

[EMBF 用户手册]

[简介了 **EMBF** 系统的常用功能以及 **API** 接口]

[Ver **EMBF-V1-A**]

升级历史:	4
1、功能.....	4
2、特性与规格.....	4
系统特性:	4
串口驱动特性:	4
CAN 口驱动特性:	4
外部看门狗特性:	4
LM240160KCW 液晶模块驱动特性:	5
EEPROM 驱动特性:	5
扩展 Flash 特性:	5
按键驱动特性:	5
网络驱动特性:	5
3、框图.....	6
4、目录与文件结构.....	6
工程文件目录与文件说明:	6
基本工程文件与库文件说明:	7
5、版本管理方式.....	7
6、使用流程.....	7
7、开发环境配置.....	8
8、API 说明.....	17
assage 模块.....	17
void assage_create_sig(int32_t fd).....	17
uint32_t assage_accept(int32_t fd).....	17
void assage_init(void).....	17
assage_handle assage_create_msg(t_assage_msg msg_buf[],uint8_t size,uint32_t mask).....	18
void assage_post_msg(assage_handle handle,t_assage_msg *msg).....	18
void assage_post_all (t_assage_msg *msg).....	18
void assage_get_msg(assage_handle handle,uint8_t timeout,t_assage_msg *lmsg).....	18
void assage_del_msg(assage_handle handle).....	18
fcntl 模块.....	19
int32_t drv_open(const char*, int32_t, int32_t);.....	19
int32_t drv_read(int32_t, void*, uint32_t).....	19
int32_t drv_write(int32_t, const void*, uint32_t).....	19
int32_t drv_lseek(int32_t, int32_t, int32_t).....	19
int32_t drv_close(int32_t).....	19
int32_t drv_ioctl(int32_t, int32_t,void *).....	19
int32_t drv_poll(int32_t fd,void *ev).....	19
lcd_module 模块 (续)	20
bool LCD_open(TLCDModule *).....	20
液晶类接口。	20
void (*clear)(void).....	20
无.....	20
void (*cleararea)(uint8_t,uint8_t,uint8_t,uint8_t,uint8_t).....	20
灰度, 这里可以按指定灰度清屏.....	20

void (*setbacklight)(uint16_t).....	20
对比度参数, 调整范围是 0 ~ 100.....	20
void (*writechar)(uint16_t ,uint16_t ,const char *,uint16_t).....	20
具体字体.....	20
void (*writestr)(uint16_t ,uint16_t ,const char *,uint8_t ,uint16_t).....	20
具体字体.....	21
void (*writestru)(uint16_t ,uint16_t ,const char *,uint8_t ,uint8_t).....	21
void (*write_str)(TLCDSrParam *param).....	21
void (*showimage)(uint8_t ,uint8_t ,uint8_t ,uint8_t ,const uint8_t *,short ,uint8_t).....	21
void (*drawline)(uint8_t ,uint8_t ,uint8_t ,uint8_t ,uint8_t).....	21
标志, LCD_DRAW_HORI -- 横线, LCD_DRAW_VERT -- 竖线.....	21
void (*drawrect)(uint8_t,uint8_t,uint8_t,uint8_t).....	22
宽度.....	22
void (*drawrectarc)(uint8_t,uint8_t,uint8_t,uint8_t,uint8_t,uint8_t).....	22
填充灰度.....	22
void (*setfontcolor)(uint8_t,uint8_t).....	22
背景色.....	22
message 模块.....	22
void InitMsg(PProcBuf Entry,PMsgProc GetMsg).....	22
void EnableMsgType(PProcBuf Entry,int MsgType,int Enabled).....	22
uint32_t SelectNewMsgMask(PProcBuf Entry,uint32_t newmsk).....	23
int DoMsgProcess(PProcBuf Entry, void *Obj, int ExitCommand).....	23
int RegMsgProc(PProcBuf Entry,PMsgProc MsgProc).....	23
int UnRegMsgProc(PProcBuf Entry,int index).....	23
int PostMessage(PProcBuf Entry,int Msg,int P1,int P2).....	23
utils 模块.....	23
9、常见问题.....	24

