

# 创e联 CREELINKS

——代码移植及二次开发指南

版本：V1.0

日期：2017-04-03

编写：北京大信科技有限公司

## 目录

第 1 章 飞控算法的移植.....	3
1.1 CREELINKS 开源无人机软件框架 .....	3
1.2 未更换硬件模块的移植方法.....	3
1.3 更换硬件模块后的移植方法.....	4
1.3.1 更换处理器.....	4
1.3.2 更换其它传感器等模块.....	5
第 2 章 无人机的二次开发.....	5
第 3 章 移植时其它需配置的参数.....	5

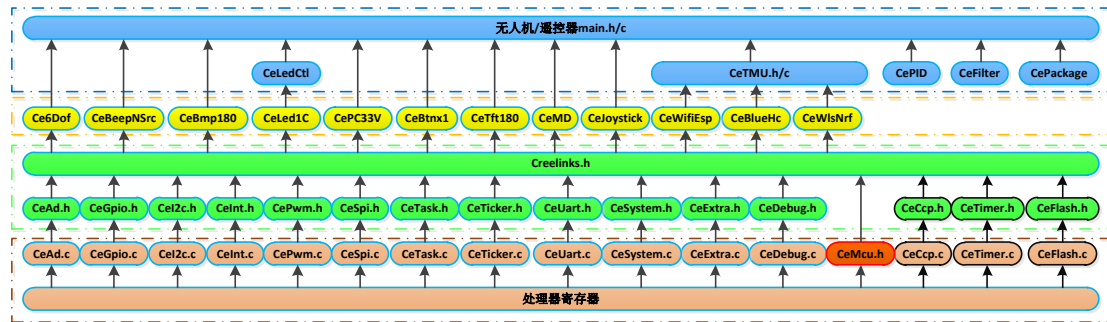
## 第1章 飞控算法的移植

主要介绍如何将基于 CREELINKS 平台的无人机飞控算法，无缝移植到其它如穿越机、DIY 的小四轴、FPGA/DSP/ARM/51/ Arduino 开发板。

移植前请先阅读文章《无人机软件框架设计》，下载地址：<http://www.creelinks.com/uav>。

### 1.1 CREELINKS 开源无人机软件框架

选看一下无人机软件框架：



从下到上，主要分为四个部分：

- 1) 棕色部分：接口实现层。
- 2) 绿色部分：硬件抽象层。
- 3) 黄色部分：模块驱动库。
- 4) 蓝色部分：小四轴无人机主控制程序。

详细内容请参考文章：《无人机软件框架设计》。

### 1.2 未更换硬件模块的移植方法

CREELINKS 小四轴无人机基于 STM32F103 最小系统设计的，如果考虑不换模块，即仍使用原来的 MPU6050、BMP180 芯片、STM32F103 处理器等，移植方法如下：

只需要按自己的 PCB 设计，重新指定模块使用的资源号即可。针对某一处理器，其所有的资源号均在 CeMcu.h 中 CE\_RESOURCE 枚举中定义。如 PC6GIP，指此处理器的 PC6 号引脚，此引脚包含 Gpio、Int、Pwm 三种功能。更多有关 CREELINKS 平台的介绍，请访问 <http://www.creelinks.com>

移植所做工作主要分为以下几部分：

- 1) 无人机代码移植

#### ➤ CeTMU.c

在 ceTMU\_initial()函数中，修改三个函数中的资源号即可：

```
ceWifiEspOp.initial(&(ceTMU.ceWifiEsp), Uart3);
ceWlsNrfOp.initial(&(ceTMU.ceWlsNrf), Spi2, PB11GIP, PB10GIP);
ceBlueHcOp.initial(&(ceTMU.ceBlueHc), Uart1, PB0AGIP, PB1AGIP, PB9GI);
```

其中粗体红色即为资源号。

#### ➤ main.c

在 initialModule()中，根据自己原理图边线方式修改以下函数中的资源号：

```
ceMDOp.initial(&(ceMD0), PC6GIP);
ceMDOp.initial(&(ceMD1), PC7GIP);
ceMDOp.initial(&(ceMD3), PC9GIP);
ceMDOp.initial(&(ceMD2), PC8GIP);
ceBeepNSrcOp.initialByGpio(&ceBeepNSrc, PD2CGI);
ceLedCtlOp.initial(PC0AGI, PC1AGI, PC2AGI, PC3AGI);
ceBtnx1Op.initialByGpio(&ceBtnLeft, PA11GIP, CE_NULL);
```

```
ceBtnx1Op.initialByGpio(&ceBtnRight, PA12CGI, CE_NULL);
ce6DofOp.initial(&ce6Dof, I2c1);
ceBmp180Op.initial(&ceBmp180, I2c1);
cePC33VOp.initial(&cePC33V, PC4AGI);
```

其中粗体红色即为资源号。

## 2) 遥控器代码移植

### ➤ CeTMU.c

在 ceTMU\_initialByWifi() 修改红色粗体部分：

```
ceWifiEspOp.initial(&(ceTMU.ceWifiEsp), Uart1);
ceTMU_initialByNrf() 修改红色粗体部分：
ceWlsNrfOp.initial(&(ceTMU.ceWlsNrf), Spi1, PA11GIP, PA12CGI);
ceTMU_initialByBlue() 修改红色粗体部分：
ceBlueHcOp.initial(&(ceTMU.ceBlueHc), Uart1, PB8GIP, PB9GIP, PB7GIP);
```

