

[REDACTED]

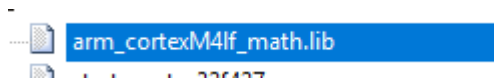
[REDACTED]

[REDACTED]

C 语言部分：

一、文件命名：

- 1、对于文件名的命名方式，除特定缩写必须采用大写之外，统一采用**小写**的方式，在出现多个单词需要进行间隔时，应采用下划线的方式，如 stm32 官方库文件所示：



二、头文件部分：

- 2、在写头文件时，首尾应采用

```
#ifndef  
#define  
#endif
```

的方式进行定义，定义的内容必须采用**全大写的方式**，单词之间必须采用下划线，如

app_Init.h 文件的定义方式应写为

```
#ifndef __APP_INIT_H  
#define __APP_INIT_H  
#endif
```

- 3、在头文件中有宏定义时，定义的目标**采用全大写**的方式，若出现多个单词需要区分时，则采用下划线的方式，如：

```
#define NUM_OF_MOTOR
```

- 4、在定义结构体时，除了引用官方库的定义方式外，所有结构体的定义方式应如图中所示的写法：

```
typedef struct {
    TaskHandle_t xHandleTask;
    BaseType_t taskEvent[LIST_OF_TASK];
    u16 rgbState;
    uint8_t taskState[LIST_OF_TASK];
    uint32_t loops;
} supervisorStruct_t;
```

name 改为所需要命名的英文单词，首字母采用**小写**，多个单词时采用**首字母大写**进行分隔，如 motorSerialStruct_t。

- 5、在**全局**数组变量中的下标，有明确含义的不应直接采用 0、1、2、3 的方式，应采用合适的代号，使用枚举的方式定义代号，代号必须全部**采用大写**，不同单词采用下划线隔开，如下图所示：

```
enum {
    IMU_TASK = 0,
    SUPERVISOR_TASK,
    DATA_LINK_TASK,
    CONTROL_TASK,
    LIST_OF_TASK
};
```

- 6、在需要对变量进行 extern 时，**应放在头文件中**，而不是放在 C 文件中。

- 7、对头文件进行撰写时，最多分为六部分，**按顺序**分别是：

- 1) 包含的头文件；
- 2) 宏定义；
- 3) 枚举与联合体；
- 4) 结构体；
- 5) Extern 变量；

- 6) 声明可以外用的公用函数;

三、C 文件部分

- 8、在 C 文件中申明变量时，除有需要初始定义单个或多个变量成员，如：

```
LedStruct_t ledFuntionData[LED_LIST]={
    {LED_DISWORK,LED_SLOW},
    {LED_WORK_RED,LED_SLOW},{LED_WORK_GREEN,LED_SLOW},{L
    {LED_WORK_RED,LED_NORMAL},{LED_WORK_GREEN,LED_NORMAL
    {LED_WORK_RED,LED_FAST},{LED_WORK_GREEN,LED_FAST},{L
};
```

其余全局变量的定义**均需采用结构体**的方式，目的是为了更方便可能的指针操作和链表操作。

- 9、对函数名进行定义时，除了现成使用的 BSP 层和官方库之外，首字母必须采用**小写的**
方式，多个单词时采用**首字母大写**进行分隔，如

```
void supervisorUpdateTask(void *Parameters){
```

- 10、 定义变量时，规则和函数名一致。如：

```
supervisorStruct_t supervisorData;
```

- 11、 在任何命名中，皆以**尽量准确直白的英文翻译为准，禁止采用拼音！**

- 12、 在定义普通变量时，应采用

uint8_t/uint16_t/uint32_t/int8_t/int16_t/int32_t/float/double/vs8/vs16/vs32/u8/u16/u32/s8/s16/s32 的方式。

- 13、 对代码进行书写时，严格遵循首行缩进**一个 Tab 的间距（两个空格）**，当一句代码超过一般 16*9 屏幕（最小 17 寸屏幕）的显示长度时，**必须要用\进行分行**。如：

```
void wiredControlInit(void){
    driver_dataLinkInit(DATA_LINK_USARTX,DATA_LINK_USARTX_TX_PIN,DATA_LINK_USARTX_RX_PIN,\
        DATA_LINK_USART_BAUD_RATE,DATA_LINK_USART_PRE_PRIORITY,DATA_LINK_USART_SUB_PRIORITY);
    supervisorData.taskEvent[DATA_LINK_TASK] = xTaskCreate(wiredControlUpdate,"DATA_LINK",\
        DATA_LINK_STACK_SIZE,NULL,\
        DATA_LINK_PRIORITY,&wiredControlData.xHandleTask);
}
```

