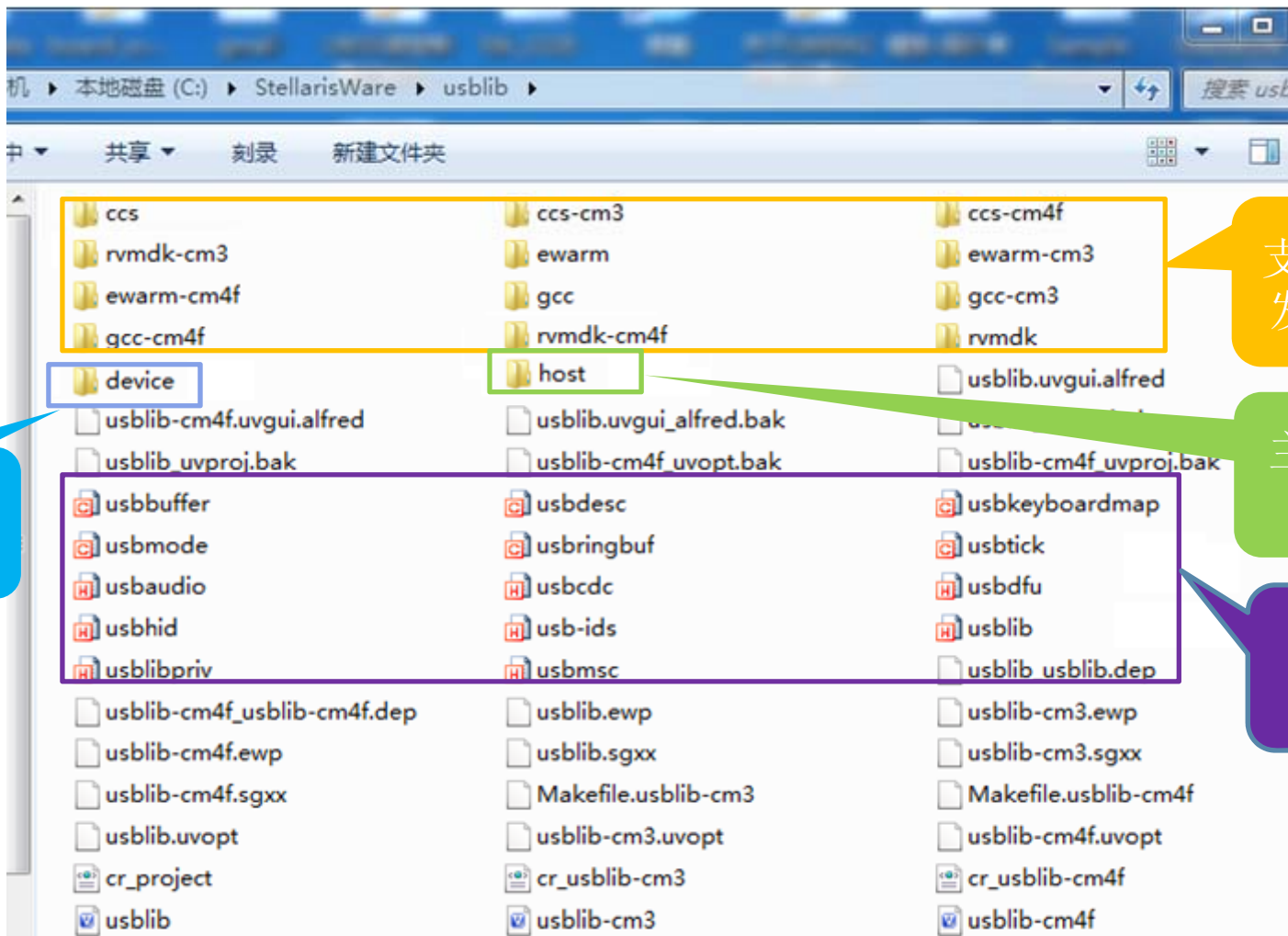
- * Bulk Device CDC Device Class
- * HID Device Class
- * Mass Storage Device Class
- * USB Host Class
- * HID Host Class
- * Mass Storage Host Class