### ➤ main.c

在 initialModule () 修改红色粗体部分：

```
ceTft180Op.initial(&ceTft180, Spi2, PB9GI, PB10GIP, PB11GIP);
ceLed1COp.initialByGpio(&ceLed1B, PC11GI);
ceLed1COp.initialByGpio(&ceLed1C, PC12GI);
ceLed1COp.initialByGpio(&ceLed1D, PC13GI);
ceBeepNSrcOp.initialByGpio(&ceBeepNSrc, PA1AGIP);
cePC33VOp.initial(&cePC33V, PA0ACGIP);
ceJoystickOp.initial(&ceJoystickLeft, PC0AGI, PC1AGI, PC2AGI);
ceJoystickOp.initial(&ceJoystickRight, PC3AGI, PC4AGI, PC5AGI);
ceBtnx1Op.initialByGpio(&ceBtnS2A, PC6GIP, CE_NULL);
ceBtnx1Op.initialByGpio(&ceBtnS2B, PC7GIP, CE_NULL);
ceBtnx1Op.initialByGpio(&ceBtnS2C, PC8GIP, btnS2cCallBack);
ceBtnx1Op.initialByGpio(&ceBtnS2D, PC9GIP, CE_NULL);
```

## 1.3 更换硬件模块后的移植方法

### 1.3.1 更换处理器

更换处理器，需要实现 CREELINKS 标准接口，可以有以下两种方式：

- 1) 使用现有工程模板。
  - a) 访问 <http://www.creelinks.com/stdlib> 网站或 <http://www.creelinks.com/bbs> ，查找有无已经移植到你所需要的处理器的标准库，即针对你的处理器的工程模板。
  - b) 如果有，则下载并打开工程，将当前小四轴无人机中的 UserCode 和 ModuleLib 文件夹，复制并添加到新下的工程模板内即可。
  - c) 再根据前一章节所提示，修改模块使用的处理器资源号即可。
- 2) 自己实现 CREELINKS 标准接口。
  - a) 访问 <http://www.creelinks.com/stdlib> 网站，下载未移植版本的 CREELINKS 平台库文件。
  - b) 根据自己无人机对资源的需求，在 CeMcu.h 中定义此处理器的资源号，并在 \*.c 文件对未添加代码有函数进行代码补充即可。

有关 CREELINKS 标准库的移植，这里不做详细说明，可访问官方网站或论坛查看相关教程。

- c) 新建工程，将移植完成的库、及四轴无人机中的 UserCode 和 ModuleLib 文件夹，复制并添加到工程内即可。
- d) 再根据前一章节所提示，修改模块使用的处理器资源号即可。

**提示：**可将自己移植的库，更新到 <http://www.creelinks.com/bbs> 内，供更多的人学习与使用。

### 1.3.2 更换其它传感器等模块

如将 MPU6050 六轴传感器模块，更换为 MPU9250 轴传感器，只需将 ModuleLib 文件夹中的 Ce6Dof.c 及 Ce6Dof.h 文件中的内容替换修改即可。

CREELINKS 模块化编程思想，可以保证单独修改一个模块的驱动库\*.c 文件，对整个软件其它部分不会产生任何影响。

**注意：**请保证移植后的参数单位及符号的一致性，如静止放置时，MPU6050 驱动输出的 Z 轴加速度为负值，而经过更换为 MPU9250 后，也要输出为负值。MPU6050 角速度输出单位为度/s，由 MPU9250 的驱动也要保证单位为度/s。

## 第2章 无人机的二次开发

请直接访问 <http://www.creelinks.com/uav> 下载 CREELINKS 小四轴无人机的原理图及 PCB、源代码，在此基础上直接进行二次开发。

有关二次开发与移植的介绍，前几单已经讲得很清楚了，这里不再说明。

## 第3章 移植时其它需配置的参数

- 1) CeMD.h 文件中，根据自己的电调，来确定如下几个参数的值：
 

```
/*!< 根据驱动的电调类型，定义 Pwm 是否采用反向输出方式*/
#define CE_MD_REVERSE

/*!< 根据驱动的电调类型，定义 Pwm 输出的最大周期，单位 Ns*/
#define CE_MD_MAX_PWM_CYCLE_NS 50000

/*!< 根据驱动的电调类型，定义 Pwm 输出的最小周期，单位 Ns*/
#define CE_MD_MIN_PWM_CYCLE_NS 0

/*!< 根据驱动的电调类型，定义 Pwm 输出的最大占空比，单位 Ns*/
#define CE_MD_MAX_PWM_DUTY_NS 50000

/*!< 根据驱动的电调类型，定义 Pwm 输出的最小占空比，单位 Ns*/
#define CE_MD_MIN_PWM_DUTY_NS 0
```
- 2) CeTMU.h 文件中，根据所有环境，定义如下几个参数值：
 

```
/*!< 配置无人机需要连接 WIFI 的 SSID*/
#define CE_TMU_WIFI_SSID "Darcern"

/*!< 配置无人机需要连接 WIFI 的密码*/
#define CE_TMU_WIFI_PWD "Dxwzh178"

/*!< 配置无人机需要连接服务器的 IP*/
#define CE_TMU_WIFI_SERVER_IP "192.168.1.244"

/*!< 配置无人机需要连接服务器的端口*/
#define CE_TMU_WIFI_SERVER_PORT 2121
```