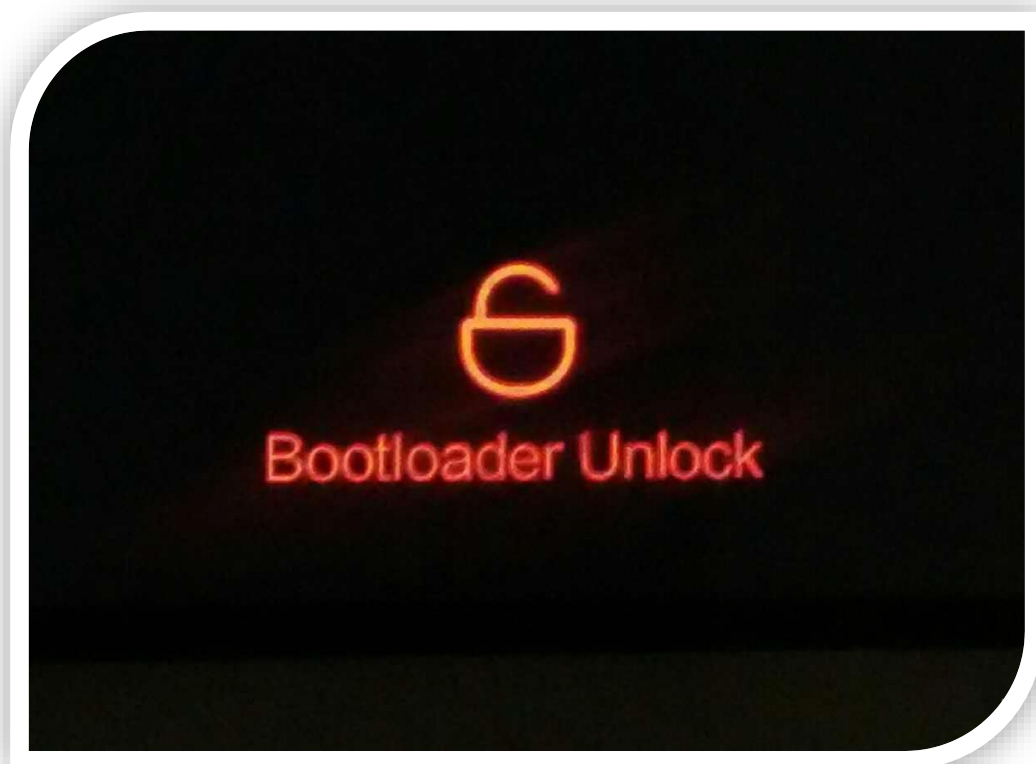




BootLoader 软件说明书

BootLoader software specification



湖南宏迅亿安新能源科技有限公司

编写：文明

2019 年 1 月 17 日



目录

§1 BOOTLOADER 设计说明.....2

 BootLoader 的设计目的及思路2

 BootLoader 中 CAN 的使用说明2

 App 和 BootLoader 中相关程序修改的说明3

 BootLoader 下载中返回的状态值6

§2 BOOTLOADER 软件测试.....7

§3 BOOTLOADER 软件升级时间测试.....9

§4 BOOTLOADER 升级过程中的注意事项.....9

§ 1 BootLoader 设计说明

BootLoader 的设计目的及思路

本次 BootLoader 软件的升级，旨在解决由于远程信号不稳定而造成的升级掉包和升级故障。设计思路为：将远程传输的数据存储至 Pflash 内，待系统确认数据完整之后，再将存储值 Pflash 的数据读取更新至代码运行区。代码存储和代码运行的 Pflash 块区如图 1 所示：