升级历史：

版本号	拟制责任人	拟制日期	修改内容及理由	审核人	批准人
1.0	刘雷	2012-10-16	新拟制、初始化		

1、功能

Ucos II 2.92 内核

全异步事件模型，关键的任务处理均通过消息触发

完整的驱动包支持，通用接口

增强型的 UI 组件，支持绘图，灰度图片显示

预定义的图片库

灵活的菜单实现，可独立编辑每个页面

完整的字库支持，中文 3 种，英文 5 种，另外支持欧洲等特殊文字字库

精简的 API 接口，简化应用设计

支持完整网络协议库

Modbus 协议库

Messaeg 消息分发库

Base64 编码解码库

通用函数库

2、特性与规格

系统特性：

额定任务数 3 个，分别为主进程，网络进程，UI 进程。

预定义进程数 8 个

最大支持事件数 10 个

驱动调度延时 6us

串口驱动特性：

串口默认波特率 9600bps

串口缓冲区大小 256B

CAN 口驱动特性：

CAN 口默认波特率 1Mbps

外部看门狗特性：

喂狗时间<1.6s

芯片工作自动判断

LM240160KCW 液晶模块驱动特性：

支持像素级操作

支持灰度显示

EEPROM 驱动特性：

32KB 存储空间

每页长度 32B

写入时间 5ms

擦写寿命 1000K

扩展 Flash 特性：

容量 8MB

每页 4096B

高速读取和写入

擦写寿命 100K

按键驱动特性：

支持长按，短按

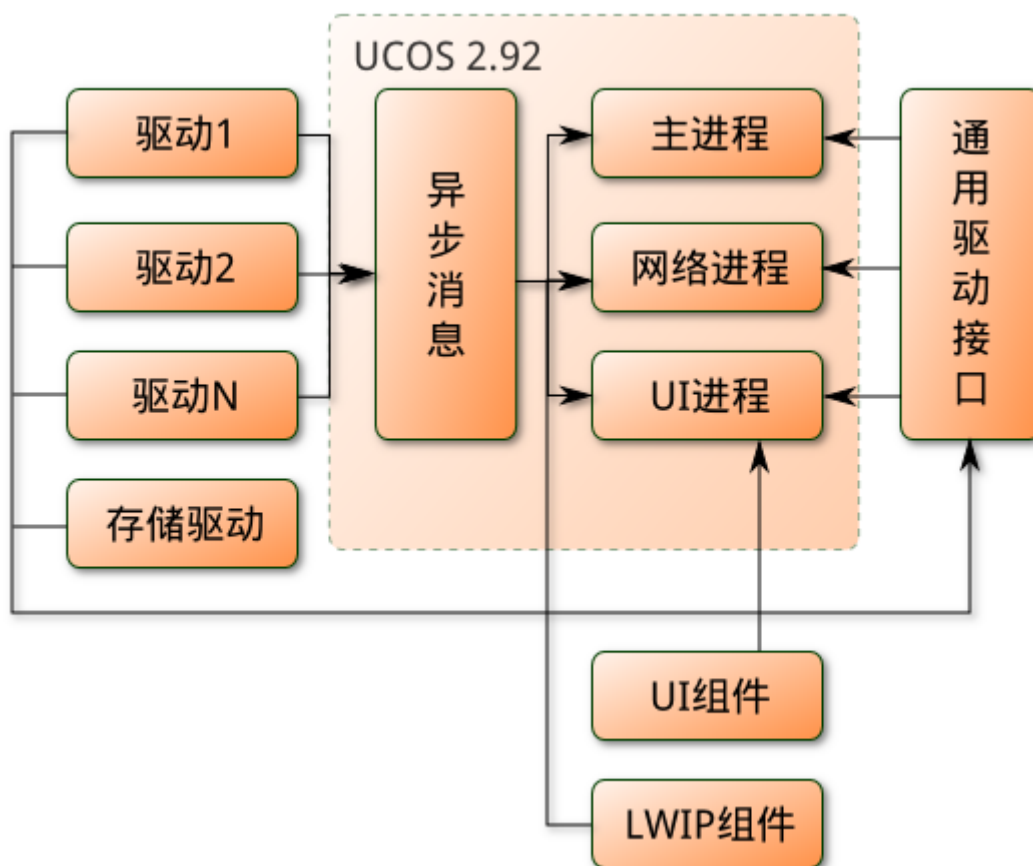
延时时间 50ms

网络驱动特性：

高速 DMA 缓冲

支持全双工操作

3、框图



4、目录与文件结构

工程文件目录与文件说明：

Includes -- 库头文件包含目录

Sys -- 系统库头文件，其中包括部分驱动头文件

Lwip -- lwip 网络库头文件

MDK-F4 -- 工程文件目录

Obj -- 二进制对象目录

EMBF.htm -- 分析堆栈空间的文件

EMBF.map -- 分析存储分配与代码资源分配的文件

EMBF-e.bin -- 加密的二进制文件

EMBF.uvproj -- 工程文件

Yyaencrypt.exe -- AES 加密工具

