**气动量仪项目总结**

1. CubeMX的使用

1.个人理解CubeMX的主要作用是更方便的建立工程，便捷的配置各个外设参数，需要注意的是，CubeMX使用的是最新的HAL库，某些函数形式与标准库有些许不同，但最终原理都是一样的。

2.编写CubeMX生成的工程时一定要将代码写入规定的区域内，否则在CubeMX重新生成工程文时没有写入指定区域的代码会被擦除。

3.当有新建的.C .H文件添加到工程时，切记在工程打开的状态点击软件的生成按钮，在显示重新加载工程文件时选择否，原先添加的工程文件会被保留下来，不用重新添加，同样，如果添加了一些外设，那么新添加的外设.c.h也不会主动添加进去。

1. 外设的使用及相关原理

## 1.电压采集之ADC

1.1 ADC采集名词介绍

分辨率：ADC的分辨率，是一个多少位数的寄存器，以STM32F103为例，其精度为12位，意思就是说他的范围时从0000000000~111111111111,分辨率越高，位数越多，能采集到的电压最小刻度就越小。

1.2 ADC功能设置

单次转换：ADC转换完一次后就停止转换

连续转换：ADC转换一结束就马上启动另一次转换

模拟看门狗：ADC转换的模拟电压值不在区间内会引发中断

扫描模式：扫描一组模拟通道，可设置循环扫描，必须采用DMA传输

1.3 本次CubeMX-ADC使用的代码流程

1. CubeMX软件配置
2. ADC校准
3. 使能DMA
4. 读取ADC的采集数据

1.4 ADC采集不稳定的一些解决方案

1. 减少ADC的采集频率
2. 基准电压采用外部电压
3. 在进入while循环之前一定要进行ADC校准

HAL库ADC的参考视频<https://www.bilibili.com/video/BV1mT4y1E7fS?from=search&seid=10093554805536417888>

## 2.屏幕显示之迪文屏

2.1 迪文屏的两种开发方式

（组态开发视频可以在B站进行搜索）

1. AT开发，通过发送AT指令对屏幕进行操作
2. 组态开发，迪文屏本身带有一个智能芯片，可通过官方的DGUS软件来进行屏幕设计，方便。
3. AT,NO;DGUS,YES!

2.2 本次所用到的迪文屏控件

1. 同步数据返回，按下可向外发送数据，
2. 数据变量显示，显示某个地址的数据
3. 图标变量，显示动态图表，本次测量界面的动态图设置以及一些动态提示都是由此完成
4. 变量数据录入，本次校准时的校准值设置用到此控件

2.3 迪文屏与单片机的交互实现

本项目中迪文屏与单片机的交互主要有以下两点共同实现

1. 同步数据返回，当触摸到具有同步数据返回的范围时迪文屏会通过串口向外发送一串带有地址的16进制数，单片机则根据接收到的地址来判断是触摸的哪一部分并进行接下来的操作
2. 迪文屏的读写指令，单片机可向迪文屏发送一串16进制指令，对迪文屏某地址的数据进行读或写，从而实现对迪文屏可控界面的控制操作

例如数据变量显示，单片机发送写指令从而改变迪文屏上显示的数值；

例如图标变量，单片机发送写指令从而改变某位置的图标，比如保存数据时的动态提示，插入优盘时的动态提示，不同数值时的条形图的长短显示；

例如读取输入的校准公差值，单片机发送读指令读取所输入的上限下限值并保存至单片机中。

2.4 在使用迪文屏中所遇到的一些问题

1. 用TTL转USB进行屏幕调试时一定一定要共地；
2. AT开发与组态开发是有区别的，有些控件只能在组态下用，开发时认真阅读操作手册，都有详细解释；
3. 迪文屏出场时会带有指令应答，每向迪文屏发送一条指令，迪文屏在接收后会返回一条指令，可以刷无应答内核或在程序中设置检测应答选项，防止通讯中断；
4. 迪文屏的触控设置是在.CFG文件中设置的，此文件可用UE软件打开并进行设置，设置时可参考操作手册进行相应的更改，实在不行问客服或者论坛。

注意，DUGS软件中包含一个CFG文件生成的小程序，一定不要在上面设置分辨率，会黑屏，论坛上的一些解决办法是手动更改CFG重新设置，更新时屏幕虽然是黑屏，但其实是在更新，等一段时间后再上电就好了，但是实际操作还是不能解决，最后送的售后寄修。

1. 电脑死机后重启电脑发现DGUS里的界面都没了，只要源文件还在，抢救措施如下：
   1. 新建一个文件夹，新建工程，添加前面DWIN\_SET文件夹里面的图片
   2. 把ico、等字库文件、config文件，复制到新的文件夹工程DWIN\_SET里面
   3. 再点击导入选择以前的DWIN\_SET
   4. 就OK了

6) 与迪文屏进行通信只能用16进制的数进行，因此，对于写入多字节的数据需要先将其转化为对应的16进制数后在进行发送

## 3.HAL库串口发送及中断

HAL库串口发送及中断接收原理与标准库一致，因为用CubeMX进行配置，所以编写代码时省去了很多步骤，会调用一些函数就可以，常用函数可参照如下链接

<https://blog.csdn.net/yigedaluobo/article/details/97795045?utm_medium=distribute.pc_aggpage_search_result.none-task-blog-2~all~sobaiduend~default-4-97795045.nonecase&utm_term=hal%E5%BA%93uart&spm=1000.2123.3001.4430>