- 14、 大括号的使用方式统一用 Kb&R 格式，如下图所示：

```
void wiredCommandReceive(u8 *array){
    if(array[0] == 0xAD && array[1] == 0x2C){
        if(!Verify_CRC8_Check_Sum(array,5)){
            //校验失败则不导入数据
            //暂时不进行操作，后续会向主机报错
        }
        else{
            wiredControlData.seq = array[3];
            //读取主机命令
            wiredReadCommad(array);
            digitalIncreasing(&wiredControlData.checkSeq);
        }
    }
}
```

方式为开括号总是与使用它的语句在同一行上，闭括号总是在它所关闭的语句下一行上。

- 15、 在单个函数中，如超过 3 个 if-else 或 for 循环嵌套（不包含三个），则**必须使用子函数增强可读性**。
- 16、 在两个函数之间**必须要留有一行增强可读性**。如：

```
controlData.robotMode = wiredControlData.cmd.robotMode;
}

void wiredReadCommad(u8 *array){
    u8 index = 0;
```

四、注释要求：

- 17、 严格要求注释，application 层需要有足够的注释，**特别是在重要逻辑判断处，**

必须要有中文注释。如：

```
if(array[0] == 0xAD && array[1] == 0x2C){  
    if(!Verify_CRC8_Check_Sum(array,5)){  
        //校验失败则不导入数据  
        //暂时不进行操作，后续会向主机报错  
    }  
}
```

- 18、 对暂时不使用的代码进行注释。

- 19、 **未维护完毕、为方便调试暂时有不符合代码规则的段落、但又不得不提交的代码**，必须施加类似以下注释**对同伴进行提示**：

```
/*=====   
=====  施工中，开始日期03/03/2018  =====   
=====  结束日期：未知  =====   
=====  请勿擅自修改此部分内容  =====   
=====*/
```

- 20、 条件同 19 点相同的代码段落，为了方便调试而使用的临时全局变量，允许暂时存在，在上传前**必须**在后注释有 `//临时调试变量`，**在调试工作结束后必须删除！**

五、代码迭代要求：

- 21、 在完成一次代码更改后，并需要提交到 git 仓库之前，**必须在 record 文件中标明更改人以及更改时间、更新内容。**
- 22、 在代码上传到 git 空间之前，**必须对已有错误和警告进行解决。（代码必须是 0 error(s) 0 warning(s) 的状态）**
- 23、 在需要对**含有公共空间的代码段**进行更改时，在以下区域内可以对代码进行自由更改：

```

/*****以下是自定义部分*****/
if(USART2_len){
    wiredCommandReceive(Array_USART2_RX);
}
/*****以上是自定义部分*****/

```


公共空间的文件首部都会有以下标记：


```

/*-----
-----
//      警告：公共代码区域，故障责任重大      //
//      除特定区域可以自由可以更改外，      //
//      其余部分更改，请与公共区域负责人取得更改权限  //
-----
-----*/

```

如果需要对**自定义部分以外的区域**进行修改时，需要向**公共空间管理者**进行申请，在**征得同意后**才可进行更改，并且在更改**公共空间**之后必须同步到 git 仓库。公共空间管理者的联系方式在 record 文件中有标出。

- 24、在单项测试任务结束后，需要及时对代码进行上传，并且在**上传前**在“钉钉”分组群中进行报告提示。
- 25、在上传代码之前，必须使用 keilkill  keilkill.bat 对已编译的链接文件进行释放，并且需要删除个人电脑的配置文件，如以下文件：

 ModularReader.uvguix.AGKODY	2019/1/4 2:05	AGKODY 文件	83 KB
---	---------------	-----------	-------

- 26、代码公共空间管理者需要**每天**对代码的更新情况进行一次调查，在有改动的情况下**每天同步一次**最新代码到 master 分支下。

请各位开发者务必遵守以上代码规范，以强迫症的精神对待公共代码。