❖ Jeansway USB软件库

- * USB Printer Class



Usblib 目录结构

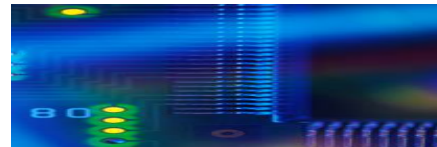


设备型
API函数

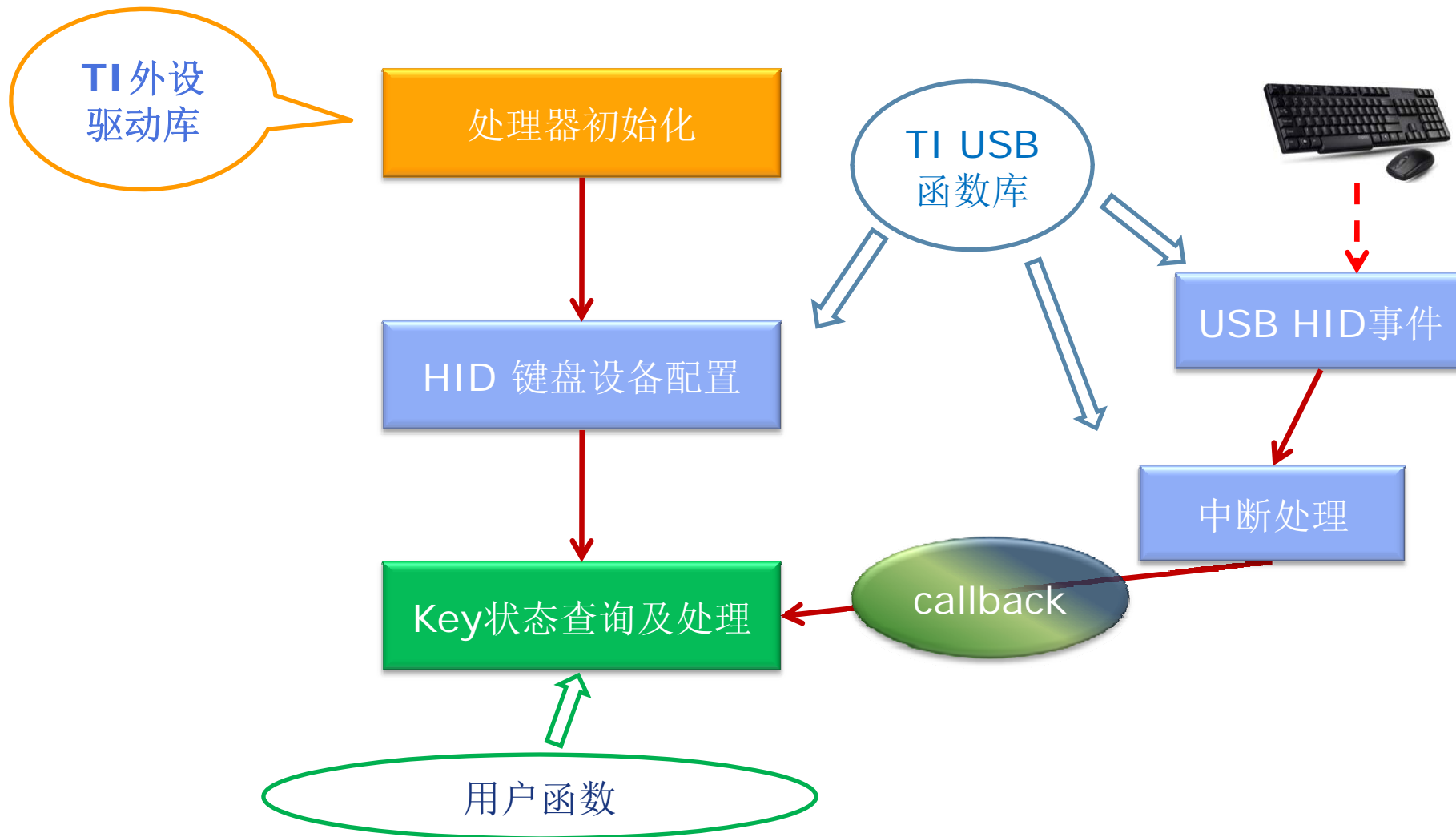
支持各种开
发环境LIB

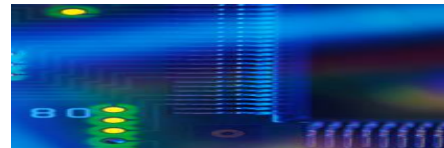
主机型API
函数

基本操作
API函数

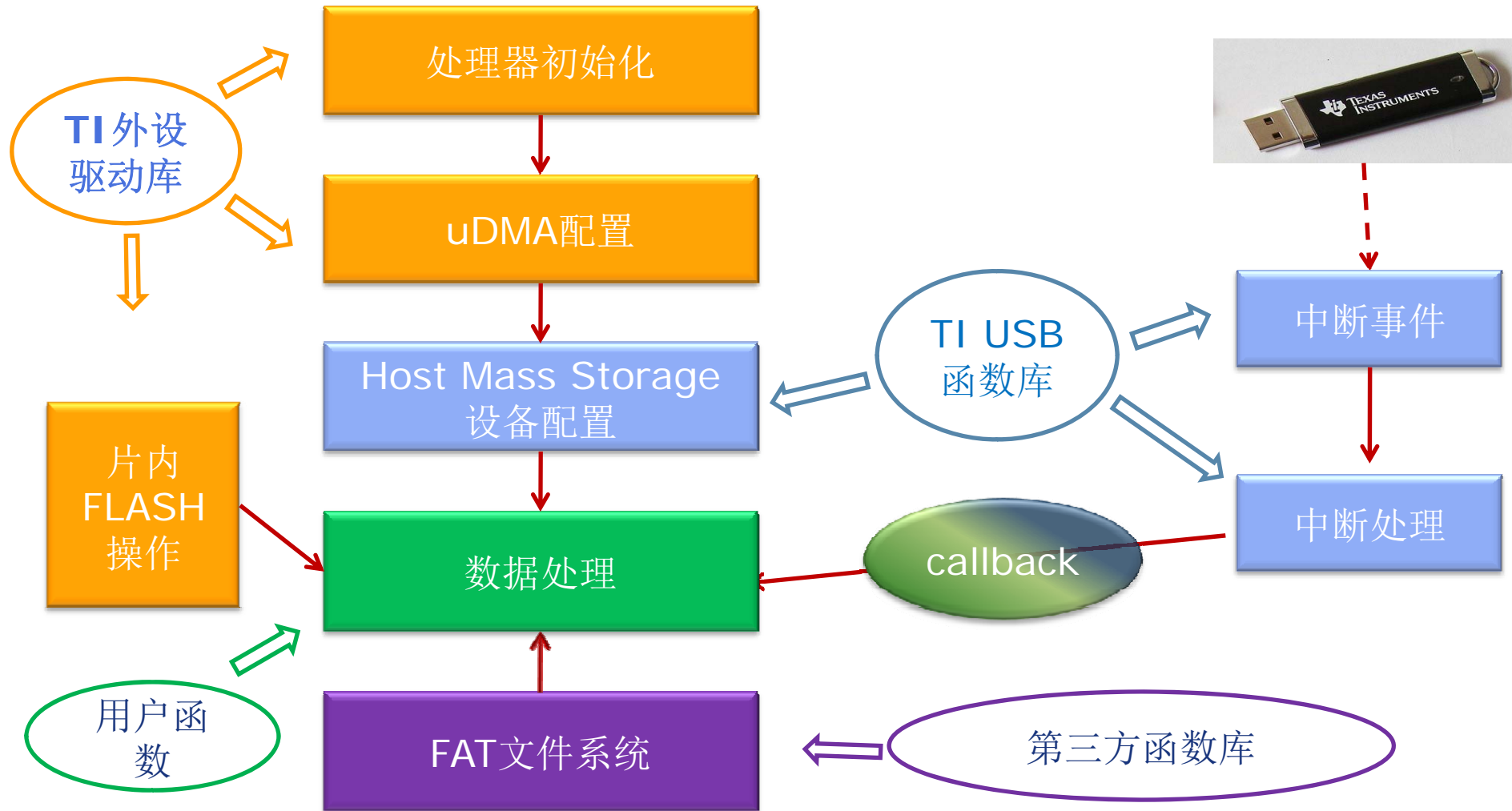


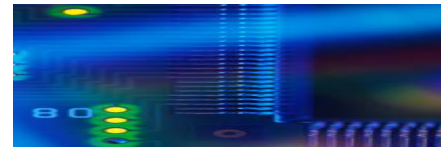
HID设备键盘模型



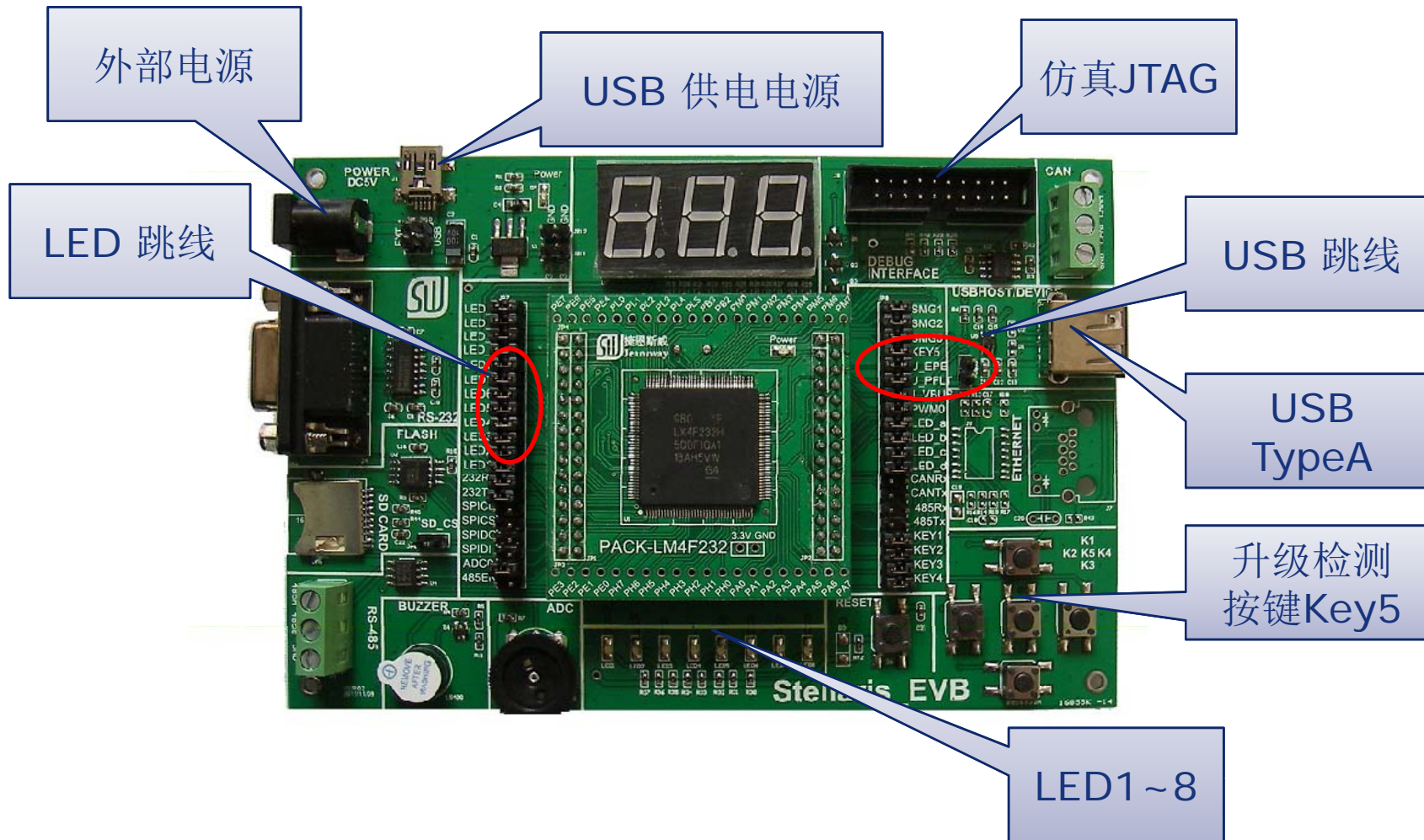


