

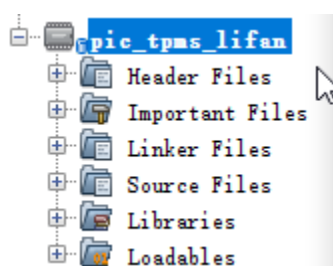
PIC18F25K80 CAN bootloader 使用说明:

微芯商贸 李域
skyler.li@microchip.com

第一步

App 项目是被升级的文件

CAN-pic18f 是 Bootloader 文件



第二步

在 app 项目设置升级地址为 1500h,并且将输出 bin 文件的选项打开:



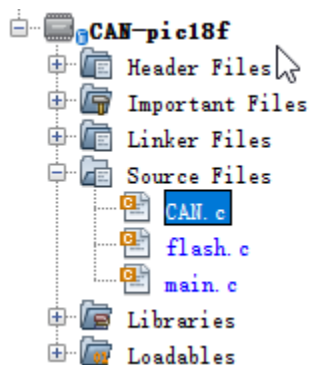
第三步

在 app 这里随便改一改代码, 比如用 CAN 输出 0x55, 0xAA 之类的

第四步

编译 APP, 拷贝出来 HEX 备用

第五步: 将 CAN-pic18f 直接下载进入 MCU, 注意该程序的初始地址为 400H, 程序中 0~400h 地方是留给中断服务程序的:





程序启动，在 CAN 接收端显示如下：

序号	传输方向	时间标识	名称	帧ID(靠右对齐)	帧格式	帧类型	数据长度	数据(十六进制)
29983	接收	18:28:52:202		0x000007df	数据帧	标准帧	08	01 02 03 04 05 06 07 08
29984	接收	18:28:52:405		0x000007df	数据帧	标准帧	01	01
29985	接收	18:28:52:603		0x000007df	数据帧	标准帧	01	02
29986	接收	18:28:52:801		0x000007df	数据帧	标准帧	01	03
29987	接收	18:28:53:004		0x000007df	数据帧	标准帧	01	04
29988	接收	18:28:53:202		0x000007df	数据帧	标准帧	01	05
29989	接收	18:28:53:404		0x000007df	数据帧	标准帧	01	06
29990	接收	18:28:53:605		0x000007df	数据帧	标准帧	01	07

在记数到十进制 30，HEX 1D 之前输入如下数据：

帧ID(Hex): 固定 707 发送

数据(Hex): 固定 aa 22 33 44 55 66 77 88 停止

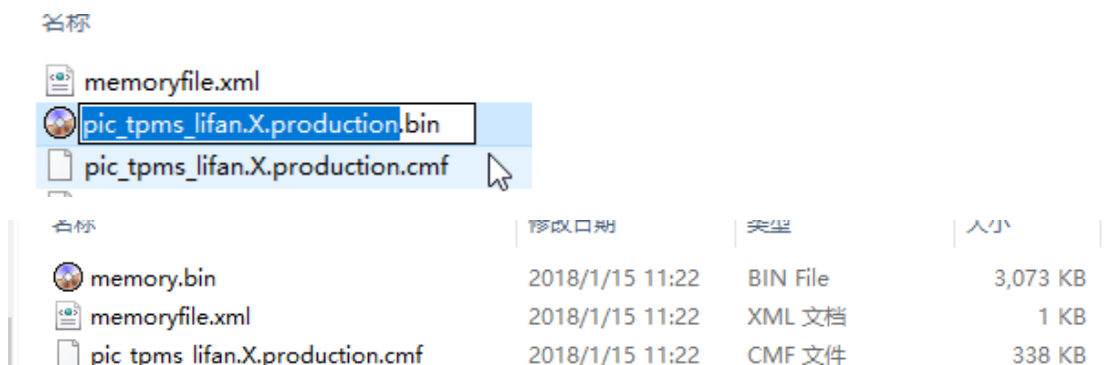
第六步：

输入成功之后中断了启动过程，进入升级准备状态：

29995	接收	18:28:54:603		0x000007df	数据帧	标准帧	01	0c
29996	接收	18:28:54:804		0x000007df	数据帧	标准帧	01	0d
29997	发送	18:28:54:979		0x00000707	数据帧	标准帧	08	aa 22 33 44 55 66 77 88
29998	接收	18:28:54:989		0x000007df	数据帧	标准帧	07	41 63 6b 2d 2d 2d aa
29999	接收	18:28:55:532		0x000007df	数据帧	标准帧	08	aa 22 33 44 55 66 77 88







第六步：

将 App 的 bin 文件名修改为 memory.bin

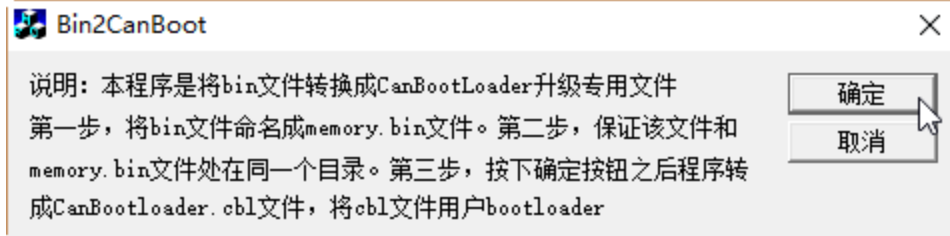


第七步：

保证这些文件在同一目录下：

	Bin2CanBoot.exe	2017/12/31 17:52	应用程序	105 KB
	memory.bin	2017/12/31 18:37	BIN File	9 KB
	MFC42D.DLL	1998/6/17 0:00	应用程序扩展	909 KB
	MFCO42D.DLL	1998/6/17 0:00	应用程序扩展	781 KB
	MSCOMM32.OCX	1998/6/24 0:00	ActiveX 控件	102 KB
	MSVCRTD.DLL	1998/6/17 0:00	应用程序扩展	377 KB