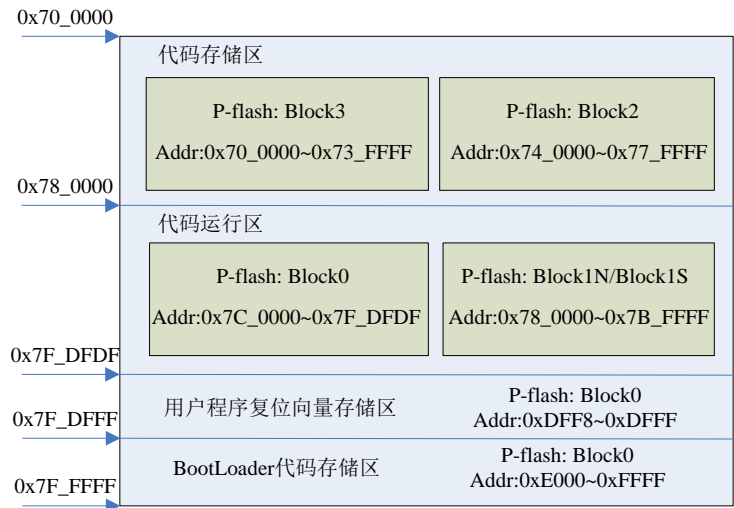


图 1 BootLoader 划分各层的 Pflash 块区

BootLoader 中 CAN 的使用说明

- CAN 通道：CAN2 接口（若需要修改则在源代码上修改，不能占用 flash 空间）；
- CAN 波特率：500kbps；
- 中断向量号：若修改了升级的 CAN 通道则需要在 APP 和 Boot 程序的 prm 文件中根据中断基地址修改 CAN 的中断向量。
- BOOT 升级中的 CAN 通信内容可以参考《BMS 内网通信协议》第 4 章“BootLoader 升级的通信说明”第 4.1 和 4.2 节，Boot 升级过程中的返回状态值按本文的表 1，表 2，表 3。

App 和 BootLoader 中相关程序修改的说明

1. APP 程序中 Task_BootLoader 任务部分的程序

在中断程序中根据升级的通信协议，修改相应的升级变量，用来响应应用程序的升级任务，升级的中断程序如图 2 所示。

```
void CAN2_GetMsg_Process(pCANFRAME receiveFrame)
{
    switch(receiveFrame->m_ID)
    {
        case Boot_ID://0xF300:
            HeartBeat.HeartBeat_CSSU1 = 1;//对子板进行程序升级的时候需要屏蔽掉线的故障
            if(receiveFrame->m_data[1] == 0xAA)
            {
                if(receiveFrame->m_data[0] == ID_BMU)//ID_BMU:0xF0
                {
                    Boot_Data.OnlineUpdateCheck = 1;
                }
            }
            break;
    }
```

图 2 App 中断程序设计

在应用程序的 Boot 任务中，根据接收的升级命令对地址 0x0C00 和地址 0x0C02 进行相应的赋值操作，赋值之后使用看门狗进行复位，根据以上 2 个地址中的值引导程序进入 Boot 模式中，应用的 Boot 程序如图 3 所示。

```
void Task_BootLoader()
{
    DisableInterrupts;
    if(Boot_Data.OnlineUpdateCheck == 1)
    {
        Boot_Data.OnlineUpdateCheck = 0;
        Boot_Data.Boot = (uint16 *)0x0C00;
        Boot_Data.Updateaddr = (uint16 *)0x0C02;
        *(Boot_Data.Boot) = 0x66;
        *(Boot_Data.Updateaddr) = 0x77;

        Boot_DelayTime(100);

        COPCTL=0x01; //单片机复位
        ARMCOP=0x00;
    }
    EnableInterrupts;
    g_Roll_Tick.Roll_Boot++;
}
```

图 3 App 应用程序的 BOOT 任务设计

2. APP 与 BootLoader 中 prm 文件地址段的划分

- **App** 的代码段划分如图 4、图 5 所示：

```

EEPROM      = READ_ONLY    DATA_NEAR IBCC_NEAR 0x0C00 TO 0x0FFF;
/* non-paged RAM */
RAM         = READ_WRITE   DATA_NEAR          0x2000 TO 0x3FFF;
/* non-banked FLASH */
ROM_4000    = READ_ONLY    DATA_NEAR IBCC_NEAR 0x4000 TO 0x7F0F; //0x7F10~0x7FFF为中断向量
//将IVBR设置为0x7F, 将中断地址转换为全局地址时, 只需要加0x7F_0000就可以。
ROM_C000    = READ_ONLY    DATA_NEAR IBCC_NEAR 0xC000 TO 0xDFDF; //0xDFE0~0xDFFF为复位向量地址