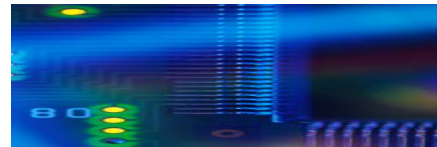
Mass Storage usb 升级模型



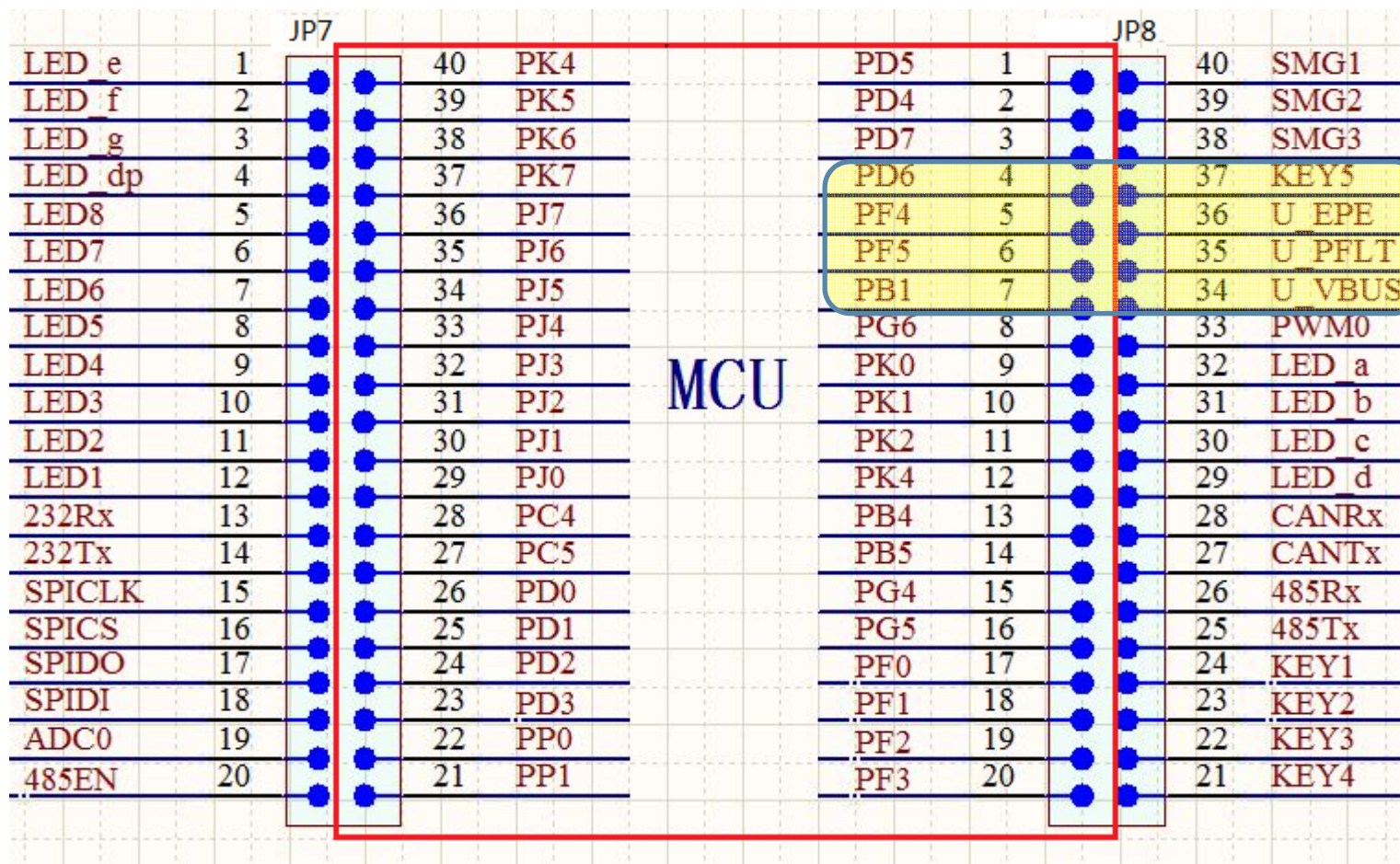


JSW-M4EVB开发板USB硬件连接





JSW-M4EVB 外设接线图





DEMO使用到的MCU资源

❖ key5:

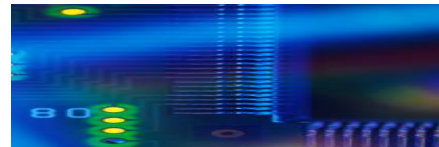
GPIO PortD6

❖ USB:

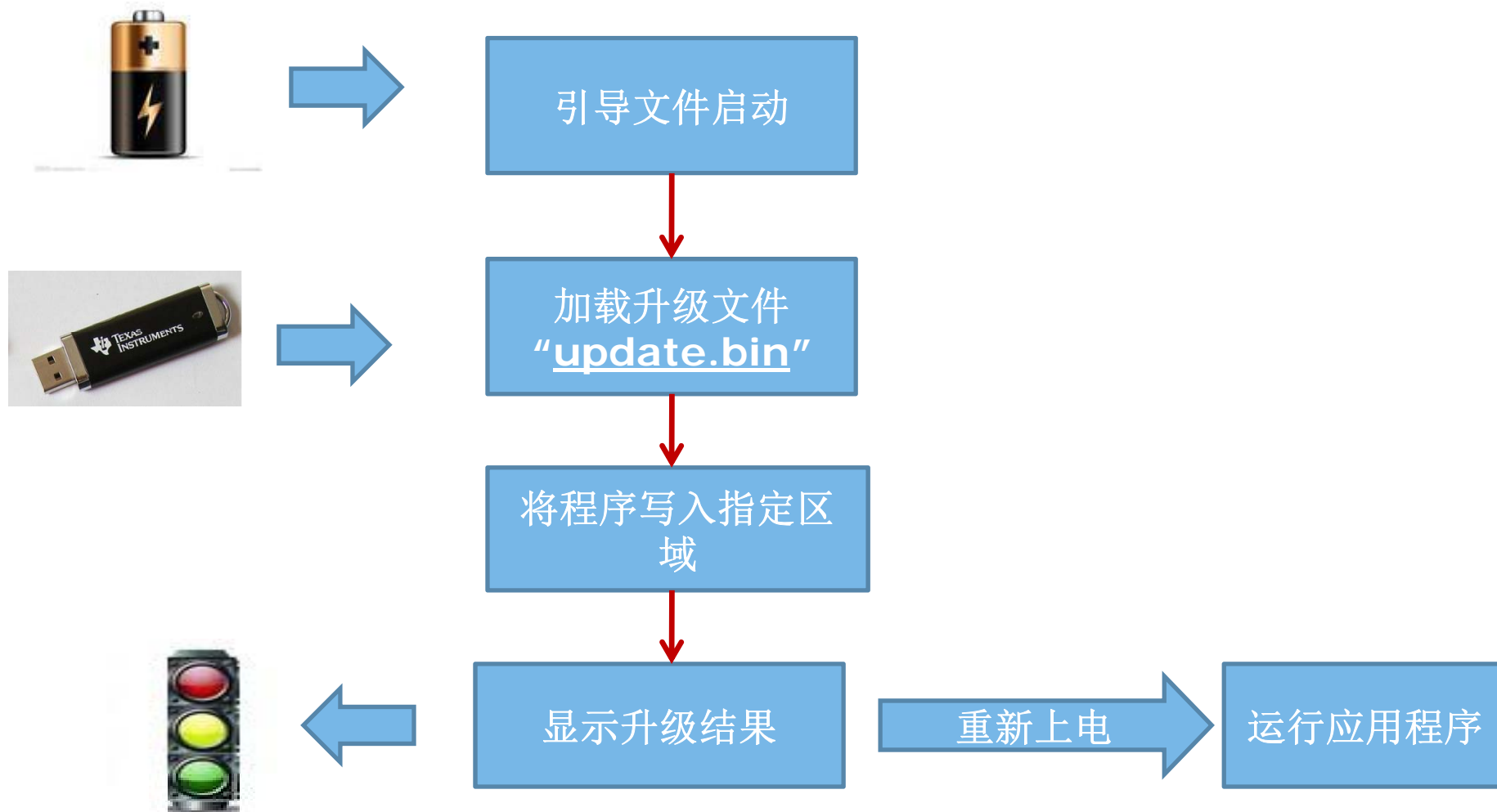
信号	引脚
D+	PL6
D-	PL7
VBUS	PB1
EPE	PF4
PFLT	PF5

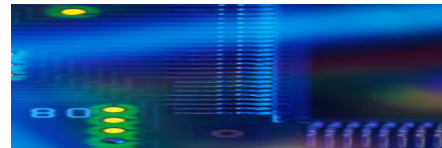
❖ LED:

Led1~Led8: PJ0~PJ7



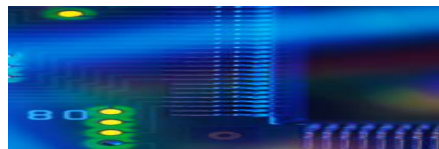
JSW USB在线升级实现流程



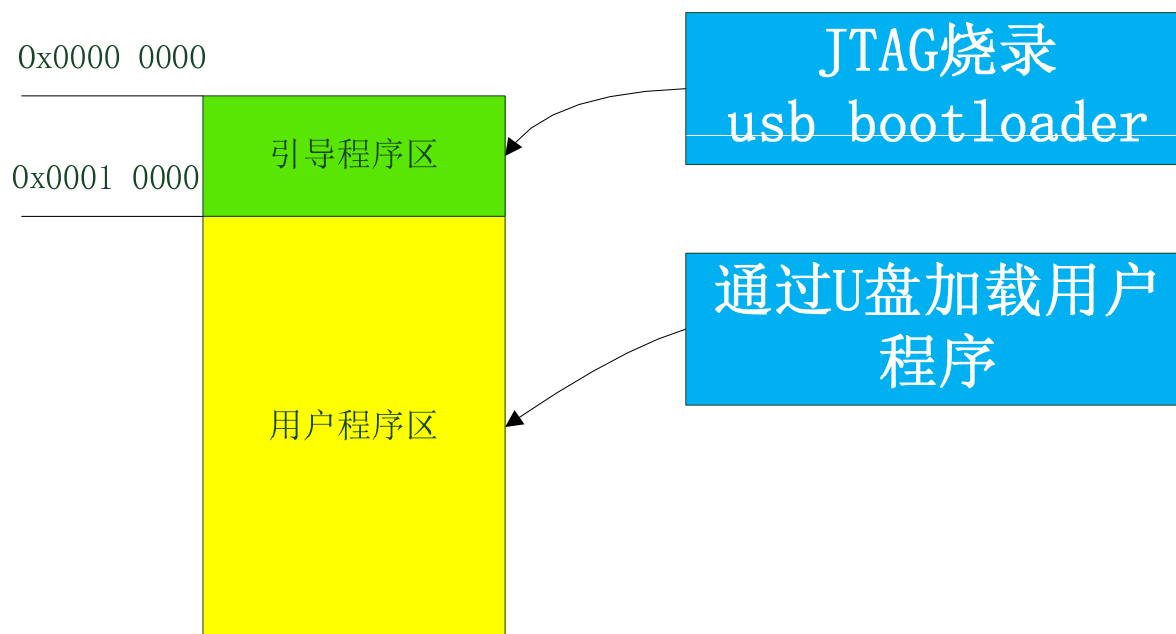


U盘升级进行步骤

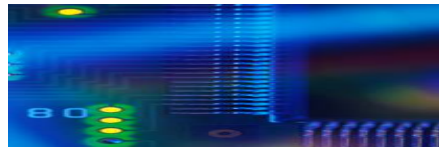
- Step1: 程序规划
- Step2: 指定Bootloader程序空间
- Step3: 编写USB bootloader程序
- Step4: 编译&下载bootloader
- Step5: 指定应用程序编译地址
- Step6: 编写用户程序
- Step7: 升级用户程序
- Step8: 检查升级结果



Step1: 程序规划 (1/2)

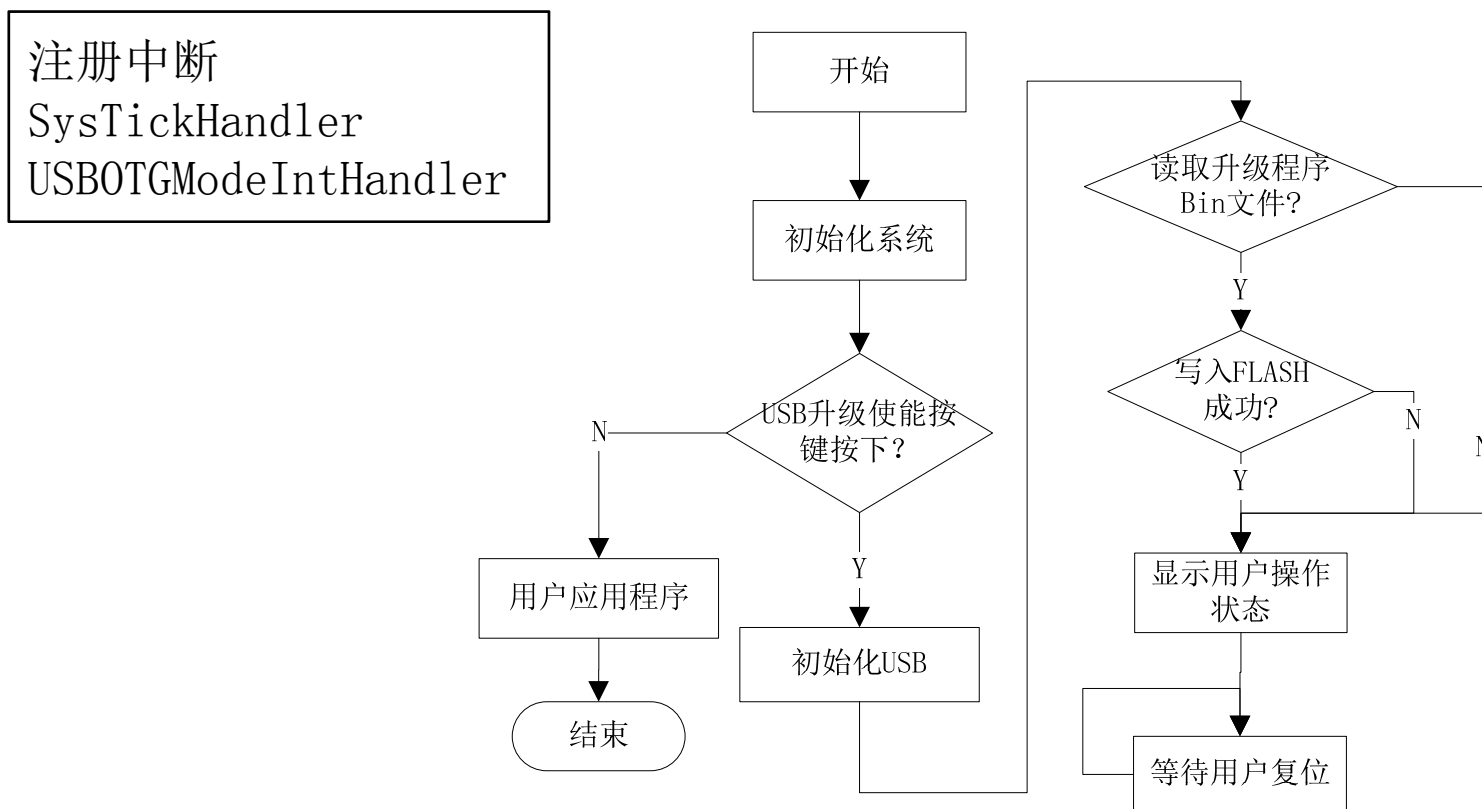


U盘升级方案程序空间分配



Step1: 程序规划 (2/2)

