

# ACE 电控软件编程风格 - 目前仅供参考

---

这是一份东莞理工ACE实验室电控组的嵌入式开发指引。

作为ACE团队的一员，各个兵种需要由不同的人完成，但是这样的话没有一份统一的规范代码，每个人的代码风格都不一样，会导致很多东西明明是很简单的可以通用的，却要重新写一份代码，导致事倍功半、效率低下，所以我稍微总结的学长的建议、RT-Thread编程风格、大疆开源的代码风格和自己的习惯，汇总了一份代码规范参考

请每一位东莞理工ACE实验室电控组的嵌入式开发者遵守这样的编程风格。

同时对于使用新人，也可通过这份文档了解代码内部一些约定从而比较容易的把握到各个兵种的开发实战中去。

## 1.目录名称

---

目录名称如果无特殊的需求，请使用**全小写**的形式。

目录名称应能够反应部分的意思，例如各芯片移植由其芯片名称构成或芯片类别构成、components 目录下能够反映组件的意义。

## 2.文件名称

---

文件名称如果无特殊的需求(如果是引用其他地方，可以保留相应的名称)，请使用**全小写**的形式。另外为了避免文件名重名的问题，一些地方请尽量不要使用通用化、使用频率高的名称。

## 3.头文件定义

---

C语言头文件为了避免多次重复包含，需要定义一个符号。这个符号的定义形式请采用如下 的风格：

```
#ifndef __TASK_CHASSIS_H
#define __TASK_CHASSIS_H
/* header file content */
#endif /* __TASK_CHASSIS_H */
```

即定义的符号左侧采用 "\_\_" 以避免重名，另外也可以根据文件名中是否包含多个词语而 采用 "\_" 连接起来。

## 4.文件头、函数注释

在每个源文件文件头上，应该包括相应的信息，日期、：

```
/**
***** 东莞理工学院ACE 实验室 *****
* @file      文件名字.c/h
* @brief     简短介绍文件
* @note      笔记
* @history   历史信息版本
*
@verbatim
=====
用于建立一个框，来说明整个文件的功能。
=====
@endverbatim
***** 东莞理工学院ACE 实验室 *****
*/
```

下面是大疆的

```
/**
***** (C) COPYRIGHT 2019 DJI *****
* @file      chassis.c/h
* @brief     chassis control task,
*            底盘控制任务
* @note
* @history
* Version    Date          Author      Modification
* V1.0.0     Dec-26-2018    RM          1. done
* V1.1.0     Nov-11-2019    RM          1. add chassis power control
*
@verbatim
=====

=====
@endverbatim
***** (C) COPYRIGHT 2019 DJI *****
*/
```

函数注释采用这种类似的方式：

```
/**
* @brief      函数说明
* @param[in]  传入参数说明(1)
* @param[in]  传入参数说明(2)
* @retval     返回参数说明(1)
* @retval     返回参数说明()
* @attention  需要注意的地方
*/
```

例如采用如上的形式。

## 5.结构体定义

结构体名称请使用**小写英文名**的形式，单词与单词之间采用 "\_" 连接，例如：

```
typedef struct
{
    const motor_measure_t *chassis_motor_measure;
    fp32 accel;
    fp32 speed;
    fp32 speed_set;
    int16_t give_current;
} chassis_motor_t;
```

其中，"{" 独立占用一行，"}"后面**空一格**，接上结构体名称加上 "\_t" 的形式作为名称

结构体等的类型定义，例如：

```
// 底盘控制数据
static chassis_control_t chassis_control;
```

为了各个文件之间对象的访问，采用返回结构体地址的形式，在另外一个文件接收，例如：

```
/**
 * @brief          返回遥控器控制变量，通过指针传递方式传递信息
 * @param[in]      none
 * @retval         返回遥控器控制变量 &rc_ctrl
 * @attention
 */
const rc_ctrl_t *get_remote_control_point(void)
{
    return &rc_ctrl;
}
```

