创e联

CREELINKS

——无人机软件框架

版本：V1.0

日期：2017-04-04

编写：北京大信科技有限公司

目录

第1章 前言

由于基于CREELINKS平台的模块化、对象化设计思想编写的代码框架，已经很容易理解并消化，加上源代码中，基本每一行均有详细注释，故此文主要从整体上介绍无人机的软件框架，而详细部分，各参考源代码、或文章。

第2章 什么是CREELINKS物联网开发平台

要了解小四轴无人机的软件结构，在了解CREELINKS物联网开发平台是什么。先了解一下Arduino平台。

Arduino是一款便捷灵活、方便上手的开源电子原型平台。包含硬件（各种型号的Arduino板）和软件（Arduino IDE)。由一个欧洲开发团队于2005年冬季开发。其成员包括Massimo Banzi、David Cuartielles、Tom Igoe、Gianluca Martino、David Mellis和Nicholas Zambetti。它构建于开放原始码simple I/O介面版，并且具有使用类似Java、C语言的Processing/Wiring开发环境。主要包含两个主要的部分：硬件部分是可以用来做电路连接的Arduino电路板；另外一个则是Arduino IDE，你的计算机中的程序开发环境。你只要在IDE中编写程序代码，将程序上传到Arduino电路板后，程序便会告诉Arduino电路板要做些什么了。

Arduino能通过各种各样的传感器来感知环境，通过控制灯光、马达和其他的装置来反馈、影响环境。板子上的微控制器可以通过Arduino的编程语言来编写程序，编译成二进制文件，烧录进微控制器。对Arduino的编程是利用Arduino编程语言(基于Wiring)和Arduino开发环境(基于Processing)来实现的。基于Arduino的项目，可以只包含Arduino，也可以包含Arduino和其他一些在PC上运行的软件，他们之间进行通信(比如Flash, Processing, MaxMSP)来实现。

CREELINKS平台与Arduino平台基本相似，但比Arduino平台更加强大。

相同点：

- 提供统一的处理器硬件资源操作接口API
- 提供大量的模块及其驱动库
- 软件硬件均开源
- 均将硬件模块、及处理器接口资源抽象为对象
- 面向对象编程，事件回调机制

不同点：

- CREELINKS使用嵌入式软件标准C语言结构，且不依赖于任何编译器，可以任何IDE环境下开发；Arduino采用C++语言，有自己的编译器，通常仅在Arduino提供的IDE下开发
- CREELINKS支持任何处理器型号；而Arduino支持很小一部分处理器型号
- CREELINKS用于工业控制、物联网、智能家居等实际产品的应用；Arduino适用于中小大学生学习、娱乐。
- CREELINKS平台特色是模块化编程，在提供大量的模块及驱动库的同时，还提供海量模块化的模块原理图，可直接用于产品设计（http://www.creelinks.com/bbs 下载）。
- 基于CREELINKS平台开发的嵌入式软件，不做修改，或少量修改（若使用了与处理器平台相关的库），即可移植到其他处理器平台，真正做到一码多用，无缝移植。
- Arduino平台的所有模块及其驱动，不做修改或稍做修改，即可完整运行在CREELINKS平台。

一句话概况CREELINKS平台：

不同处理器平台核心库+模块原理图/成品/驱动+嵌入式操作系统+大量物联网解决方案。

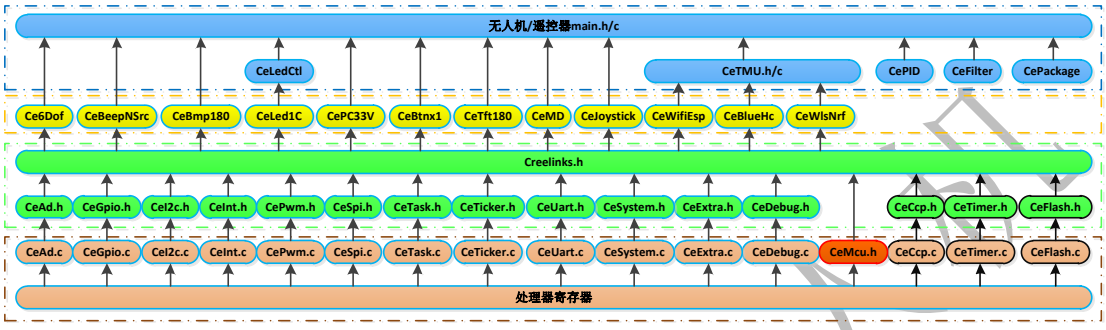
详细可访问 http://www.creelinks.com 了解更多。

第3章 小四轴软件框架

目前小四轴无人机采用的是 CREELINKS 平台基于 STM32F103 系列的处理器的核心库，有读者可能会问到，ARM 为 STM32F103 提供的库已经很成熟，且相关资料文档较多，为什么还要采用其它的库？