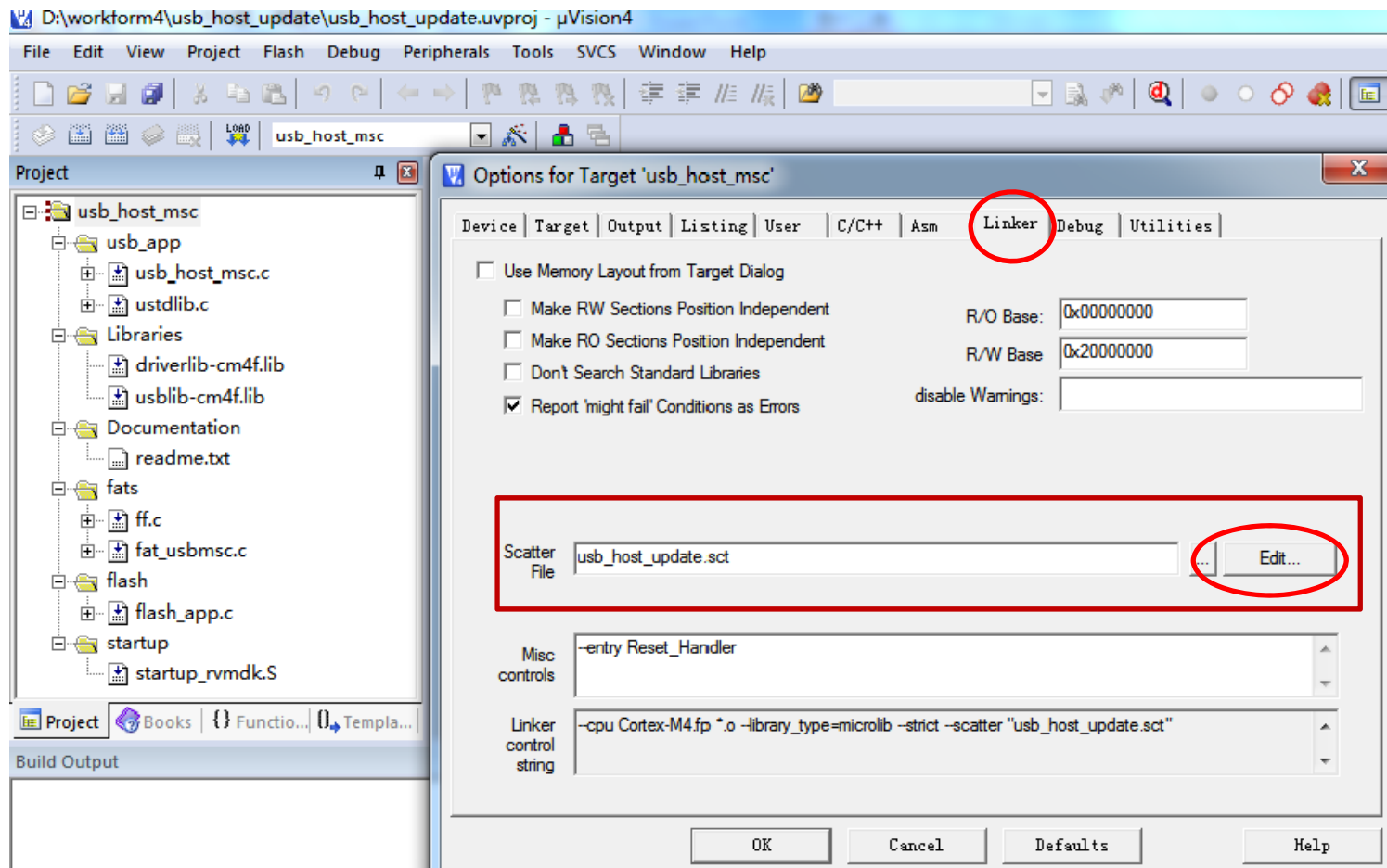
程序流程图

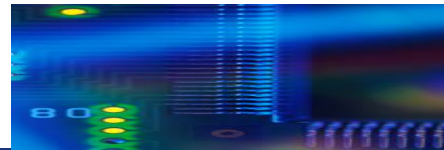




Step2: 指定Bootloader程序空间 (1/2)

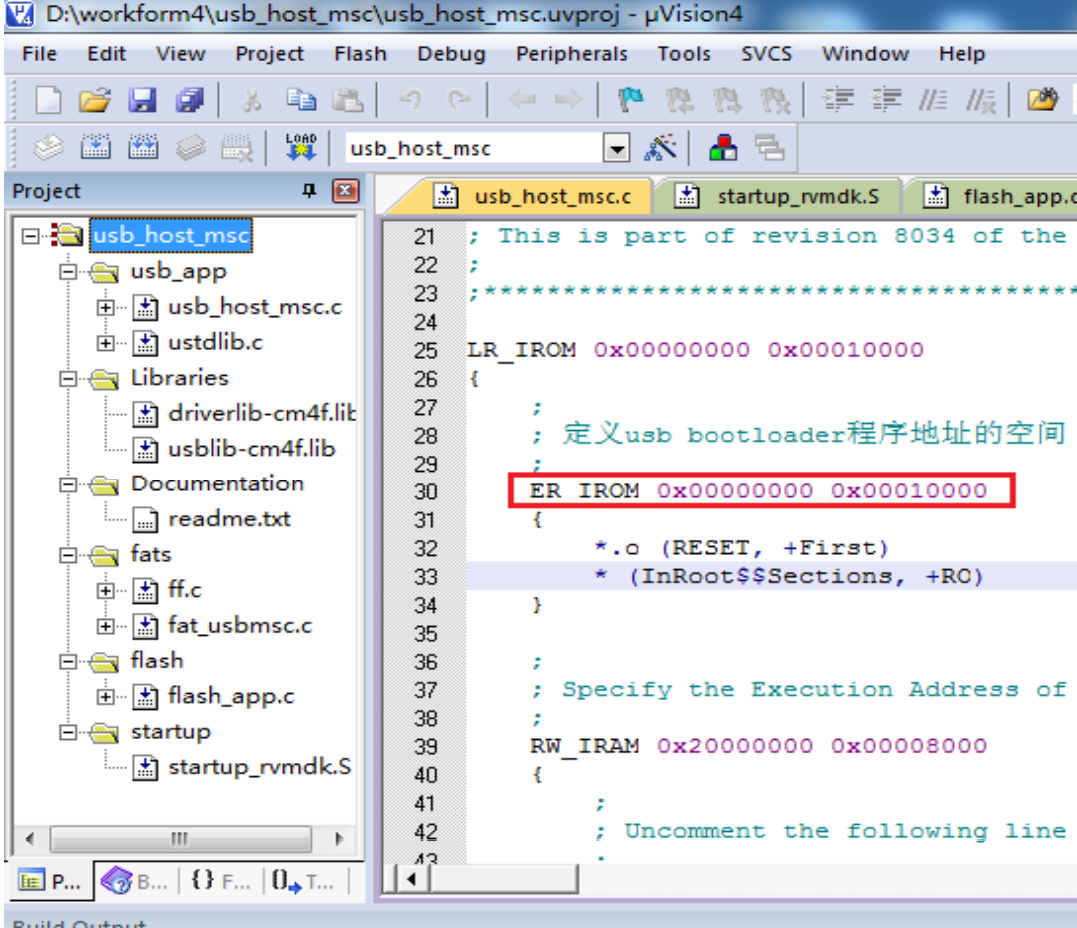
打开USBBootloader工程(usb_host_update.uvproj), 设置分散加载配置文件





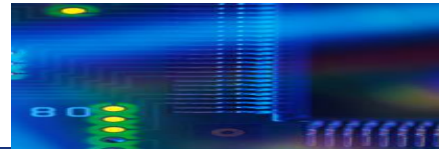
Step2: 指定Bootloader程序空间 (2/2)

设定Bootloader地址0x00000000~0x00010000



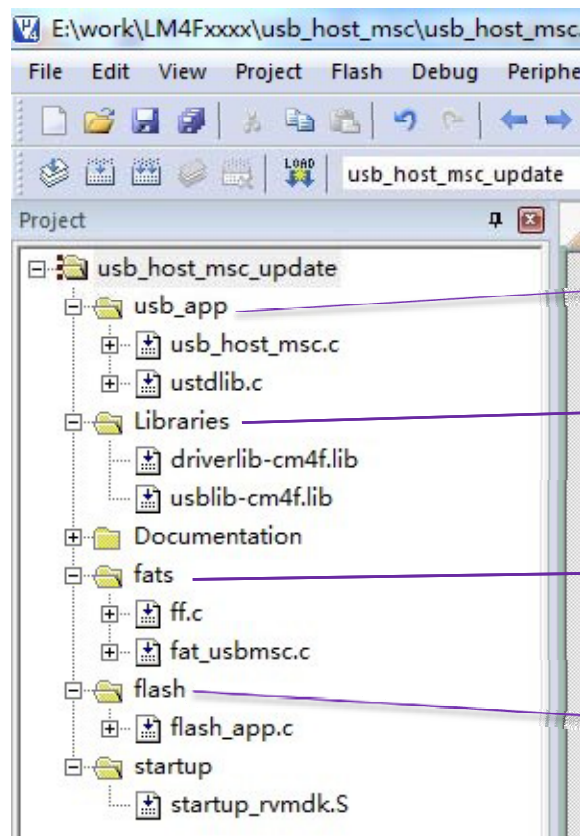
```
21 ; This is part of revision 8034 of the
22 ;
23 ;*****
24
25 LR_IROM 0x00000000 0x00010000
26 {
27 ;
28 ; 定义usb bootloader程序地址的空间
29 ;
30 ER IROM 0x00000000 0x00010000
31 {
32 ;
33 ; *.o (RESET, +First)
34 ; * (InRoot$$Sections, +RO)
35 }
36 ;
37 ; Specify the Execution Address of
38 ;
39 RW_IRAM 0x20000000 0x00008000
40 {
41 ;
42 ; Uncomment the following line
43 ;
```





Step3: 编写USB bootloader程序 (1/2)