```

图 4 App 的 EEprom 及 Rom 使用区

```

DEFAULT_ROM INTO PAGE_FE, PAGE_FC, PAGE_FB, PAGE_FA, PAGE_F9, PAGE_F8,
PAGE_F7, PAGE_F6, PAGE_F5, PAGE_F4, PAGE_F3, PAGE_F2, PAGE_F1, PAGE_F0,
PAGE_EF, PAGE_EE, PAGE_ED, PAGE_EC, PAGE_EB, PAGE_EA, PAGE_E9, PAGE_E8,
PAGE_E7, PAGE_E6, PAGE_E5, PAGE_E4, PAGE_E3, PAGE_E2, PAGE_E1, PAGE_E0,
//PAGE_DF, PAGE_DE, PAGE_DD, PAGE_DC, PAGE_DB, PAGE_DA, PAGE_D9, PAGE_D8,
//PAGE_D7, PAGE_D6, PAGE_D5, PAGE_D4, PAGE_D3, PAGE_D2, PAGE_D1, PAGE_D0,
//PAGE_CF, PAGE_CE, PAGE_CD, PAGE_CC, PAGE_CB, PAGE_CA, PAGE_C9, PAGE_C8,
//PAGE_C7, PAGE_C6, PAGE_C5, PAGE_C4, PAGE_C3, PAGE_C2, PAGE_C1, PAGE_C0;

VECTOR ADDRESS 0x7F7A isrPIT0
VECTOR ADDRESS 0x7F9A Monitor_CAN3
VECTOR ADDRESS 0x7FB2 Charge_CAN0
VECTOR ADDRESS 0x7FA2 VCU_CAN2

```

图 5 App 的 Pflash 使用块区及中断向量地址

- **BootLoader** 的代码段划分如图 6、图 7 所示：

```

EEPROM      = READ_ONLY    0x0C00 TO 0x0FFF;
//EEPROM    = READ_WRITE   0x0C00 TO 0x0FFF;
RAM         = READ_WRITE   0x2000 TO 0x3FFF; //0x3CFF;
RAM_CODE_SEG = READ_ONLY    0xFC00 TO 0xFEFF RELOCATE_TO 0x3C00;
ROM_F000    = READ_ONLY    0xE000 TO 0xFBFF;

```

图 6 BootLoader 的 EEprom 及 Rom 使用区

```

PLACEMENT /* here all predefined and user segments are placed into the SEGMENTS defined above. */
ROM_VAR.
STRINGS.
DEFAULT_ROM.
NON_BANKED INTO ROM_F000;

DEFAULT_RAM INTO RAM;

RAM_CODE INTO RAM_CODE_SEG;

END

ENTRIES
// flash_array
END

STACKSIZE 0x100

//handle all reset vectors
VECTOR 0 _BootStart
VECTOR 1 _BootStart
VECTOR 2 _BootStart
VECTOR ADDRESS 0xFF9A CAN3RrISR
VECTOR ADDRESS 0xFF7A isrPIT0

```

图 7 BootLoader 的 Pflash 使用块区及中断向量地址

3. BootLoader 升级新程序的步骤

- 将 Bootloader 程序下至单片机内；
单片机首次使用时建议在 Boot 程序的 Flash 初始化中进行全部的块擦除。细节如图 8 所示。

```
//部分分区(单片机的首次使用时最好进行全分区和擦除flash所有的块)  
FlashErr = LaunchFlashCommand(2, PARTITION_D_FLASH, 0, DF PART, EEE_RAM, 0, 0, 0, 0, 0);
```