这要说到此小四轴无人机的目的之一是，上层控制与底层处理器完全无关，用户拿到此无人机的开源代码，只要修改少量底层接口，即可轻松移植到其它处理器平台。

下图为无人机软件框架图。



从下到上，主要分为四个部分：

1) **棕色部分：**接口实现层。即 CREELINKS 平台标准接口 API，基于不同处理器平台的具体实现。
2) **绿色部分：**硬件抽象层。即 CREELINKS 平台标准库，即一些*.h 文件的合集，在任何处理器平台，这些文件中的接口全部相同（CeMcu.h 比较特殊，这里不详细介绍）。如果需要移植到一个新的处理器平台，如 STM32F407 或其它 51 处理器，只需实现这些*.h 文件中的接口即可。
3) **黄色部分：**模块驱动库。即基于 CREELINKS 平台库，实现的模块驱动库，如此这些模块，则可完全独立于任何处理器。
4) **蓝色部分：**为用户设计的商品软件，在这里指小四轴无人机主控制程序。

1) 第 1、2 部分。

序号	模块型号	说明
1	CeAd	处理器 Ad 资源抽像接口
2	CeGpio	处理器 Gpio 资源抽像接口
3	CeI2c	处理器 I2c 资源抽像接口
4	CeInt	处理器 Int 资源抽像接口
5	CePwm	处理器 Pwm 资源抽像接口
6	CeSpi	处理器 Spi 资源抽像接口
7	CeTask	处理器 Task 任务管理
8	CeTicker	处理器 Ticker 任务管理
9	CeUart	处理器 Uart 资源抽像接口
10	CeSystem	处理器 System 资源抽像接口
11	CeExtra	不基于任何处理器平台的常用操作
12	CeDebug	不基于任何处理器平台的调试打印操作
13	CeMcu	与处理器平台相关的参数配置
14	CeCcp	处理器 Ccp 资源抽像接口
15	CeTimer	处理器 Timer 定时器资源抽像接口
16	CeFlash	处理器 Flash 资源抽像接口

重点说明下：

CeTask：

裸奔下：即有一个函数指针链表，当调用 ceTaskOp. registerTask()函数时，会向此链表中添加一个函数。

链表中所有的函数，均在 main 函数的 while(1)循环中的 ceTaskOp.mainTask()

中依次执行。

RTOS 下：链表不存在，每当调用 ceTaskOp. registerTask()函数时，都会向操作系统注册一个新的任务。

CeTicker：

定时器任务。即有一个函数指针链表，当调用 ceTickerOp. registerTicker ()函数时，会向此链表中添加一个函数。

系统在初始化 CeTicker 时，会提供一个定时器，此定时器在中断中，每隔 CE_TICKER_CALL_TIME_MS（CeMcu.h 中定义）时间，便会扫描一次链表，寻找需要执行的任务。

由于无论在裸奔还是 RTOS 下，链表中的内容均工作在中断环境内，故注册的定时器任务函数内，不要有大量耗时操作。

注：在每个 CeXx.h 文件的最后，均有此功能的使用示例。

有关 1、2 这里仅做简单介绍，详细可参考 CREELINKS 平台手册（http://www.creelinks.com/stdlib ）。

2) 第 3、4 部分

模块驱动库：

序号	模块型号	说明
1	Ce6Dof	基于 MPU6050 的加速度/角速度传感器模块
2	CeBeepNSrc	无源蜂鸣器模块
3	CeBmp180	基于 BMP180 的大气压力传感器模块
4	CeLed1C	单色 LED 模块
5	CePC33V	电压测量模块
6	CeBtnx1	单个按钮模块
7	CeTft180	1.8 寸 TFT 显示屏模块
8	CeMD	直流电机驱动模块
9	CeJoystick	带按钮的遥杆模块
10	CeWifiEsp	基于 ESP8266 的 WIFI 模块
11	CeBlueHc	基于 HC-05 的蓝牙模块
12	CeWlsNrf	基于 NRF24L01+的射频传输模块