Bootloader工程文件



Bootloader
主程序

Usblib
driverlib

FAT文件系
统

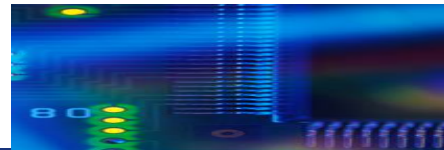
FLASH操作



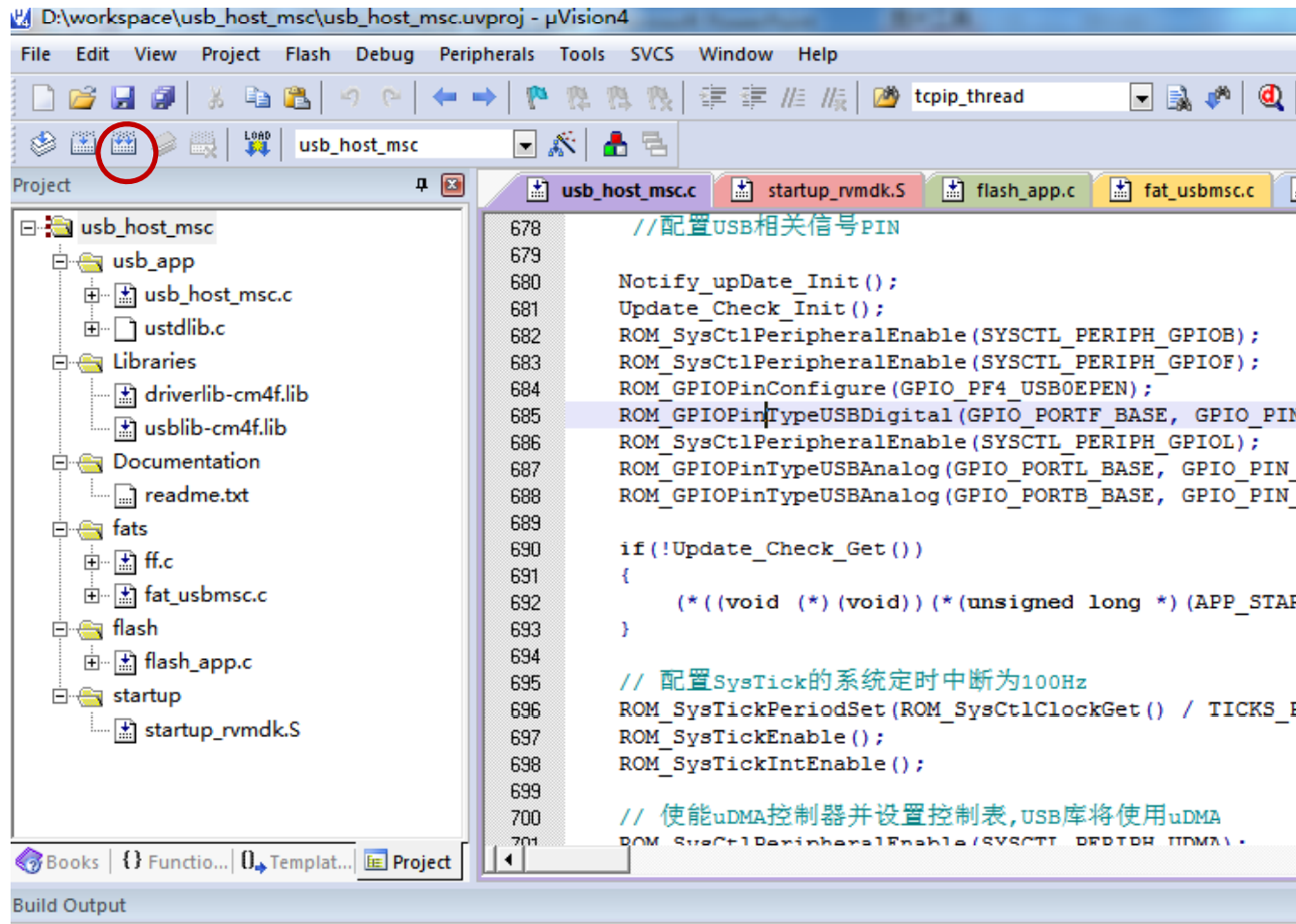
Step3: 编写USB bootloader程序 (2/2)

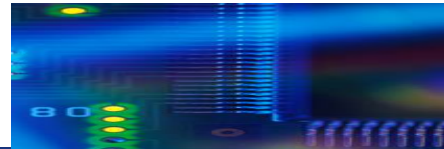
Bootloader 使用的USB库API

- ❖ 初始化HOST模式USB协议栈
`USBStackModeSet(0, USB_MODE_HOST, 0);`
- ❖ 注册主设备类驱动
`USBHCDRegisterDrivers(0, g_ppHostClassDrivers, g_ulNumHostClassDrivers);`
- ❖ 打开一个批量存储类设备驱动实例
`USBHMSCDriveOpen(0, MSCCallback);`
- ❖ 初始化电源配置
`USBHCDPowerConfigInit(0, USBHCD_VBUS_AUTO_HIGH |
USBHCD_VBUS_FILTER);`
- ❖ 初始化主设备操作的USB控制器
`USBHCDInit(0, g_pHCDPool, HCD_MEMORY_SIZE);`
- ❖ OTG模式中中断函数
`USBOTGModeIntHandler()`



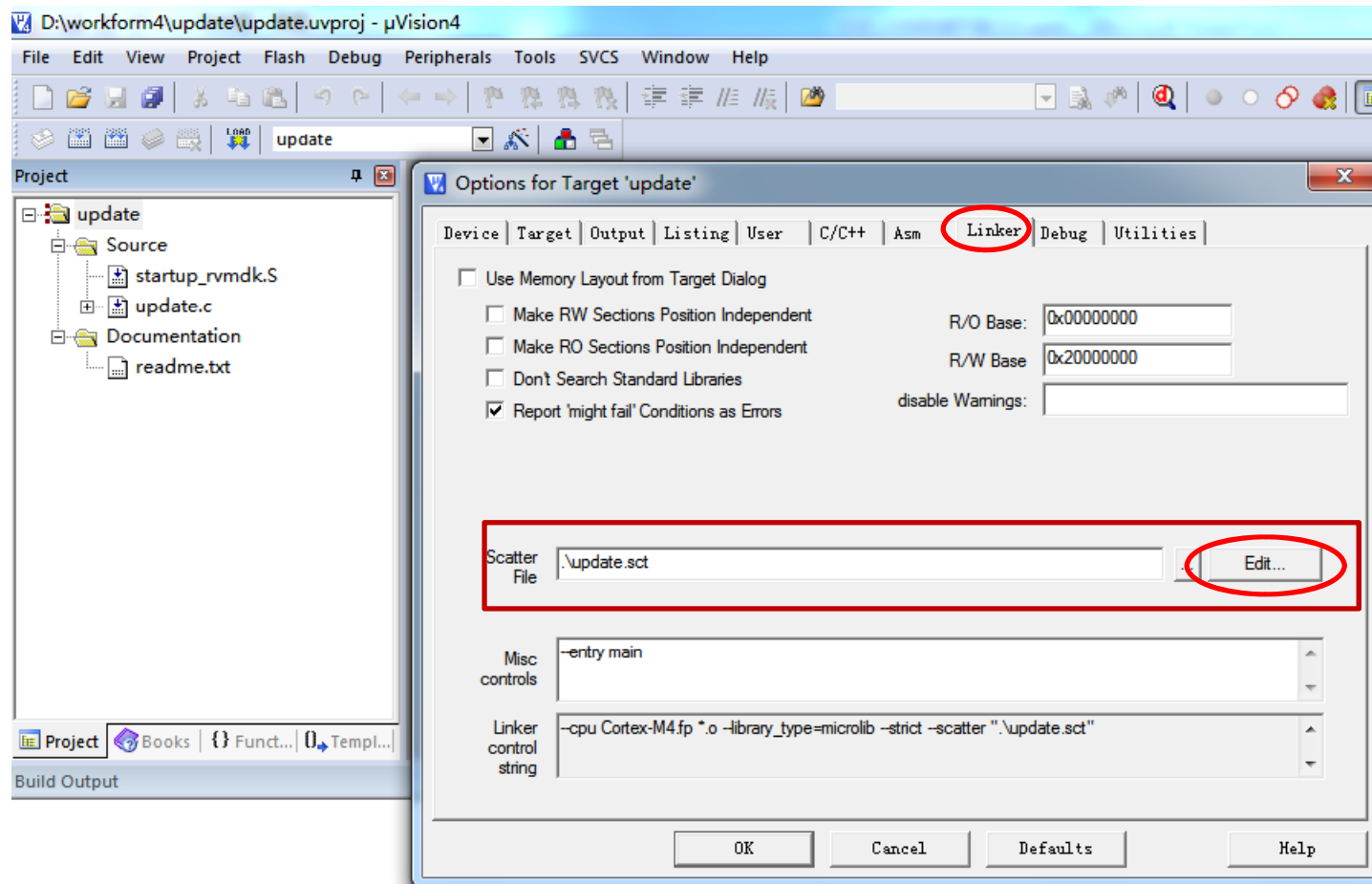
Step4: 编译&下载Bootloader程序

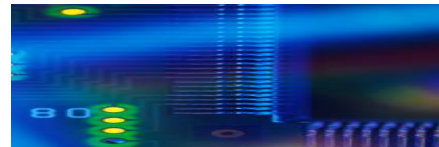




Step5: 指定用户程序编译地址 (1/2)

