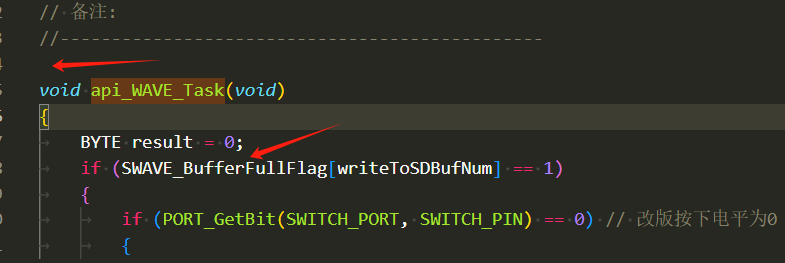
# 2025.1.16

1、编程规范（已修改）

（1）上面的去掉回车

（2）下面的加上回车



（3）文件名最好也按照单词的首字母大写来定义



（4）同类型的变量定义可以放在一起，初始化可以放在后面



（5）加括号

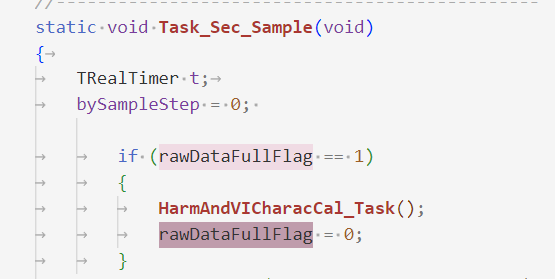


（6）没用到的都删掉

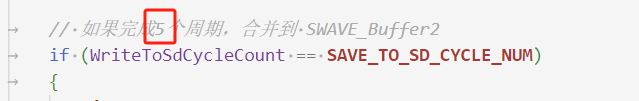


（7）测试用代码加注释

（8）对齐



（9）改宏定义后，同步更新注释



（10）定义放最前面



2、CRC异常，数据应该清零（改为使用前一个半波数据）

3、按键

（1）应该去抖，不能只判断一次；可以参考其他产品的按键检测（按键机制不同，他们的按键按下会回弹，工装的不会，自己加了去抖判断）

（2）按键按下，创建文件，不用关心是否有存储的标志（已修改）

（3）弹起按键，关闭文件，不用关心是否有存储的标志（已修改）

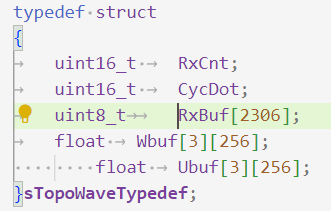
（4）按键按下和弹起单独一个函数，根据标志存文件单独一个函数，需要检查当前文件是否已打开（已修改）