hex2bin.exe -- hex 文件转二进制格式工具

Build.bat -- 加密文件处理脚本

Session -- 应用程序代码目录

基本工程文件与库文件说明：

base64.c -- base64 编码与解码库

chart.c -- 报表控件

chart_interface.c -- 报表控件接口文件

Image.c -- 图片库

Language.c -- 语言库

Menu.c -- 用户菜单

Modbus.c -- modbus 通信库

task_main.c -- 主进程

task_net.c -- 网卡进程

task_ui.c -- ui 进程

Widget.c -- 图形控件

5、版本管理方式

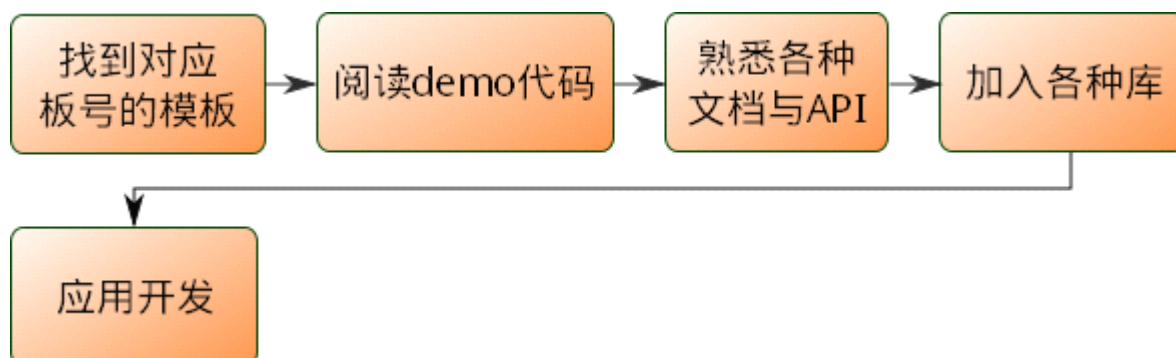
为强化版本管理加强质量控制，EMBF 平台的版本管理采用不同的版本管理方式。

版本管理的约定：

1、EMBF 库的分发方式为库+头文件的方式，库的命名是按 PCB 板号来

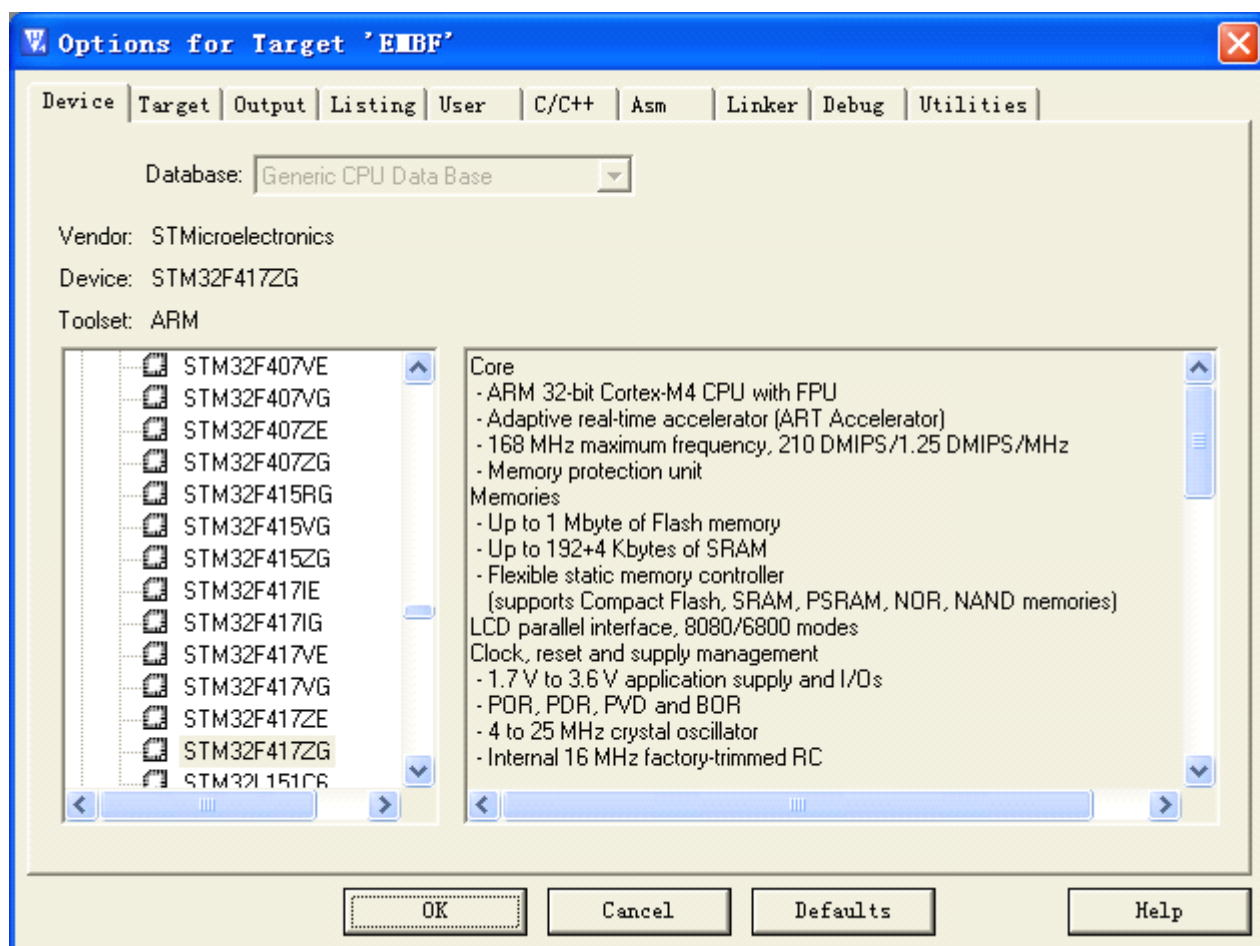
2、