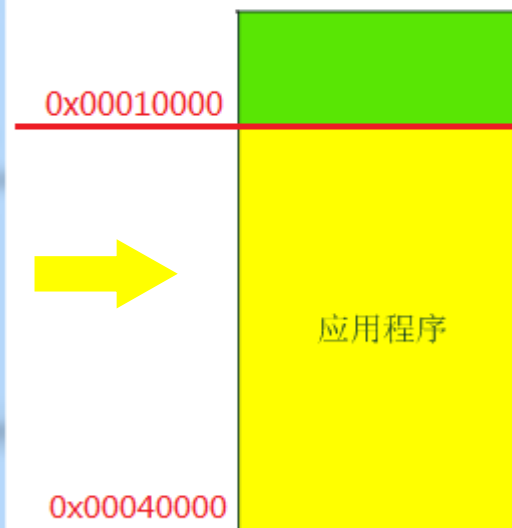
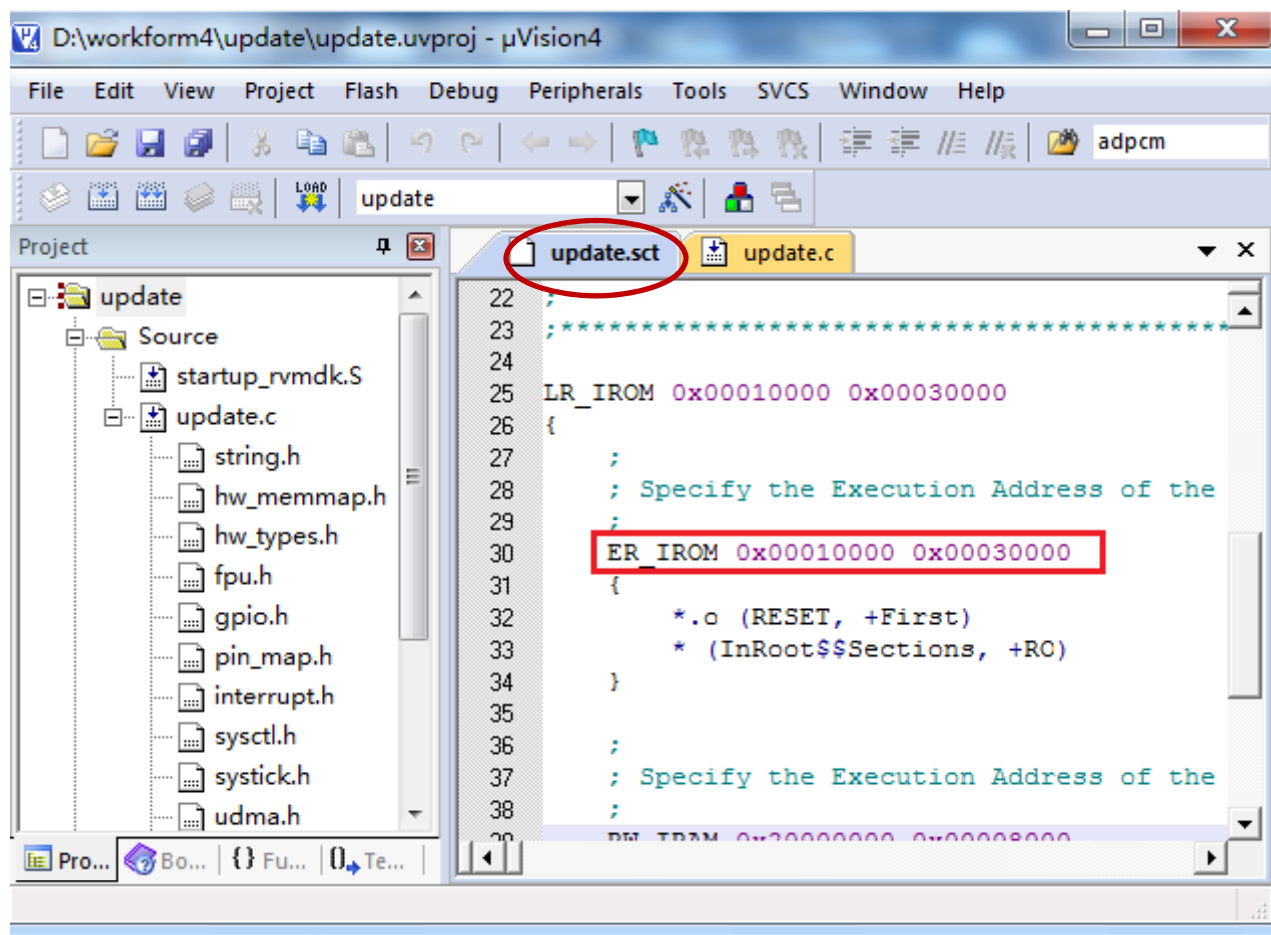
打开用户程序工程(update .uvproj),设置分散加载文件





Step5: 指定用户程序编译地址 (2/2)