图 8 BootLoader 初始化中部分分区和全分区的设定

- 升级步骤如图 9 所示：

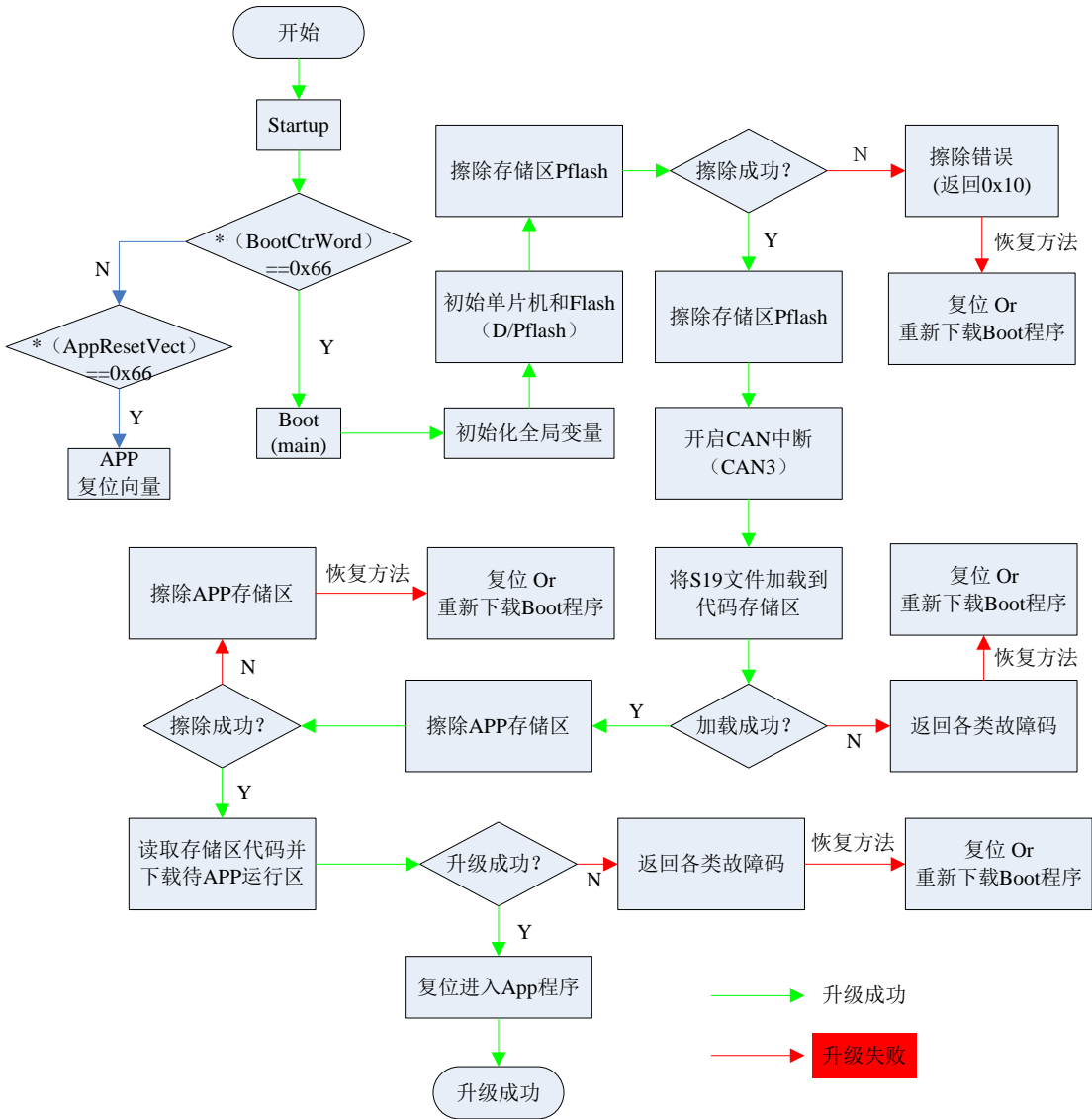


图 9 Boot 升级流程图

BootLoader 下载中返回的状态值

表 1 数据返回总类型

CAN ID 值	数据内容	数据说明
0x1FF (标准帧)	0xC0	进入了 Boot 程序，开始进行存储区擦除
	0xC1	存储区 flash 擦除成功，准备进行加载
	0xC2	S19 代码加载至存储区成功
	0xC3	开始进行程序升级，准备进行 App 区擦除
	0xC4	App 运行区擦除成功，准备进行升级
	0xC5	代码升级成功
	0xC6	代码升级错误，需要重启或重新下载 Boot

表 2 程序下载过程中返回错误类型类型

CAN ID 值	数据内容	数据说明
0x1FF (标准帧)	0x10	代码存储区擦除失败（复位解除）
	0x11	代码接收错误
	0x12	对非 8 字节进行对齐时存储队列溢出
	0x13	下载待到代码存储区错误
	0x14	8 字节对其的代码下载到代码存储区错误
	0x01	下载过程中成功
	0x21	字符存储溢出
	0x22	S19 文件数据长度转换为 16 进制错误
	0x23	S19 文件接收数据长度错误
	0x24	S19 中接收地址转换错误
	0x25	S19 中接收数据转换错误
	0x26	S19 中接收校验符转换错误
	0x27	CRC 校验错误

表 3 程序升级过程中返回错误类型类型

CAN ID 值	数据内容	数据说明
0x1FF (标准帧)	0x80	App 存储区擦除失败（复位解除）
	0x81	从代码存储器读取代码错误
	0x82	对复位向量加载的地址错误
	0x83	代码升级过程中出现错误
	0x02	升级过程中成功
	0x91	S19 文件读取数据长度转换为 16 进制错误
	0x92	S19 文件读取字符长度错误
	0x93	S19 中读取地址转换错误
	0x94	S19 中读取数据转换错误
	0x95	S19 中读取校验符转换错误
	0x96	CRC 校验错误