## 6.宏定义

请使用**大写英文**名称作为宏定义，单词之间使用 "\_" 连接，例如：

```
#define ACE_TRUE    1
```

## 7.函数名称、声明

函数名称请使用**小写英文**的形式，单词之间使用"\_"连接。提供给上层应用使用的API接口，必须在相应的头文件中声明；如果函数入口参数是空，必须使用 void 作为入口参数，例如：

```
/**
 * @brief          返回遥控器控制变量，通过指针传递方式传递信息
 * @param[in]      none
 * @retval         返回遥控器控制变量 &rc_ctrl
 * @attention
 */
const rc_ctrl_t *get_remote_control_point(void)
{
    return &rc_ctrl;
}
```

## 8.变量声明

最终代码**不要出现全局变量**，全局变量仅仅供调试，例如：

```
//底盘测试模式 宏定义 0 为不使用测试模式
#define CHASSIS_TEST_MODE    1

#if CHASSIS_TEST_MODE
int JSCOPE_chassis_1;
int JSCOPE_chassis_2;
int JSCOPE_chassis_3;
int JSCOPE_chassis_4;
int temp_1, temp_2, Can_send_time;
#endif

#if CHASSIS_TEST_MODE
static void chassis_jscope_print_curve(void); /* jscope打印曲线 */
#endif

#if CHASSIS_TEST_MODE
/* jscope打印曲线 */
static void chassis_jscope_print_curve(void)
{
    JSCOPE_chassis_1 = chassis_control.chassis_motor[0].output;
    JSCOPE_chassis_2 = chassis_control.chassis_motor[1].output;
    JSCOPE_chassis_3 = chassis_control.chassis_motor[2].output;
    JSCOPE_chassis_4 = chassis_control.chassis_motor[3].output;
}
#endif
```

局部变量可以出现简单的存储作用，例如：

```
uint8_t i; //用于for循环计数
```

```
for (int i = 0; i < 10; i++)
{
    /* 循环代码 */
}
```

重要的变量请在结构体内定义。

## 9. 注释编写

相应提示性的注释以解释一段复杂的算法它是如何工作的。对语句的注释只能写在它的**上方或右方**，其他位置都是非法的。某些函数比较复杂或者特殊可以在函数上方加上对这个函数的**详细解释**

## 10. 缩进及分行

缩进请采用**\*\* 4 个空格\*\***的方式。如果没有什么特殊意义，请在 "{" 后进行分行，并在下一行都采用缩进的方式，例如：

```
if (condition)
{
    ..... /* others */
}
```

**唯一的例外**是 switch 语句，switch-case 语句采用 case 语句与 switch 对齐的方式，例如：

```
switch (value)
{
    case value1:
        ..... break;
}
```

case 语句与前面的 switch 语句对齐，后续的语句则采用缩进的方式。分行上，如果没有什么特殊考虑，请**不要在代码中连续使用两个以上的空行**。

## 11. 大括号与空格

从代码阅读角度，建议**每个大括号单独占用一行**，而不是跟在语句的后面，例如：

```
if (condition)
{
    ..... /* others */
}
```

匹配的大括号单独占用一行，代码阅读起来就会有相应的层次而不会容易出现混淆的情况。

**空格建议在非函数方式的括号调用前留一个空格以和前面的进行区分，例如：**

```
if (x <= y)
{
    ..... /* others */
}

for (index = 0; index < MAX_NUMBER; index ++ )
{
    ..... /* others */
}
```

建议在括号前留出一个空格(涉及的包括 if、for、while、switch 语句)，而运算表达式中，运算符与字符串间留一个空格。另外，**不要在括号的表达式两侧留空格**，例如：

```
if ( x <= y )
{
    ..... /* other */
}
```