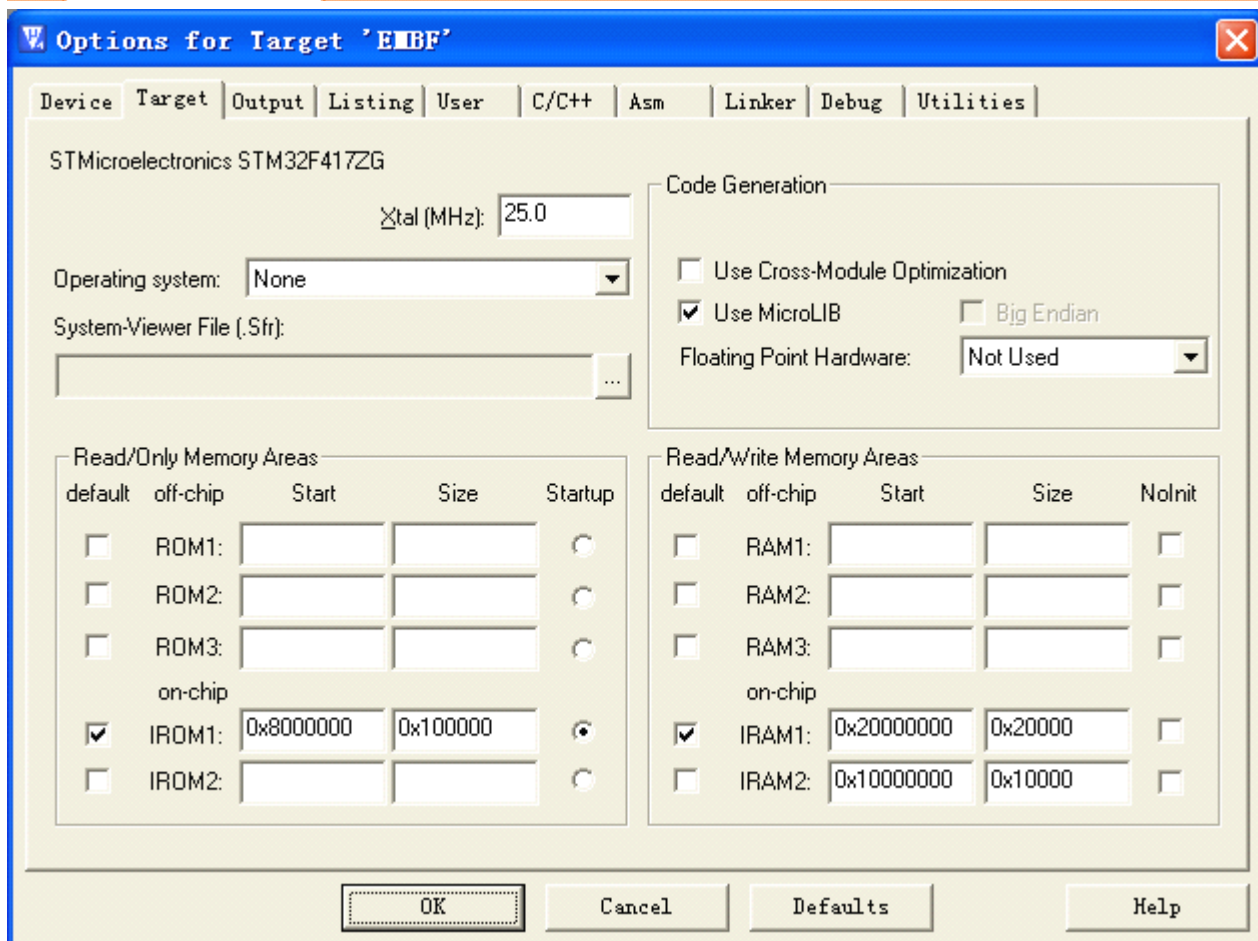
6、使用流程

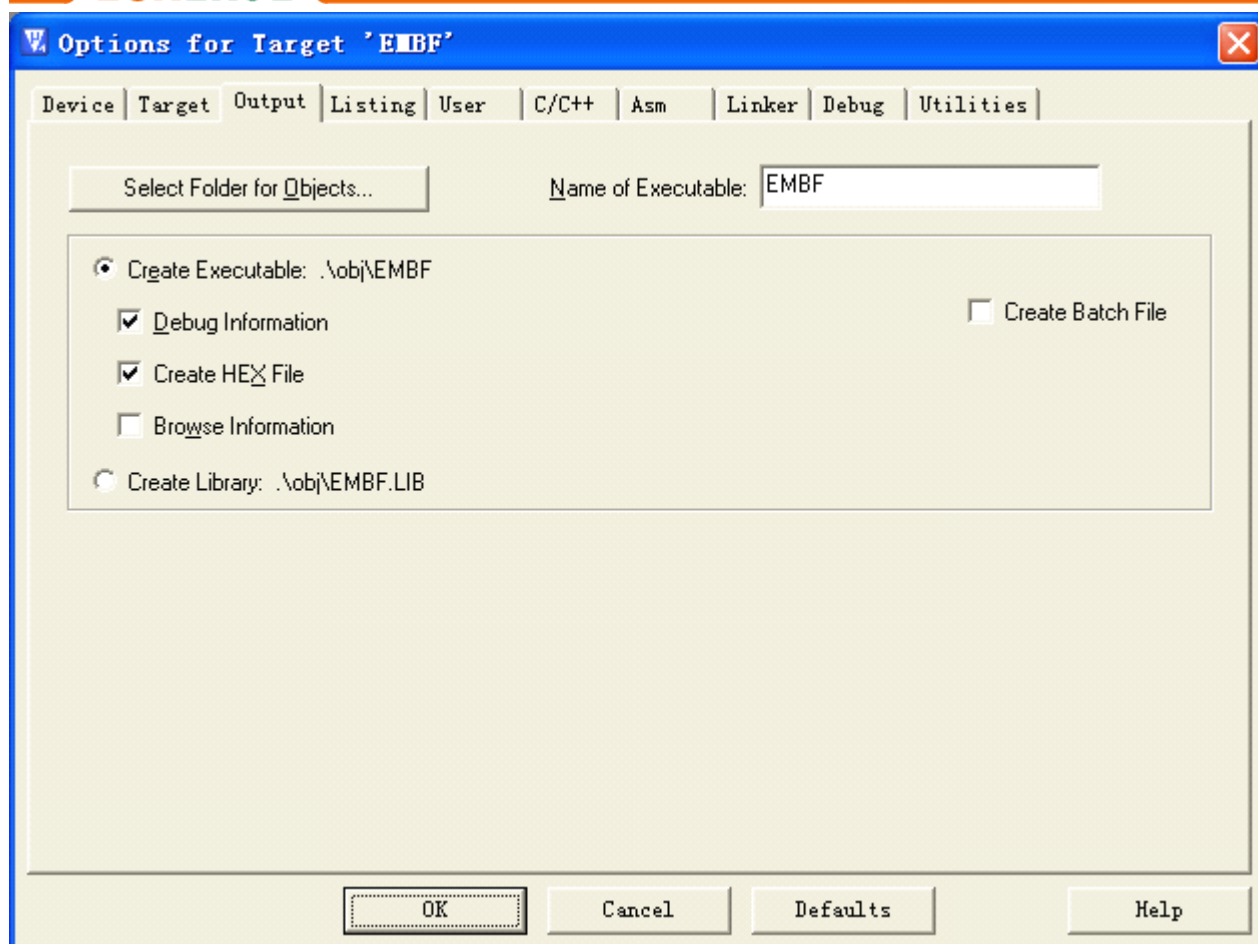


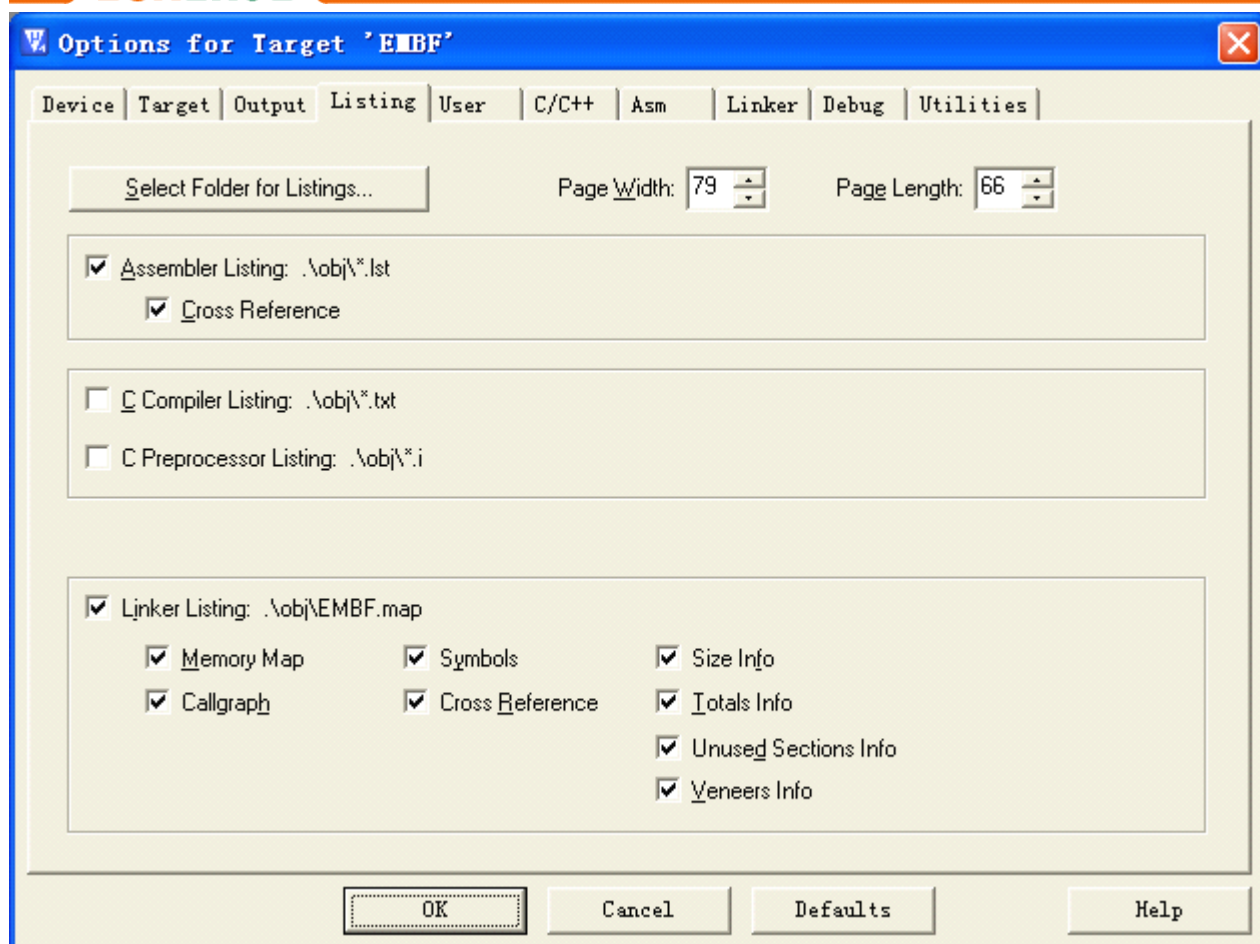
7、开发环境配置

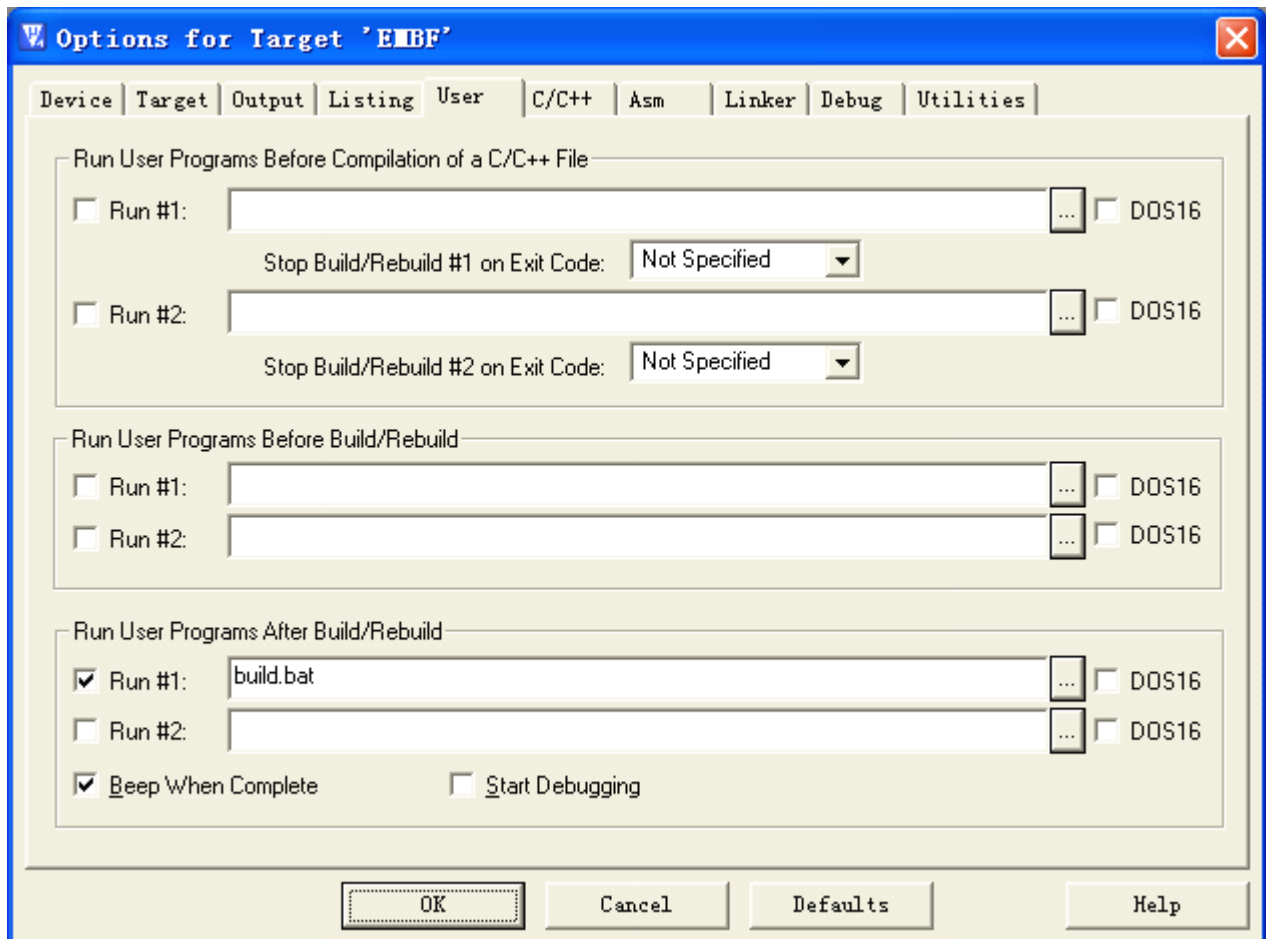
目前平台使用 keil 平台来做开发环境配置信息如下：

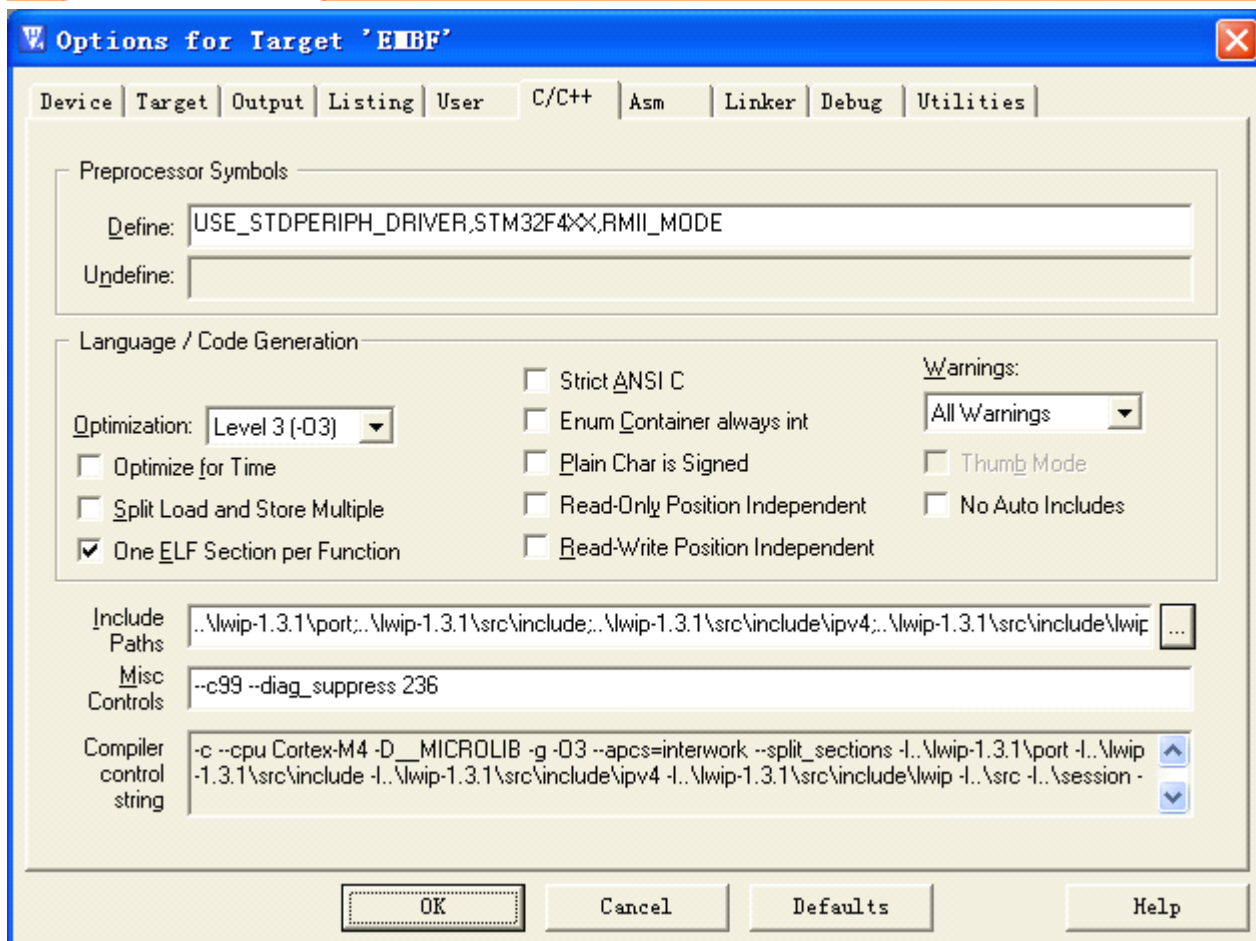


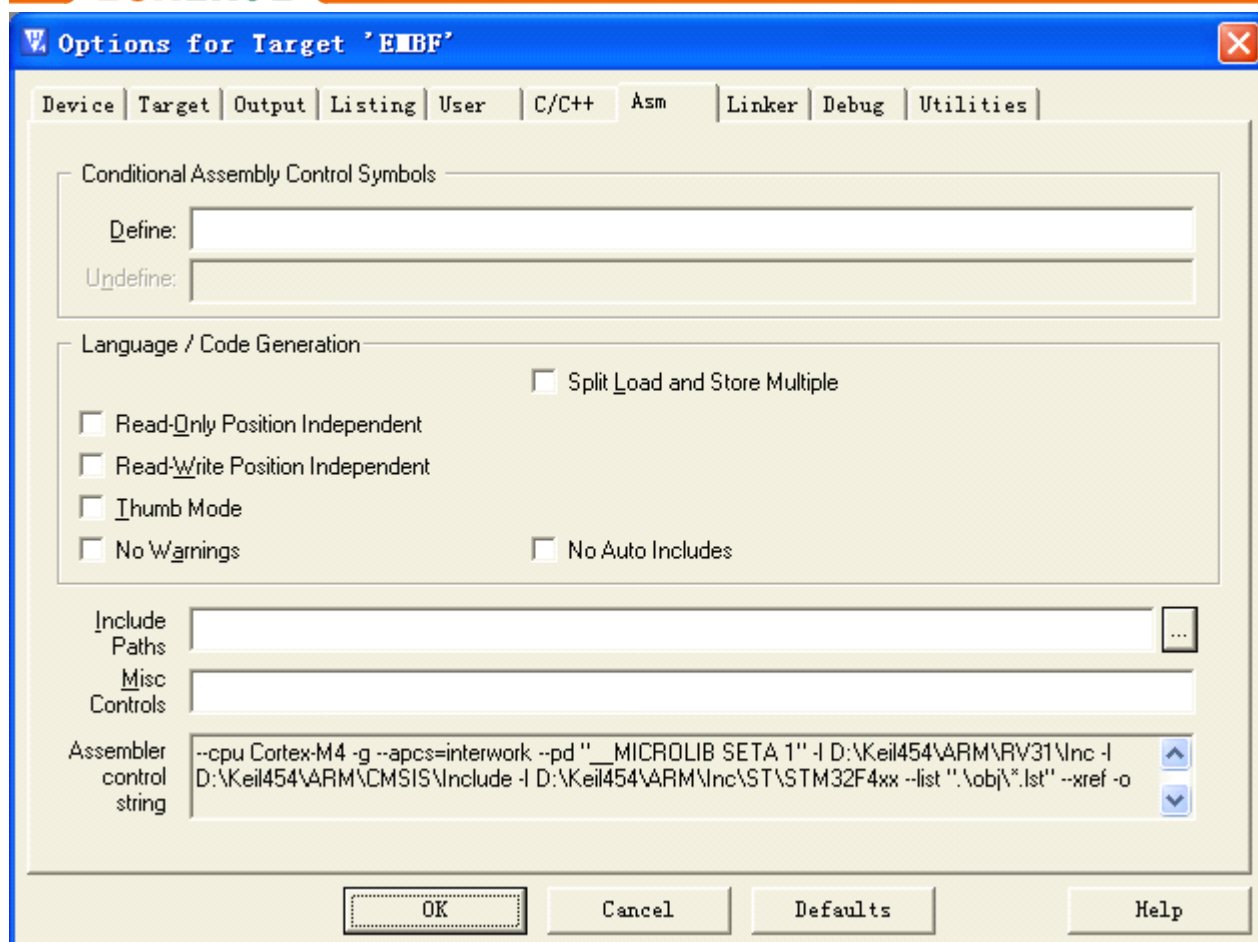


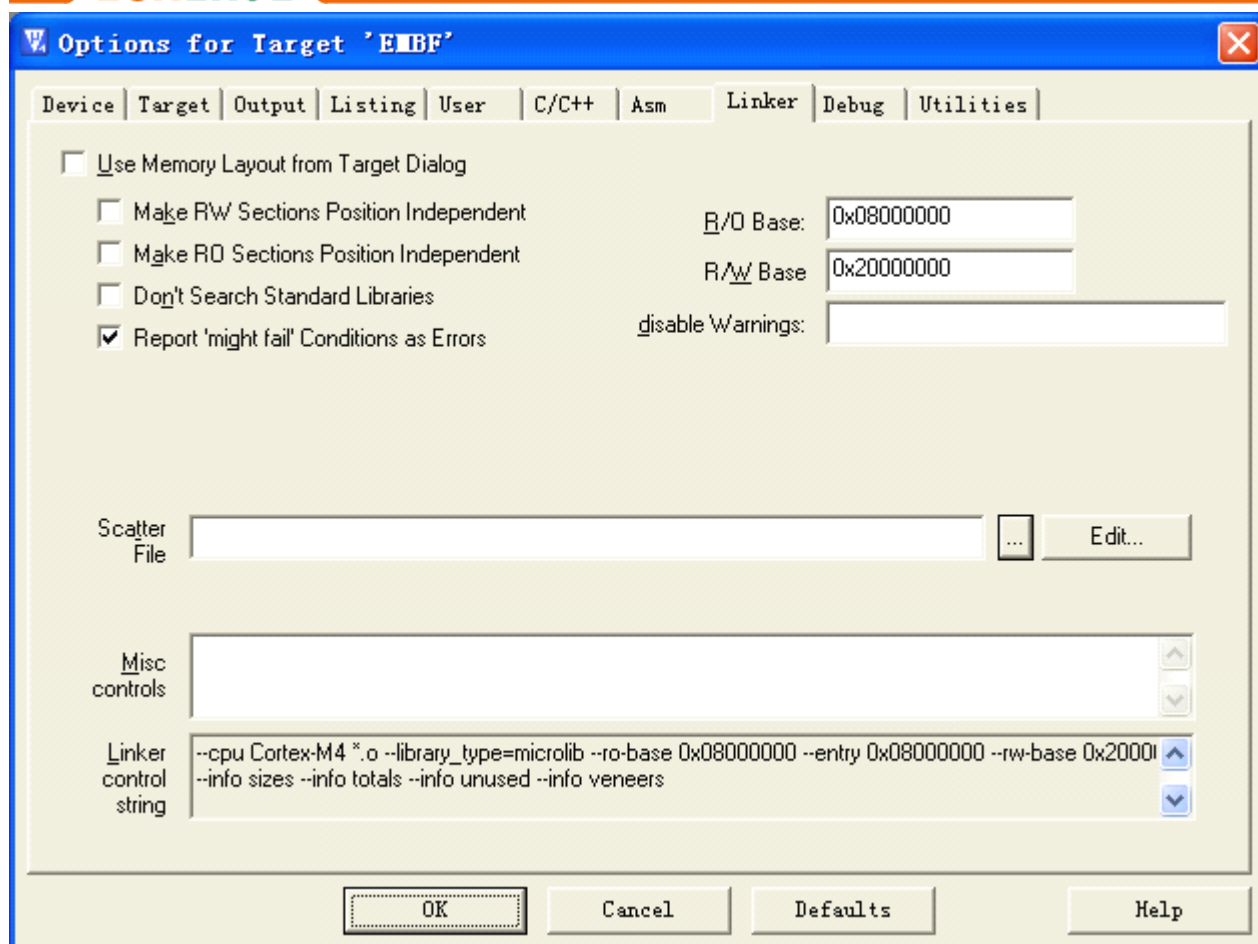


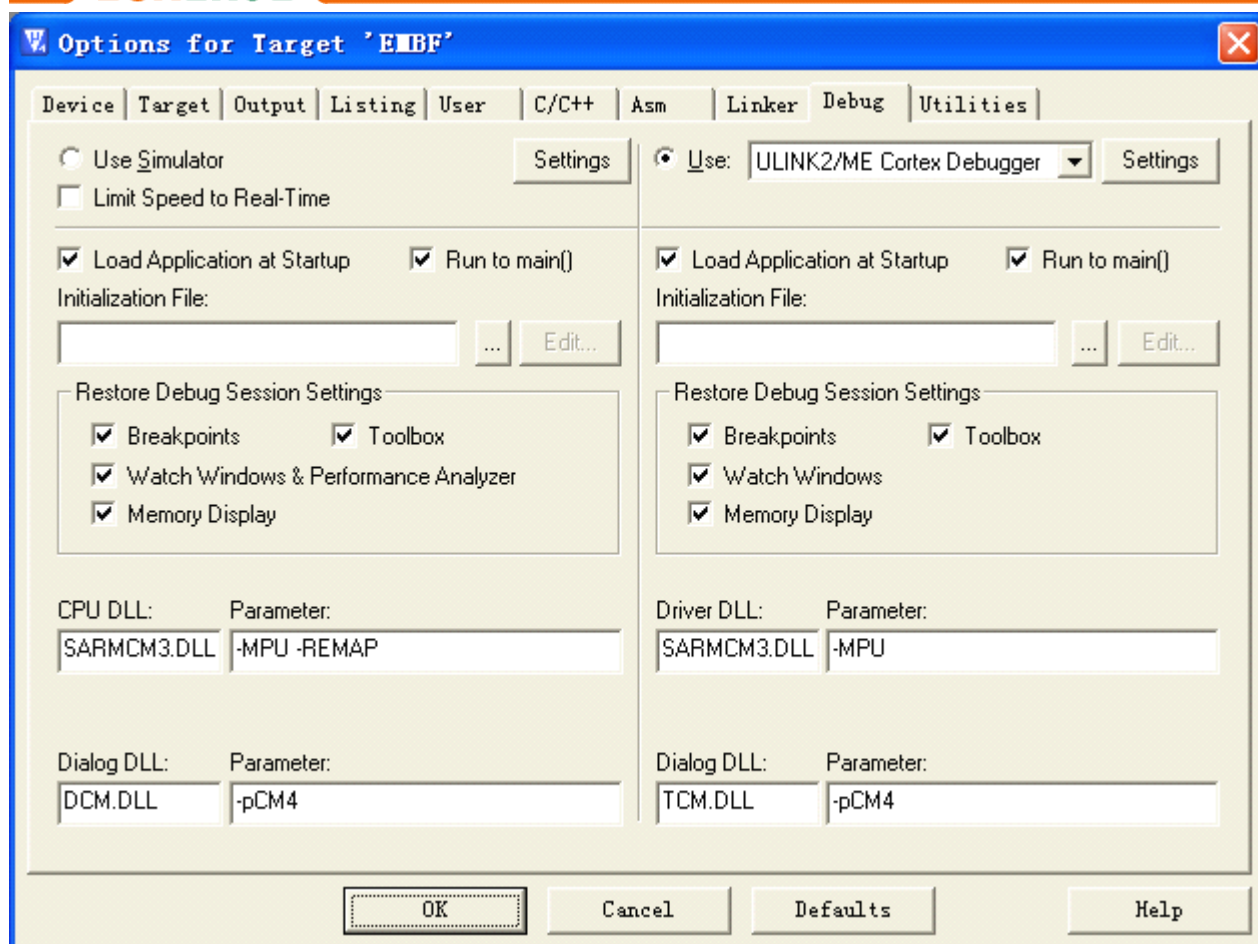


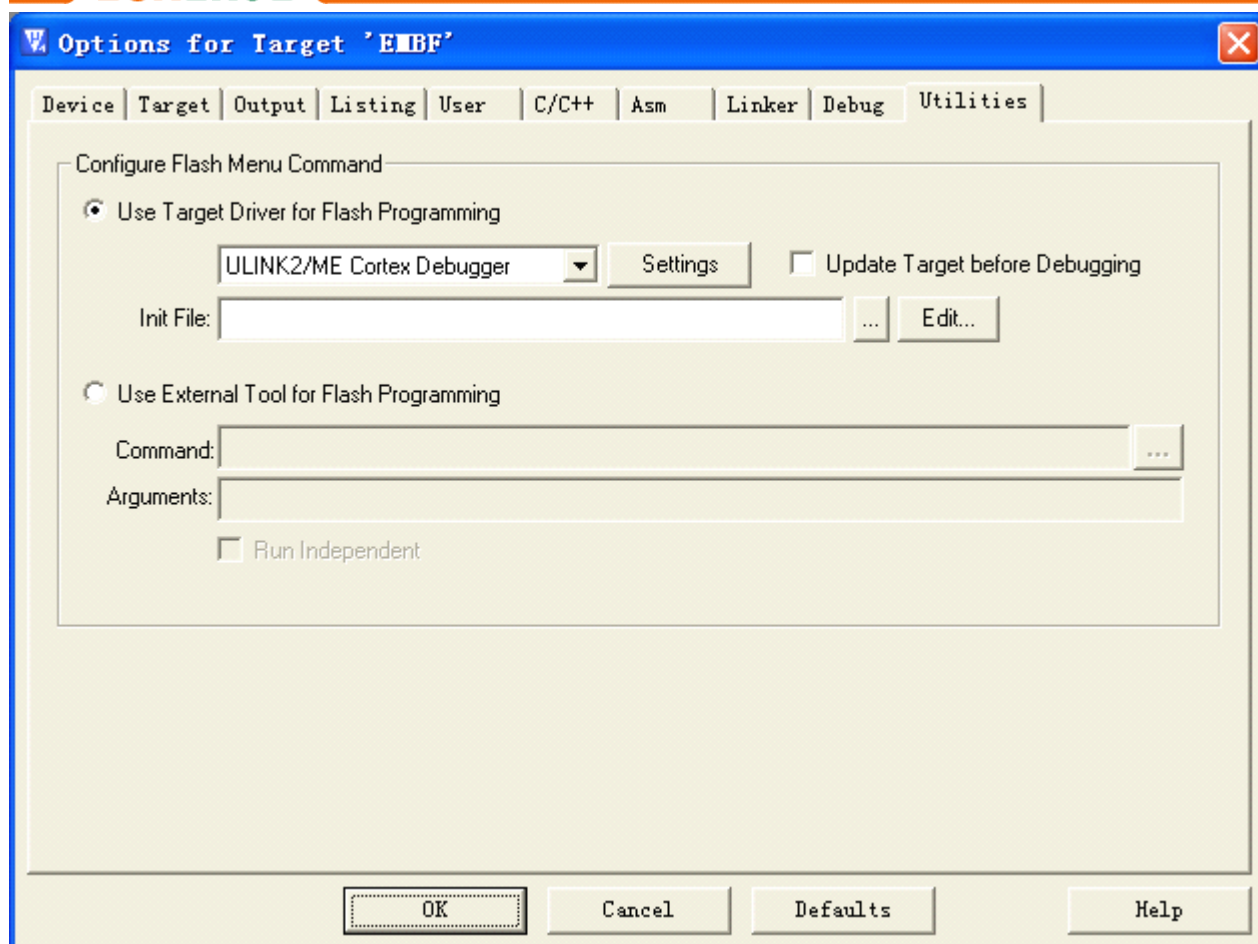












8、API 说明

assage 模块

1 void assage_create_sig(int32_t fd)

说明:

根据驱动句柄创建异步消息。此函数接口为驱动专用，不能用于其它功能。

参数说明:

fd -- 驱动程序文件句柄

2 uint32_t assage_accept(int32_t fd)

说明:

获取驱动消息。此函数接口为驱动专用，不能用于其它功能。

参数说明:

fd -- 驱动程序文件句柄

3 void assage_init(void)

说明:

异步消息初始化。异步消息使用的总数不能超过异步消息能够管理的最大范围，此数值由

ASSAGE_MAX_TASK 宏定义

参数说明:

无

4	<code>assage_handle assage_create_msg(t_assage_msg msg_buf[],uint8_t size,uint32_t mask)</code>
---	---

说明:

创建一个异步消息。

参数说明:

`msg_buf` -- 一个本地使用的消息缓冲, 此缓冲是由用户申请, 系统管理`size` -- 缓冲的个数, 注意这里并不是缓冲的字节数`mask` -- 消息的掩码, 用来定义消息类型的, 此掩码是用来处理定向消息时用的, 如果不需要处理定向消息可以设置此参数为 0xFFFFFFFF, 此时将接收和发送所有类型的消息。

返回值:

创建成功将返回一个非 0 的数值, 此数值是异步消息的操作句柄。

5	<code>void assage_post_msg(assage_handle handle,t_assage_msg *msg)</code>
---	---

说明:

向指定的队列中插入一条消息。

参数说明:

`handle` -- 消息句柄`msg` -- 具体发送的消息

6	<code>void assage_post_all (t_assage_msg *msg)</code>
---	---

说明:

发送一条广播消息, 但此消息会受消息掩码控制, 只有允许接收此消息的队列才能得到此消息。

参数说明:

`msg` -- 具体发送的消息

7	<code>void assage_get_msg(assage_handle handle,uint8_t timeout,t_assage_msg *lmsg)</code>
---	---

说明:

从队列中获取一条消息。

参数说明:

`handle` -- 消息句柄`timeout` -- 获取消息的延时。如果为 0 则永久等待一个消息, 为 255 时则不等待立刻返回, 其它数值则会等待相应的秒数后超时返回。`lmsg` -- 为本地消息缓冲的指针, 如果其中的 `msg` 成员为 0 则表示未得到任何消息, 否则得到了相应的信息。

8	<code>void assage_del_msg(assage_handle handle)</code>
---	--

说明:

从队列中删除一条消息。

参数说明:

`handle` -- 消息句柄

fcntl 模块

```
1 int32_t drv_open(const char*, int32_t, int32_t);
```

说明:

打开相应的驱动。

参数说明:

驱动文件名称

```
2 int32_t drv_read(int32_t, void*, uint32_t)
```

说明:

从驱动中读取数据，每个不同的驱动所读出的数据是不一样的，具体需要参考驱动的使用说明。

参数说明:

驱动句柄

缓冲区指针

缓冲区大小

```
3 int32_t drv_write(int32_t, const void*, uint32_t)
```

说明:

向驱动中写入数据，每个不同的驱动所写入的数据是不一样的，具体需要参考驱动的使用说明。

参数说明:

驱动句柄

缓冲区指针

缓冲区大小

```
4 int32_t drv_lseek(int32_t, int32_t, int32_t)
```

说明:

跳转读写指针到指定位置。

参数说明:

驱动句柄

偏移长度

起始位置

```
5 int32_t drv_close(int32_t)
```

说明:

关闭驱动。

参数说明:

驱动句柄

```
6 int32_t drv_ioctl(int32_t, int32_t, void *)
```

说明:

控制驱动的一些自定义选项，此功能不是所有驱动都会支持。

参数说明:

驱动句柄

数据类型，具体视驱动而定

数据指针

```
7 int32_t drv_poll(int32_t fd, void *ev)
```

说明:

驱动的异步数据赋值，此功能不是所有驱动都支持。

参数说明:

驱动句柄
异步消息的指针

lcd_module 模块 (续)

1 `bool LCD_open(TLCDModule *)`

说明:

lcd 液晶模块初始化函数。

参数说明:

液晶类接口。

2 `void (*clear)(void)`

说明:

清屏函数。

参数说明:

无

3 `void (*cleararea)(uint8_t,uint8_t,uint8_t,uint8_t,uint8_t)`

说明:

区域清屏函数。

参数说明:

列坐标

行坐标

宽度

高度

灰度, 这里可以按指定灰度清屏

4 `void (*setbacklight)(uint16_t)`

说明:

设置对比度。

参数说明:

对比度参数, 调整范围是 0 ~ 100

5 `void (*writechar)(uint16_t ,uint16_t ,const char *,uint16_t)`

说明:

向指定位置写一个字符。

参数说明:

行坐标

列坐标

字符串指针

具体字体

6 `void (*writestr)(uint16_t ,uint16_t ,const char *,uint8_t ,uint16_t)`

说明:

写英文字符串, 此函数不建议使用。

参数说明:

行坐标
列坐标
字符串指针
字符串长度
具体字体

7 void (*writestru)(uint16_t ,uint16_t ,const char *,uint8_t ,uint8_t)

说明:

写混合字符串，仅支持中英混合，此函数不建议使用。

参数说明:

列坐标
行坐标
字符串指针
长度
字体尺寸

8 void (*write_str)(TLCWStrParam *param)

说明:

写混合字符串，推荐使用此函数。

参数说明:

列坐标
字符串指针
长度
字体尺寸
反显索引
编码模式，0-西方字体，1-东方字体

9 void (*showimage)(uint8_t ,uint8_t ,uint8_t ,uint8_t ,const uint8_t *,short ,uint8_t)

说明:

显示图像。

参数说明:

行坐标
列坐标
宽
高
图像数据
灰度控制
标志，此参数未用

10 void (*drawline)(uint8_t ,uint8_t ,uint8_t ,uint8_t ,uint8_t)

说明:

显示图像。

参数说明:

列坐标
行坐标
长度
灰度

标志，LCD_DRAW_HORI -- 横线，LCD_DRAW_VERT -- 竖线

11	<code>void (*drawrect)(uint8_t,uint8_t,uint8_t,uint8_t)</code>
<p>说明： 画矩形。</p> <p>参数说明： 列坐标 行坐标 长度 宽度</p>	
12	<code>void (*drawrectarc)(uint8_t,uint8_t,uint8_t,uint8_t,uint8_t,uint8_t)</code>
<p>说明： 画矩形。</p> <p>参数说明： 列坐标 行坐标 长度 宽度 矩形大小 填充灰度</p>	
13	<code>void (*setfontcolor)(uint8_t,uint8_t)</code>
<p>说明： 设置字体颜色。</p> <p>参数说明： 前景色 背景色</p>	

说明：液晶显示最多支持 8 位灰度，对应的数值是 0~255

message 模块

1	<code>void InitMsg(PProcBuf Entry,PMsgProc GetMsg)</code>
<p>说明： 消息模块初始化。</p> <p>参数说明： Entry -- 队列缓冲区指针。 GetMsg -- 消息获取函数的函数指针</p>	
2	<code>void EnableMsgType(PProcBuf Entry,int MsgType,int Enabled)</code>
<p>说明： 设置消息掩码。</p> <p>参数说明： Entry -- 队列缓冲区指针。 MsgType -- 消息类型 Enabled -- 开启还是关闭此消息</p>	

3	<code>uint32_t SelectNewMsgMask(PProcBuf Entry,uint32_t newmsk)</code>
<p>说明： 设置新的消息掩码。</p> <p>参数说明： Entry -- 队列缓冲区指针。 Newmsk -- 新消息类型</p> <p>返回值： 返回原先的消息掩码</p>	
4	<code>int DoMsgProcess(PProcBuf Entry, void *Obj, int ExitCommand)</code>
<p>说明： 执行消息。</p> <p>参数说明 Entry -- 队列缓冲区指针。 Obj -- 参数 ExitCommand -- 推出循环的参数</p>	
5	<code>int RegMsgProc(PProcBuf Entry,PMsgProc MsgProc)</code>
<p>说明： 注册消息处理函数指针。</p> <p>参数说明 Entry -- 队列缓冲区指针。 MsgProc -- 消息函数的函数指针</p> <p>返回： 当前消息处理函数的索引号。</p>	
6	<code>int UnRegMsgProc(PProcBuf Entry,int index)</code>
<p>说明： 协助消息函数。</p> <p>参数说明 Entry -- 队列缓冲区指针。 Index -- 当前消息处理函数的索引号。</p>	
7	<code>int PostMessage(PProcBuf Entry,int Msg,int P1,int P2)</code>
<p>说明： 发送消息。</p> <p>参数说明 Entry -- 队列缓冲区指针。 Msg -- 消息 P1,P2 -- 消息传递的参数</p>	

utils 模块

1	<code>uint32_t embf_get_tick(void)</code>
<p>说明： 读取系统计时器数值，单位是 ms。</p> <p>参数说明： 无。</p>	

2	void embf_delay(uint16_t time)
说明： 延时函数。 参数说明： Time -- 延时时间 ms	
3	uint8_t embf_get_usage(void)
说明： Cpu 占用率。 返回值： Cpu 占用率，0-100%	
4	void embf_get_current_ver(char *buf,uint8_t size)
说明： 读取当前平台的版本号。 参数说明： buf -- 缓冲指针 Size -- 缓冲尺寸	

9、常见问题