§ 2 BootLoader 软件测试

使用 CAN-II 及 Codewarrior 调试界面如图 10~图 17 所示。

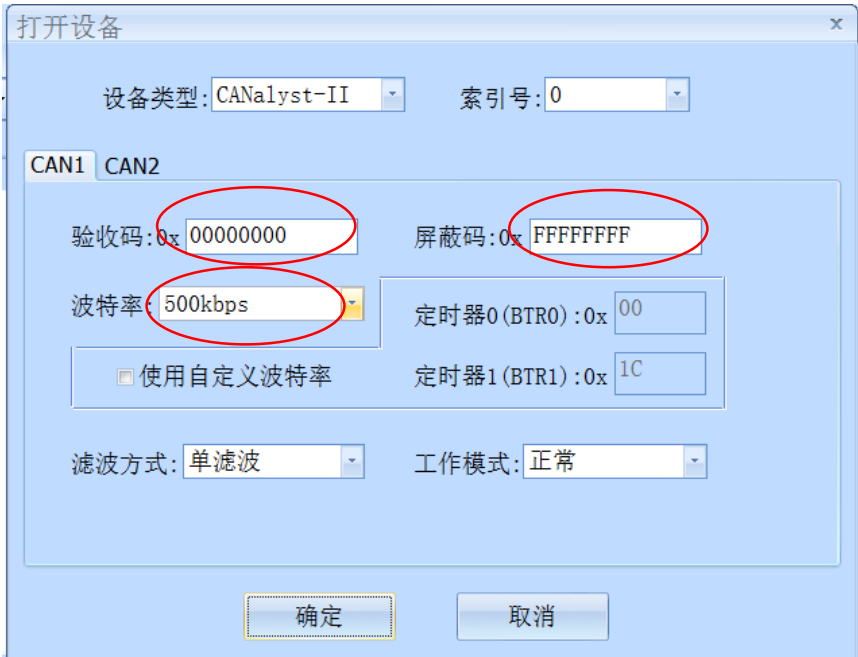


图 10 打开 CAN-II

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	接收	22:05:53.148			0x00000002	数据帧	标准帧	0x01	C0
1	接收	22:05:53.148			0x00000002	数据帧	标准帧	0x01	C1

图 11 进入 Boot 程序的启动帧（参考表 1）

0	接收	22:05:53.148			0x00000002	数据帧	标准帧	0x01	C0
1	接收	22:05:53.148			0x00000002	数据帧	标准帧	0x01	C1