这样括号内两侧的空格是不允许的。

## 12.log信息

使用letter\_shell+elog进行调试。（最终代码删除、注释掉相关调试函数）

```
log_a("断言(Assert)"); // 断言(Assert)
log_e("错误(Error)"); // 错误(Error)
log_w("警告(Warn)"); // 警告(Warn)
log_i("信息(Info)"); // 信息(Info)
log_d("调试(Debug)"); // 调试(Debug)
log_v("详细(Verbose)"); // 详细(Verbose)
```

使用方式和printf函数类似

## 13.函数（模块化）

在编程中，函数应该尽量精简，仅完成相对独立的简单功能。函数的实现不应该太长，函数实现太长，应该反思能够如何修改(或拆分)使得函数更为精简、易懂，函数级层不要太多。

## 14.对象

采用了 C 语言对象化，命名表现形式，例如：

```
/*底盘电机数据*/
typedef struct
{
    const motor_measure_t *chassis_motor_measure;

    fp32 accel;
    fp32 speed;
    fp32 speed_set;
    fp32 position;
    int16_t output;
} motor_t;

static void mk_data_process(chassis_control_t *mk_data_process_f);           // 键盘鼠标选择
static void chassis_mode_choose(chassis_control_t *chassis_mode_choose_f); // 底盘模式选择
static void chassis_follow(chassis_control_t *chassis_follow_f);           // 底盘跟随
static void chassis_twist(chassis_control_t *chassis_twist_f);             // 底盘扭腰
static void chassis_rotation(chassis_control_t *chassis_rotation_f);       // 底盘小陀螺
static void chassis_independent(chassis_control_t *chassis_independent_f); // 底盘不跟随
```

## 15. 用 astyle 自动格式化代码

```
参数: --style=allman
      --indent=spaces=4
      --indent-preproc-block
      --pad-oper
      --pad-header
      --unpad-paren
      --suffix=none
      --align-pointer=name
      --lineend=linux
      --convert-tabs
      --verbose
```

## 16. 最后附上我的工程开头

```

/***** 东莞理工学院ACE实验室 *****/
* @project    工程名字
* @Author     东莞理工学院  ACE 机器人战队  蔡瀚源
* @Date       日期
* @Version    版本

P                               :u7  :Ii
QBQ                             sQBBQBB PBBBBQI.  XQBBBBBBBBBQBBBBBBBBBBBM
bbbBZ                          .MQBQBBBQB 5BBBBBBBbi uBBBBBBBBBQBBBBBBBBBQBBBP
bBBQQB5                        XBBBRQQBBBP sQBQBQBBBZ IBBBBBBBBBBBBBBBBBBBBBBD
rBBgRQBY                       BBQQRBBQr   rgBBBQr
. iBBgRQB7                     BBQRgBQ:      iE.
:BY 7BBgRQB:                   sBQMgBB
.BBB: uBBgRBB.                 BBMDQQ:
BBQBB: XBQgRBB                 QBggQB
BBQgBBB KBRDRBB               BBgDBB
BBQgRBB dQggQBB               BBggQB.
gBQgRQB BBggQDggQBQ           YBQDMBB
PBQgRBB BBBBBRQgMgQBg        BBQgRBB:      iZ:
2BQgMBB. BBBBBBBBQRgQBK      BBBRQBBL.    .rBBQBr
vQBgRQB: :uriiiiirBQQgBB1    XQBQQBBBBBE  uBQBQBQBBBD  SBBBBBBBBBBBBBBBBBBQBBBD
7QBQBBr  :BBBQBBY             .ZBQBBBBBB qBBQBBBBB:  UBBBBQBBBBBBBBBBBBBBBBBd
LBBBBBBJ 7BBBBBQu             YRBBBQB  KBBBBBJ.  IBQBBBBBQBBBBBBBBBBBBBBBZ

                               7i .7.
*****/
/***** 东莞理工学院ACE实验室 *****/
```

有任何疑问，或者觉得要改进的地方请提出来，最后组内讨论决定修改该文档