CeLedCtl：根据当前无人机工作方式，控制四个 LED 闪烁方式。

内部注册了一个 CeTicker 定时器任务，每 10ms 就会更新一次四个 LED 的显示方式，详细可参考源代码。

CePackage：传输数据的打包与解包。

内有两个结构体，CePackageSend 和 CePackageRecv。

当调用 cePackageOp.dealSend()时，将 CePackageSend 结构体中的参数，打包并返回成带帧头、帧尾、校验合的 byte 型数组。

当调用 cePackageOp.dealRecv()时输入一个 byte 型的数组，CePackage 会自动校验此数组的帧头、帧尾、校验合，并将解析后的值更新到 CePackageRecv 结构体中。

详细可参考源代码，这里不作详细说明。

CeTMU：数据发送与接收管理。

主要功能为：发送并接收数据。

内部定义并初始化如 WIFI、蓝牙、2.4G 射频模块对象。

将 cePackageOp.dealSend()函数返回的 byte 型数组，通过通讯模块发送出去。

中断等待(2.4G 射频)、或在注册的 CeTask 任务中循环检测是否接收到数据。如果接收到数据，将读取的 byte 型数组，通过调用 cePackageOp.dealRecv()将数据进行解包，并更新到 CePackageRecv 结构体内。

CeFilter： 一阶、二阶、四元数、卡尔曼滤波/姿态解片算法。

主要功能为：通过调用函数 ceFilterOp. Filter()，传入当前未经过任何处理的三轴角速度、三轴加速度、和程序执行周期 dt，经过一种（可通过地面站选择哪种姿态解算/滤波算法）算法，将解算出的当前姿态角，更新到对应变量内。

CePID： 串级 PID 控制算法。

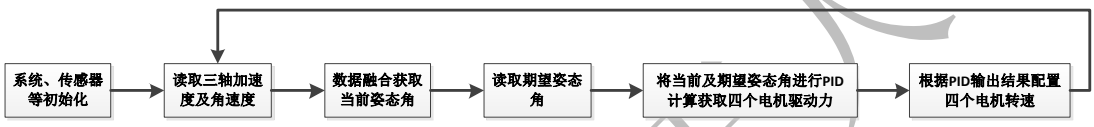
主要功能为：输入当前无人机状态（Pitch/Roll/Yaw 角、高度、油门），和期望的无人机状态，CePID 会自动根据当前与期望的状态，进行串级 PID 计算、定高飞行计算、阻尼油门计算等，返回四个电机的驱动强度信息。