然后启动 Bin2CanBoot.exe 文件，按下确定



生成 CanBootLoader.cbt 文件

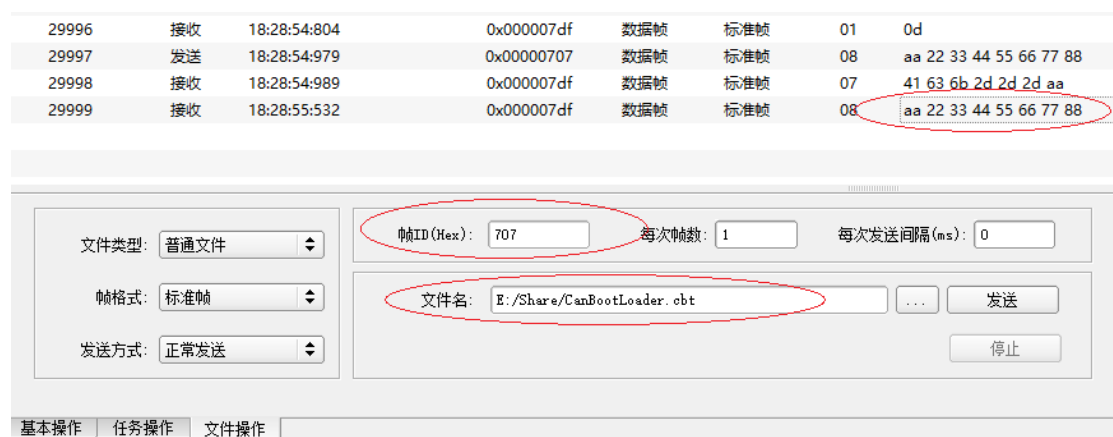
此时，回到第六步的界面：

第八步：

回到文件操作的界面：



检查界面如下：



点击发送

第十步：

等界面出现 7777...的字样升级结束。

000707	数据帧	标准帧	02	aa aa
0007df	数据帧	标准帧	08	aa aa aa aa aa aa aa aa
0007df	数据帧	标准帧	08	aa aa aa aa aa aa aa aa
0007df	数据帧	标准帧	08	aa aa aa aa aa aa aa aa
0007df	数据帧	标准帧	08	77 77 77 77 77 77 77 77

最后：断电重启

可以看到，发送 55aa 的情况，升级成功。

29990	接收	18:48:40:141	0x000007df	数据帧	标准帧	01	55
29991	接收	18:48:40:159	0x000007df	数据帧	标准帧	01	aa
29992	接收	18:48:40:181	0x000007df	数据帧	标准帧	01	55
29993	接收	18:48:40:199	0x000007df	数据帧	标准帧	01	aa
29994	接收	18:48:40:221	0x000007df	数据帧	标准帧	01	55
29995	接收	18:48:40:243	0x000007df	数据帧	标准帧	01	aa
29996	接收	18:48:40:260	0x000007df	数据帧	标准帧	01	55
29997	接收	18:48:40:283	0x000007df	数据帧	标准帧	01	aa
29998	接收	18:48:40:300	0x000007df	数据帧	标准帧	01	55
29999	接收	18:48:40:321	0x000007df	数据帧	标准帧	01	aa

备注：

Bin2CanBoot 文件代码如下，该文件强制修改了 boot 的跳入指针：

```
void CBin2CanBootDlg::OnOK()
```

```
{
```

```
    // TODO: Add extra validation her
```

```
    unsigned char startpoint[]={0x00,0xEF,0x02,0xF0};
```

```
    char *chfname;
```

```
    FILE *fps,*fpo;
```

```
    char ch,j,chend;
```

```
    unsigned long i=0;
```

```
    fps=fopen("memory.bin","rb");
```

```
    fpo=fopen("CanBootLoader.cbt","wb");
```

```
    if ((fps==NULL) || (fpo==NULL))
```

```
    {
```

```
        ::MessageBox(NULL,"程序打开错误","请检查 bin 文件是否存在并且可读,以及 cbt 文件是否被占用导致无法输出",MB_OK);
```

```
        return ;
```

```
    }
```

```
    while(!feof(fps))
```

```
{
    ch=fgetc(fps);
    if((0<=i)&&(i<=3))
    {
        fprintf(fpo,"%c",startpoint[i]);    //将程序指针强制跳入 0x400BootLoader 位置
    }
    else
    {
        fprintf(fpo,"%c",ch);
    }

    i++;
    if(i>=0x8000)
    {
        fclose(fps);
        fclose(fpo);
        return;
    }
}

return ;

}
```