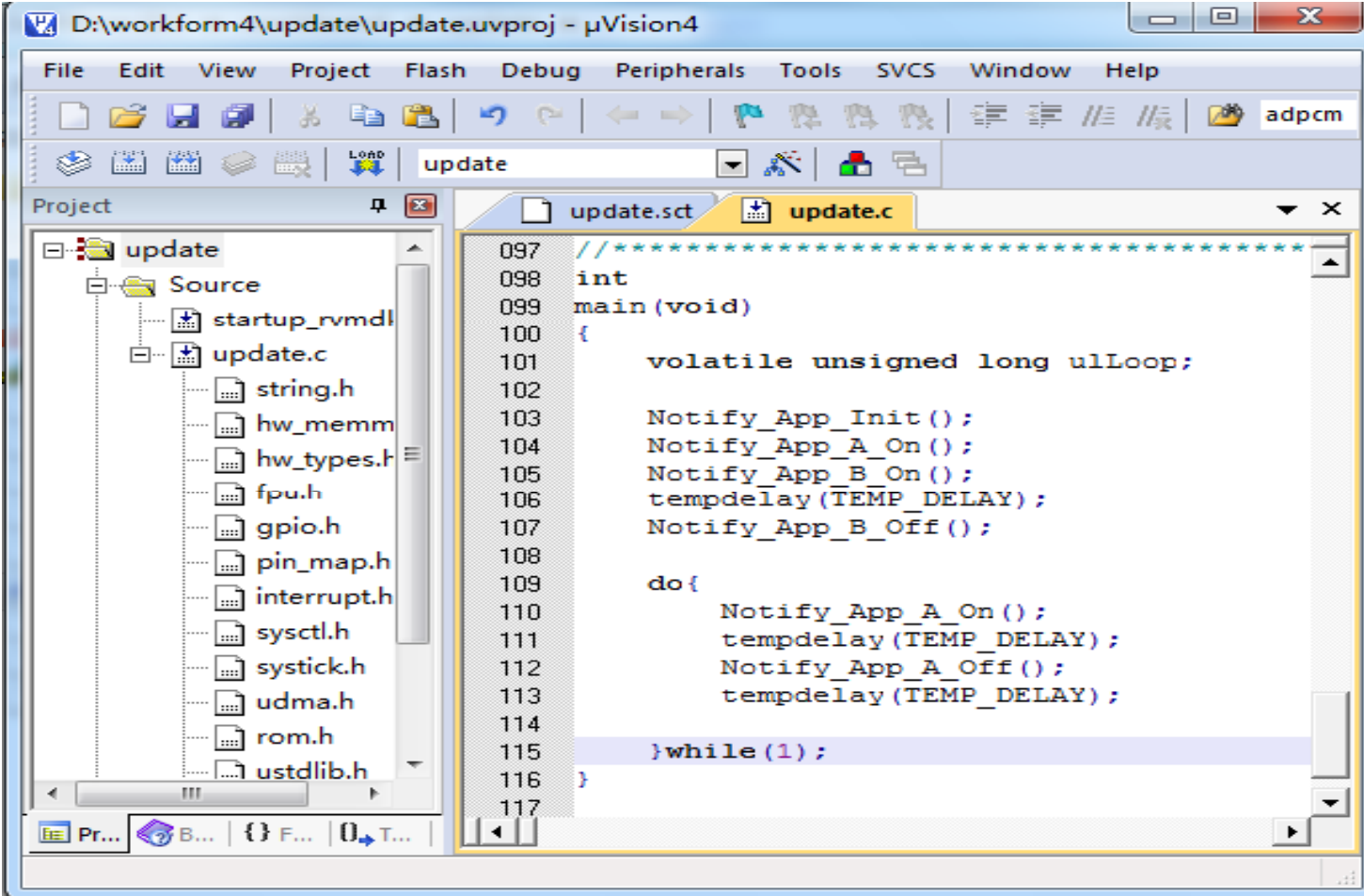
设定用户程序地址: 0x00010000~0x00040000





Step6: 编写用户程序

编译用户程序update, 生成update.bin文件

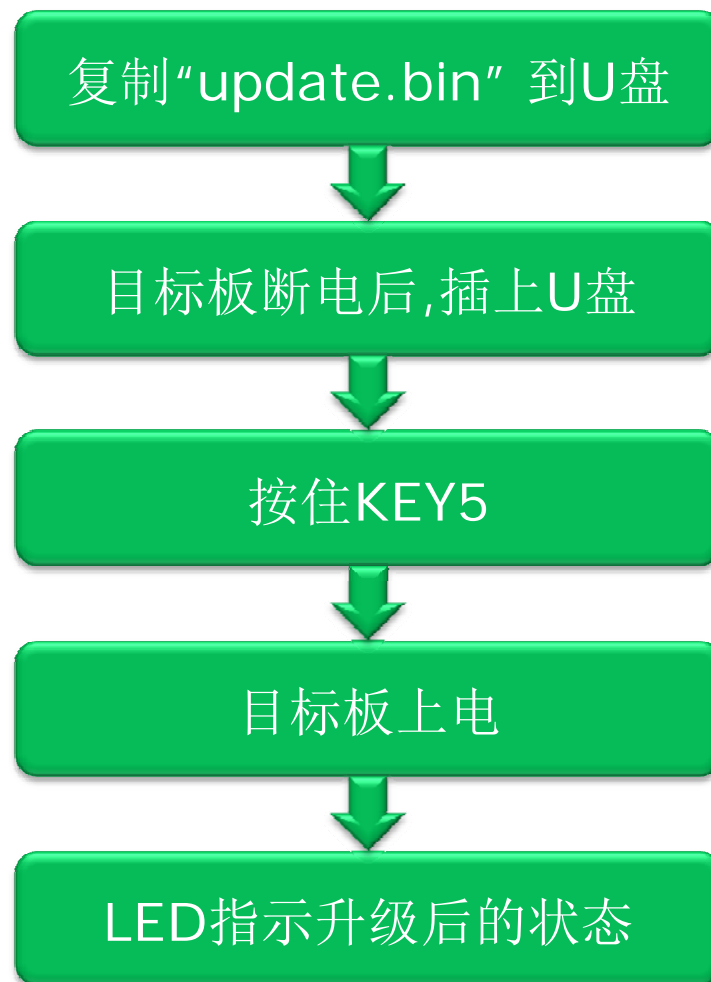


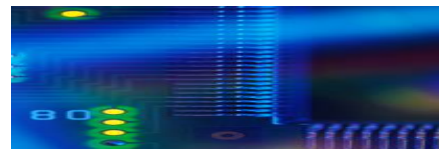
The screenshot shows the µVision4 IDE interface. The Project window on the left displays the project structure for 'update', including a 'Source' folder with files like 'startup_rvmdl', 'update.c', and various header files. The main editor window shows the source code for 'update.c' with the following content:

```
097 //*****
098 int
099 main(void)
100 {
101     volatile unsigned long ulLoop;
102
103     Notify_App_Init();
104     Notify_App_A_On();
105     Notify_App_B_On();
106     tempdelay(TEMP_DELAY);
107     Notify_App_B_Off();
108
109     do{
110         Notify_App_A_On();
111         tempdelay(TEMP_DELAY);
112         Notify_App_A_Off();
113         tempdelay(TEMP_DELAY);
114
115     }while(1);
116 }
117
```



Step7: 升级用户程序





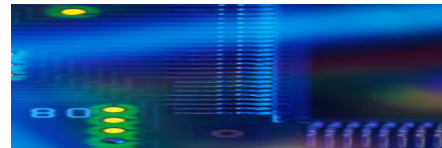
Step8: 检查升级结果

❖ LED指示升级状态

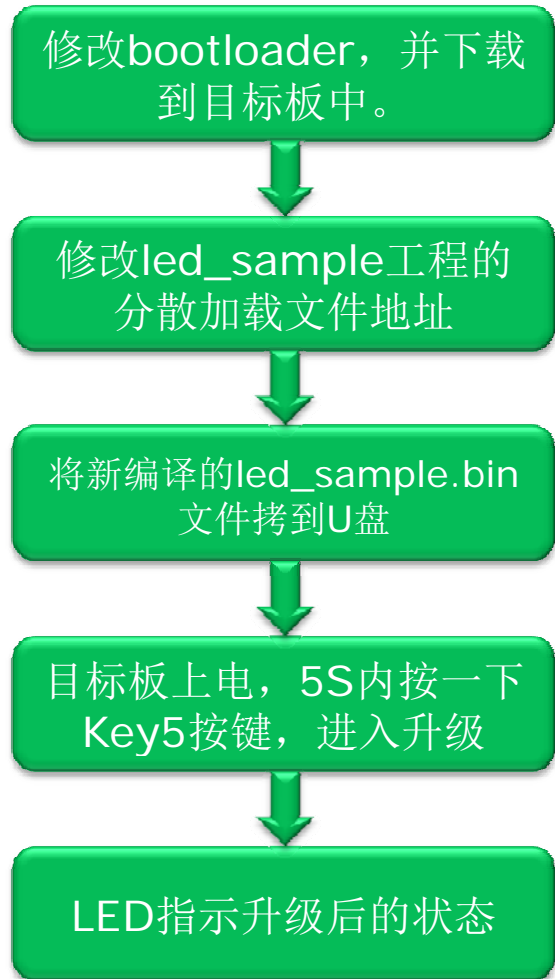
LED编号	点亮	熄灭
LED1	成功	失败
LED2	打开目录失败	打开目录成功
LED3	打开文件失败	打开文件成功
LED4	获取文件状态失败	获取文件状态成功
LED5	文件超过最大长度	文件长度符合要求
LED6	读取文件失败	读取文件成功
LED7	写入FLASH失败	写入FLASH成功
LED8	未使用	未使用

❖ 用户程序运行

升级用户程序成功后，LED1~LED4 闪烁

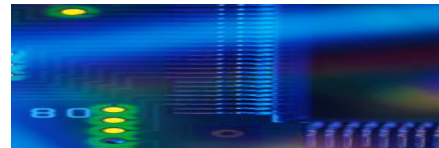


实际操作



1. 修改bootloader中进入升级烧录模式的按键方式：由原来的长按Key5不放上电的方式，改为上电5秒内单按一下Key5键就可以进入升级烧录模式。
2. 将上午的LED工程（led_sample.uvproj），将其分散加载文件改成0x10000地址开始。通过U盘升级方式加载到开发板中
3. 检验升级后的用户程序运行是否正确。

完成后请马上举手示意，最快完成者，经现场确认合格后可获得奖品一个。



在线技术支持

<https://www.deyisupport.com/>

https://www.deyisupport.com/question_answer/f/57.aspx



德州仪器在线技术支持社区
www.deyisupport.com
TEXAS INSTRUMENTS

主页 合作伙伴 咨询专家 大学 社交媒体 登录 / 注册

首页 » 咨询专家 » 基于 Stellaris® ARM® Cortex™-M3 的 MCU

输入搜索关键字 搜索

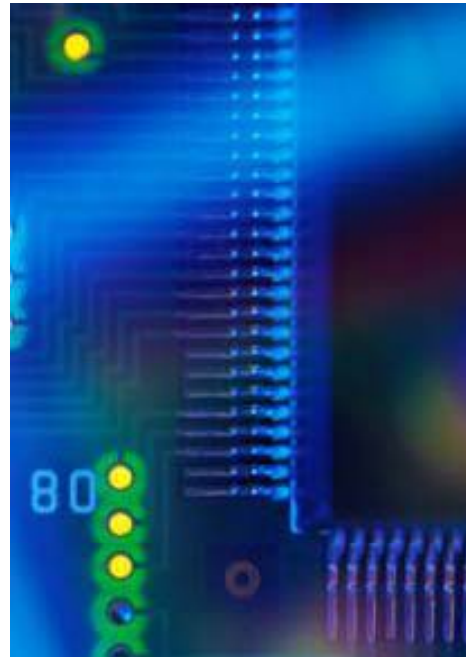
高级搜索

咨询专家

- 模拟与混合信号
- 放大器
- 数据转换器
- 接口时钟
- 无线连接
- 电源管理

如果您有问题需要解答, 请点击此链接去发表新帖子。 发表新帖

分享到...



Thank You !

www.jeansway.cn