只需再将此四个电机的驱动强度信息转换为 PWM 的占空比，更新到输出的 PWM 中即可。

第4章 小四轴软件执行流程

CREELINKS 平台小四轴无人机的核心飞控算法，是在主 main 函数中循环执行，而非中断中定时调用。

先看软件的基本流程框图：



再附上无人机 main 函数：

```
/**
  * @brief   CREELINKS 平台主入口函数(裸奔)
  * @return 0
  */
int main(void)
{
    ceSystemOp.initial();                                   //Creelinks 环境初始化
    ceDebugOp.initial(Uart4);                               //通过 Uart 串口输出 Debug 信息到上位机

    initialParment();                                        //结构体的参数初始化
    initialModule();                                        //初始化所有功能模块
    while (1)
    {
        ceTaskOp.mainTask();                                //Creelinks 环境主循环任务，请保证此函数能够被周期调用

        acc = ce6DofOp.getAcceleration(&ce6Dof);            //获取当前无人机加速度
        gyr = ce6DofOp.getGyroscope(&ce6Dof);               //获取当前无人机角速度
        ceEnvirmont = ceBmp180Op.getEnvironmentAsync(&ceBmp180);//获取当前无人机大气压相关数据

        ceNowAcc.x = acc->x ;                               //转存加速度数据
        ceNowAcc.y = acc->y ;
        ceNowAcc.z = acc->z ;

        ceNowGyr.x = -gyr->x;                               //转存角速度数据
        ceNowGyr.y = gyr->y;
        ceNowGyr.z = gyr->z;

        ceNowAngles.altitude = ceEnvirmont->altitude;   //当前高度暂存

        //对当前加速度、当前角速度进行姿态解析及滤波，以获取无人机姿态角数据
        ceFilterOp.filter(&ceNowAcc,&ceNowGyr,&ceNowAngles,dtS);
        //根据当前加速度及角速度、姿态角、还有期望姿态进行串级 PID 运算，并获得四个电机的驱动强度
        ceDrivePower = cePIDOp.calculate(&ceNowAcc, &ceNowGyr, &ceNowAngles, &ceHopeAngles,dtS);

        //配置第 0 路电机驱动强度，0~1000
        ceMDOp.setDriverPower(&ceMD0,(uavStatus != UAV_STATUS_READY)? (ceDrivePower->driverPower0):0);
        //配置第 1 路电机驱动强度，0~1000
        ceMDOp.setDriverPower(&ceMD1,(uavStatus != UAV_STATUS_READY)? (ceDrivePower->driverPower1):0);
        //配置第 2 路电机驱动强度，0~1000
        ceMDOp.setDriverPower(&ceMD2,(uavStatus != UAV_STATUS_READY)? (ceDrivePower->driverPower2):0);
        //配置第 3 路电机驱动强度，0~1000
        ceMDOp.setDriverPower(&ceMD3,(uavStatus != UAV_STATUS_READY)? (ceDrivePower->driverPower3):0);
```

```
        sendStatusToCtl();                  //整合当前数据，并发送给控制端
        checkConnectStatus();               //检测连接是否断开，如果断开，无人机姿态规中并缓慢下降
        calSystemRunCycle();                //计算程序执行周期时间
        checkTurnOver();                    //检测无人机是否翻机，以保护 MOS 管及电机
    };
}
```

1) 系统、传感器模块等初始化

仅执行一次，其中：

ceSystemOp.initial()：

主要完成 CREELINKS 平台系统的初始化操作，如时钟配置、全局中断配置等。

ceDebugOp.initial(Uart4)：

主要用来实现无人机将调试信息通过串口打印到上位机，以供观察。

initialParment()：

由于 CREELINKS 平台思想即模块化、对象化，故整个无人机解决方案中，每一个功能均为一个对象，而对象采用结构体方式来实现，故基本上整个解决方案中没有全局变量。而此函数即初始化定义的对象结构体中的参数。

initialModule()：

初始化无人机使用到的所有模块，如 MPU6050、BMP180、蓝牙等等。需要注意的是，如果处于 WIFI 和蓝牙工作状态，则需要建立 WIFI 连接后，此函数才会返回。而在 2.4G 射频模式下，是不进行连接检查，而直接返回。

2) 读取期望姿态角

这里的姿态角，指的是 CeAngles 结构体，定义如下：

```
/*
  *结构体，姿态角结构体
  */
typedef struct
{
    fp32 roll;              /*!< 翻滚角*/
    fp32 picth;             /*!< 俯仰角*/
    fp32 yaw;               /*!< 偏航角*/
    fp32 altitude;          /*!< 海拔高度*/
    fp32 accelerator;       /*!< 上升与下降的油门*/
}CeAngles;
```

定义了两个全局对象，分别是 ceNowAngles 和 ceHopeAngles。从字面就可以理解，即 ceNowAnges 指当前无人机的状态信息，由 CeFilter 对像根据当前无人机的加速度及角速度计算而出；而 ceHopeAngles 指控制无人机的控制端所期望的无人机的状态，由地面站、手机 APP、遥控器发送来的数据经过处理得出。

前面说过,ceTaskOp.mainTask()里，在裸奔情况下，会依次调用 CeTask 对象的函数链表，而 CeTMU(传输管理)的初始化过程中，需要提供一个回调函数：

CE_STATUS (***initial**)(CE_TUM_USE useType, void (***recvCallBack**)(uint8* recvBuf, uint16 recvCount));

而这个回调函数，即注册到了 CeTask 对象的链表里，在每个 ceTaskOp.mainTask()函数的执行过程中，均会检测有无接收到数据，如果有，则自动调用回调，并输入接收到的 byte 数组。

```
/**
  * @brief   TMU 数据传输管理模块接收数据后，调用的回调函数
  * @param   dataBuf:接收到数据的缓存地址
  * @param   dataCount:接收到数据的长度
  * @return  无
  */
void recvDataCallBack(uint8* dataBuf, uint16 dataCount)
{
        …
        //对接收到的数据进行解析及拆包处理，并更新参数到 ceHopeAngles 对象内。
        …
}
```