注意，与标准库不同的时，HAL库串口中断会进入中断回调函数，可在此函数中编写中断操作。

## 4.数据导出之USB文件系统

本项目实现了U盘插入后数据导出到优盘的的一个TXT文件中，实现此功能需要两个部分协同完成，一个是FATFS文件系统，一个是USB-HOST。

首先来说文件系统，FATFS文件系统在标准库中需要手动移植，但在CubeMX中可一键配置，需要设置的提前设置的只有FATFS（用户逻辑驱动的文件系统对象）和FIL（文件对象）两个结构体以及 用户逻辑驱动路径，然后使用FATFS系统提供的一些函数来实现对文件的操作，需要注意的是，向文件里写数据有f\_write与f\_printf两种函数，f\_write是以二进制的形式写入文件，f\_printf是将格式化的数据写入文件，类似printf函数的用法，总之，会用fatfs系统提供的函数就可以。

其次是优盘的设置，STM32优盘模式有两种，一种是作为从机使用，即USB-device，单片机作为一个存储优盘，可以通过电脑往里面存文件；另一种是作为主机使用，即USB-HOST，单片机作为主机，可以向接入的优盘读写数据，进行文件操作，本次使用的是USB-HOST模式，将单片机中的数据以TXT文件的形式保存到优盘中。设置USB-HOST的流程可参考如下文档：<https://blog.csdn.net/love33521/article/details/8535050>。

最后，FATFS与USB-HOST的结合范例可参照如下网页：<https://www.stm32cube.com/question/514>，本项目的USB导出的大体框架及相关设置就是以此为蓝本。

## 5.flash存储及读取

Flash的读取及存储流程较为简单，写入操作为解锁flash，擦除flash，写入flash，上锁flash；读出操作为解锁flash，读出flash，上锁flash。需要注意的是flash 的地址以字节为一个单位，一个字节为8比特，比如从0X8000开始存储一个4字节的数据，第二个数据则从0X8004开始存储，常说的32位芯片与64位芯片是指芯片在读写数据的时候可以一次读写32bit或64bit的数据，即一个字位32bit或64bit。

1. 程序编写问题

## 程序整体流程

本程序的流程主要包含以下几个部分

初始化部分

ADC采集部分

屏幕与单片机交互部分

优盘导出部分

## ADC采集部分

* 1. 原始数据采集，将DMA中的数据读取出来并转化为实际的电压值
  2. 数据滤波，由于各方面的因素采集的电压值总是会出现小幅度的跳动，进而会使得屏幕显示的数值一直在不停的跳变，因此对采集的电压进行了平均滤波与阈值滤波处理，使最终采集到的数据保持在一个平稳的状态
  3. 显示数值的获取，将采集到的电压值带入到计算公式中

需要显示的值 =（采集到的电压值 – 校准下限时对应的电压值）\*比例系数K + 校准下限

K =（校准上限-下限）/（上限对应的电压值-下限对应的电压值）

## 屏幕与单片机的交互部分

* 1. 触摸检测，在触摸屏幕后，屏幕会返回一串16进制数，单片机根据这些返回值以及接下来的返回值判断进入了哪个界面的哪种程序，并使能其对应的标志位，开启对应操作
  2. 首页部分，进入首页部分后其他页面的开关标志位关闭，选择对应的通道数，并将数字发送至屏幕上（注意，屏幕只接受符合协议的16进制数组，因此需要将发送的内容转换为相应的16进制数，一个16进制代表一个字节，多字节的数可通过位运算转换为16进制来进行发送）
  3. 测量界面包括实时数据显示，跟随数据变化的条形图显示，圆度显示及开关，保存数据

1. 实时数据显示，向迪文屏变量显示控件写数据的方法有多种，可一个个写入，也可以一次写入多个，但需要多个写入变量的地址必须连续，因此在设计迪文屏的界面时需要将所需要显示的变量控件地址部署合理且连续，在写入时可将需要写入的数据统成为一个数组一次性发出从而节省时间，简化代码，并且迪文屏的接收数据间隔为20ms，此举也能防止数据发生丢失；
2. 变化条形图，其本质是不同的数字代表不同的图案，先对显示的数据进行范围判断，再发送相对应的数据，从而实现条形腿跟随显示数值发生对应的变化，同时控制三色顶对应的IO口的高低电平，进而实现对灯的跟随控制
3. 圆度显示及开关，因为圆度值代表的是极差值，只能设置一个开关来控制极差的测量及显示
4. 存储数据，点击存储数据后首先将对应flash地址的原始数据读出，将原始数据更新为新数据，擦除原flash，将新数据写入flash中，需要注意的是，flash只能存储16进制数，需要将多字节的数据通过位操纵转换为对应的16进制数，同理，在从flash中取出数据后，需要将读出来的数据转化为对应的多字节数据并进行更新。
5. 校准及公差界面，这两个界面包含通道选择与校准输入，上下限输入时的小键盘是通过迪文屏的数据录入设置，点击校准后单片机对写入的数据进行读取并存储到单片机的flash中
6. 数据导出界面，此界面展示存储到flash里面的历史数据，其本质和数据存储差不多，都是将数据从flash中读出来并发送到屏幕中，并且进入此界面后才可以插入优盘进行数据导出