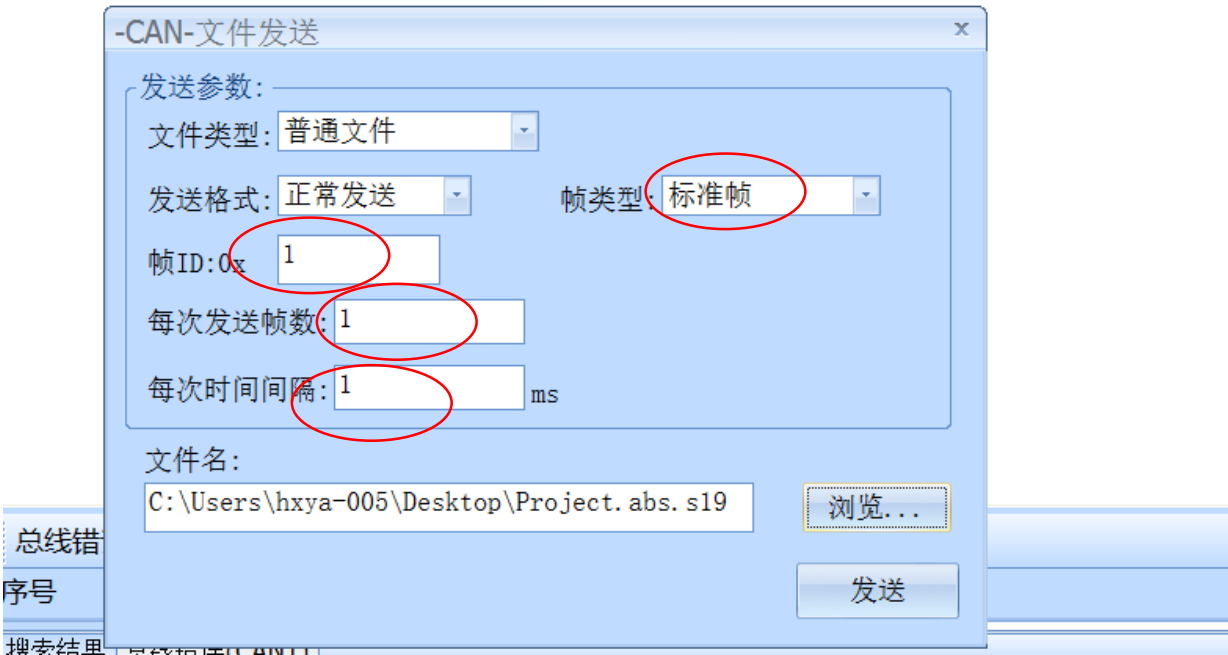


图 12 发送 S19 文件的相关配置

6151	接收	22:09:55.927		0x00000002	数据帧	标准帧	0x01	01
6152	发送	22:09:55.927	成功	0x00000001	数据帧	标准帧	0x08	35 43 31 36 42 41 39 33
6153	发送	22:09:55.930	成功	0x00000001	数据帧	标准帧	0x08	38 36 38 30 43 37 33 42
6154	发送	22:09:55.932	成功	0x00000001	数据帧	标准帧	0x08	33 35 46 43 32 32 32 31
6155	发送	22:09:55.935	成功	0x00000001	数据帧	标准帧	0x08	46 45 32 32 31 46 31 36
6156	发送	22:09:55.938	成功	0x00000001	数据帧	标准帧	0x08	43 31 39 42 31 42 38 34
6157	发送	22:09:55.940	成功	0x00000001	数据帧	标准帧	0x08	31 36 42 41 31 36 31 36
6158	发送	22:09:55.943	成功	0x00000001	数据帧	标准帧	0x08	42 39 42 30 38 36 38 30
6159	发送	22:09:55.946	成功	0x00000001	数据帧	标准帧	0x08	43 37 33 42 33 35 46 43
6160	发送	22:09:55.949	成功	0x00000001	数据帧	标准帧	0x08	45 31 0D 0A 53 32 32 34
6161	发送	22:09:55.951	成功	0x00000001	数据帧	标准帧	0x08	46 43 42 35 34 30 32 32
6162	发送	22:09:55.953	成功	0x00000001	数据帧	标准帧	0x08	32 35 46 45 32 32 32 33
6163	接收	22:09:55.954		0x00000002	数据帧	标准帧	0x01	01

图 13 程序加载至相应存储区的返回值……成功（参考表 2）

12081	接收	22:10:13.933		0x00000002	数据帧	标准帧	0x01	C2
12082	接收	22:10:13.933		0x00000002	数据帧	标准帧	0x01	C3
12083	接收	22:10:19.102		0x00000002	数据帧	标准帧	0x01	C4

图 14 程序加载加载成功并进行升级（参考表 1）

12084	接收	22:10:19.102		0x00000002	数据帧	标准帧	0x01	02
12085	接收	22:10:19.102		0x00000002	数据帧	标准帧	0x01	02
12086	接收	22:10:19.110		0x00000002	数据帧	标准帧	0x01	02
12087	接收	22:10:19.110		0x00000002	数据帧	标准帧	0x01	02
12088	接收	22:10:19.126		0x00000002	数据帧	标准帧	0x01	02

图 15 程序升级中……成功（参考表 3）

13209	接收	22:10:33.102		0x00000002	数据帧	标准帧	0x01	02
13210	接收	22:10:33.118		0x00000002	数据帧	标准帧	0x01	02
13211	接收	22:10:33.123		0x00000002	数据帧	标准帧	0x01	02
13212	接收	22:10:33.127		0x00000002	数据帧	标准帧	0x01	C5