有关接收回调中，对接收到的参数处理为期望姿态角的过程，可参考源代码，这里不作详细说明。

3) 读取三轴加速度及角速度

acc = ce6DofOp.getAcceleration(&ce6Dof)：

从 MPU6050 中读取当前未经过处理的三轴加速度数据。其中 acc 是加速度结构体，内有参数 x/y/z，可参考源代码。

gyr = ce6DofOp.getGyroscope(&ce6Dof)：

从 MPU6050 中读取当前未经处理的三轴角速度数据。其中 gyr 是角速度结构体，内有参数 x/y/z，可参考源代码。

ceEnvirmont = ceBmp180Op.getEnvironmentAsync(&ceBmp180)：

读取由 BMP180 进行计算而来的当前环境数据，ceEnvirmont 是环境结构体，内有当前大气压、温度、海拔高度信息。

```
ceNowAcc.x = acc->x ;
ceNowAcc.y = acc->y ;
ceNowAcc.z = acc->z ;

ceNowGyr.x = -gyr->x;
ceNowGyr.y = gyr->y;
ceNowGyr.z = gyr->z;
```

ceNowAngles.altitude = ceEnvirmont->altitude;： 对数据进行转存的原因为：acc 与 gyr 是针对 MPU6050 模块驱动中结构体，而 ceNowAcc 和 ceNowGyr 为针对 CeFilter 模块对象驱动的结构体，两者本质上不同，需要转换。而 gyr->x 取负值的原因与 MPU6050 安装方向有关。

4) 数据融合获取当前姿态角

ceFilterOp.filter(&ceNowAcc,&ceNowGyr,&ceNowAngles,dtS);

即将当前加速度、角速度、执行周期 dt，进行滤波、姿态解算，并解算的结果，更新到 ceNowAngles 中，有关滤波及解算的详细过程，可参考文章《小四轴无人机滤波及姿态解算详解》，这里不再重复说明。

5) 根据期望姿态角，及当前姿态角，进行串级 PID 运算。

ceDrivePower = cePIDOp.calculate(&ceNowAcc, &ceNowGyr, &ceNowAngles, &ceHopeAngles,dtS);

前面说过，ceHopeAngles 与 ceNowAngles 的关系，通过 cePIDOp.calculate()，将对这两个结构体进行计算，并得出四个电机的驱动强度信息。而需要额外的 ceNowAcc 与 ceNowGyr 的原因是 PID 计算过程中角速度环需要的参数。

ceDrivePower 是结构体：

```
/**
 * @brief   结构体，四个电机的驱动力（油门），0~10000，可按需求转化为电机的转速
 */
 typedef struct
 {
      int16 driverPower0; /*!< M0 电机驱动模块的驱动输出强度，0~1000*/
      int16 driverPower1; /*!< M1 电机驱动模块的驱动输出强度，0~1000*/
      int16 driverPower2; /*!< M2 电机驱动模块的驱动输出强度，0~1000*/
      int16 driverPower3; /*!< M3 电机驱动模块的驱动输出强度，0~1000*/
 }CeDrivePower;
```

有关串级 PID 的详细运行过程，这里不再详细描述，可参考文章《小四轴无人机串级 PID 功能详解》

6) 根据 PID 运算的结果，更新四个电机的转速。

```
ceMDOp.setDriverPower(&ceMD0,(uavStatus != UAV_STATUS_READY)? (ceDrivePower->driverPower0):0);
ceMDOp.setDriverPower(&ceMD1,(uavStatus != UAV_STATUS_READY)? (ceDrivePower->driverPower1):0);
ceMDOp.setDriverPower(&ceMD2,(uavStatus != UAV_STATUS_READY)? (ceDrivePower->driverPower2):0);
ceMDOp.setDriverPower(&ceMD3,(uavStatus != UAV_STATUS_READY)? (ceDrivePower->driverPower3):0);
```

ceMDOp.setDriverPower()作用是将 0~1000 的驱动强度，转化为 0~100%的占空比。

第5章 遥控器软件设计

与无人机源代码框架基本相同，且所有硬件模块使用的驱动相同，唯一不同处在于

CePackage 与 CeTMU 两个功能。这里不再详细说明，可直接参考源代码。

第6章 手机 APP 控制软件

暂略。

附录：

1) CREELINKS 平台相关资料：http://www.creelinks.com/stdlib
2) 小四轴无源无人机原理图及源代码资料：http://www.creelinks.com/uav