图 16 程序升级完成（参考表 1）

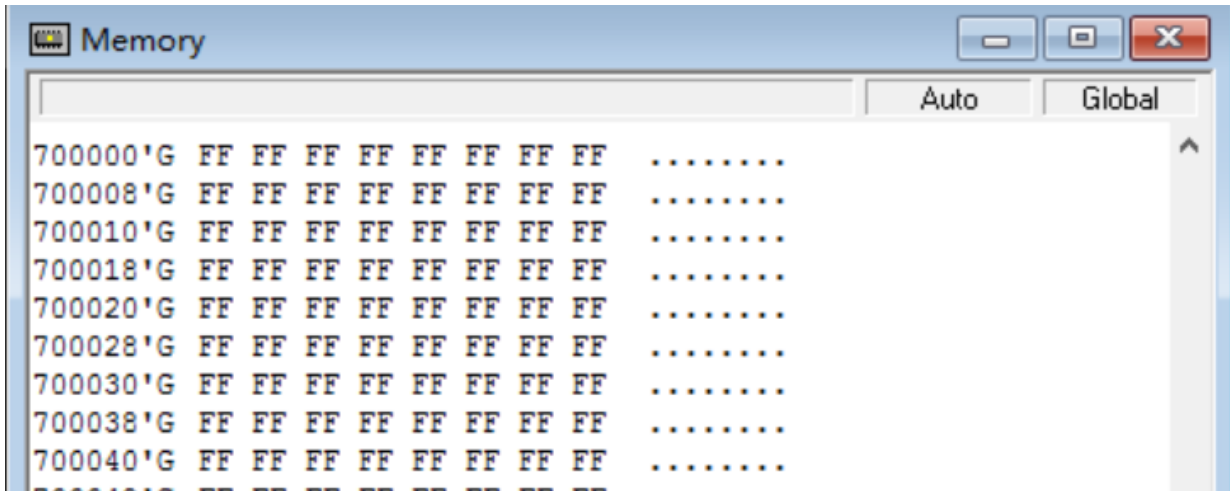


图 17 调试界面中擦除存储区的块

§ 3 BootLoader 软件升级时间测试

- 代码下载的时间 (第 2 节 CAN-II 配置为准 ;以东方襄旅的裸机程序为例 85.5Kb)

2	发送	22:09:35.627	成功	0x00000001	数据帧	标准帧	0x08	53 30 34 46 30 30 30 30
---	----	--------------	----	------------	-----	-----	------	-------------------------

图 18 S19 文件首帧下载的时间

12080	接收	22:10:13.933		0x00000002	数据帧	标准帧	0x01	01
12081	接收	22:10:13.933		0x00000002	数据帧	标准帧	0x01	C2

图 19 S19 文件末帧下载的时间

可以确认：程序从擦除 Pflash 到下载至相应的 Pflash 块中运行的时间大约为：38S

- 代码升级的时间

12082	接收	22:10:13.933		0x00000002	数据帧	标准帧	0x01	C3
-------	----	--------------	--	------------	-----	-----	------	----

图 20 读取代码开始的时间

13211	接收	22:10:33.123		0x00000002	数据帧	标准帧	0x01	02
13212	接收	22:10:33.127		0x00000002	数据帧	标准帧	0x01	C5

图 21 程序升级成功的时间

可以确认：程序从擦除 Pflash 到读取相应的 Pflash 块中程序到升级成功运行的时间大约为：20S

§ 4 BootLoader 升级过程中的注意事项

1. 在 Boot 程序下载之后，必须拔掉 BDM，否则带电的复位引脚会影响程序中看门狗的复位，也会影响后续的程序重新升级；
2. 若 S19 文件在下载的过程中出现通信中断，那么 CAN 中持续会返回错误码，此时 BMS 需要重新启动，启动后进入旧 APP 程序中；
3. 若 S19 下载成功但在升级的过程中出现掉电情况，那么此时 BMS 重新启动后会对存储区的新程序进行自动升级（此时不需要重新进行 S19 文件的下载），升级完成后需要重启，那么启动后进入新 APP 程序中；
4. 如果下载的 App 程序有 Bug，那么经过 BootLoader 下载升级后进入含有 Bug 的程序的，此时会导致系统崩溃。不能通过重启来进行重新升级，此时需要重新烧写 BootLoader 程序进行重新下载、升级正常的 App 程序；
5. 本程序的进行程序下载和升级是经过调试后最快时间，若再缩短程序中的中断（读写时间），那么会导致系统在升级的过程中出现未知错误。

警告 (Waring)

在使用程序之前请仔细阅读说明文档，本 Bootloader 程序的地址和应用程序的地址是经过详细计算得出，未经过本人同意擅自改动造成的系统故障或程序升级不成功，本人概不负责。

特此说明!