4、（1）文件名个数太大了（改为10000）

  
（2）创建文件后，闪灯可以放在定时中断中（已修改）



5、为什么定义了2306字节的接收缓冲？（改为1154）



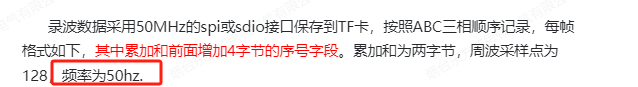
实际DMA只收了1000多个字节



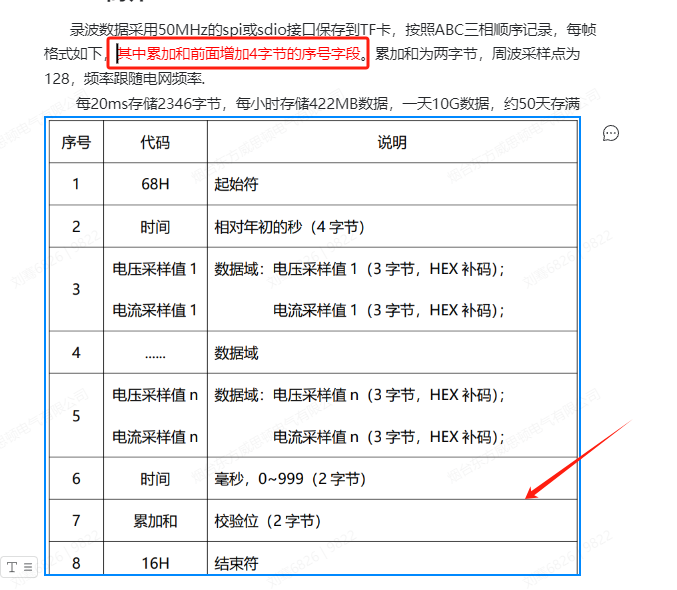
6 **CalSecondsSinceYear**

用**api\_CalcInTimeRelativeSec替换**

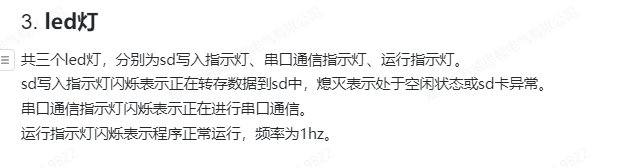
7、（1）确认SPI数据的输出是与电网同步的（8306寄存器），下面的写法不对（原来配置不同步，已修改）



（2）直接修改原来的表格（已修改）



（3）sd都改为tf（已修改）

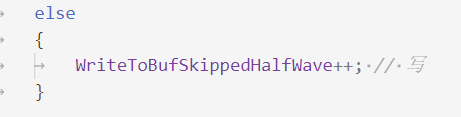


（4）测试TF卡写满是否会覆盖之前的数据（写满后返回FR\_OK,但是写入字节数为0）

8、fullWaveRecFlag用sTopoWave.RxCnt替代（已修改）

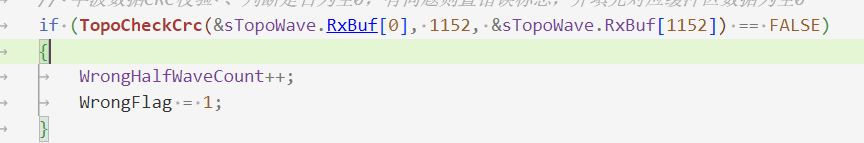


异常时，可以清一下sTopoWave.RxCnt（最好不要破坏半波计数）



9、

校验错误，需要将对应的缓冲前面的半波进行取反操作（已修改）



WrongFlag

就可以删掉了（该标志用于区分正常的拷贝操作，防止错误时下面再拷贝一次接收到的波形数据）

10、满一个完整周波，一次拷贝一个周波的数据用于谐波计算；不需要else分支分两次拷贝（只能每半波保存需要的数据，主循环buf的数据格式不同）



11、rawDataHalfWaveCnt可以用sTopoWave.RxCnt替代吗？

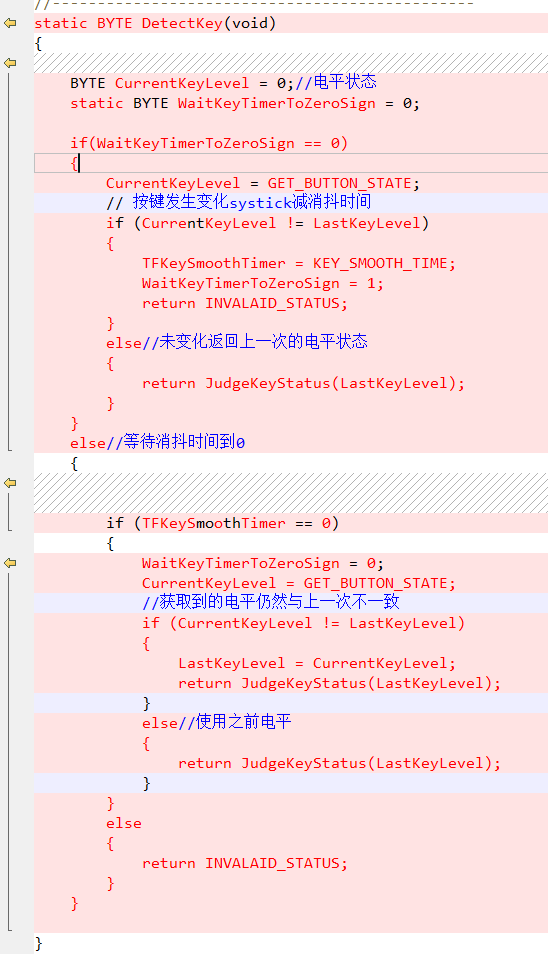


# 2025.1.21

1、掉电后关闭文件，需要结合线损排查仪的掉电波形测试（问二勇）（法拉电容撑很久）

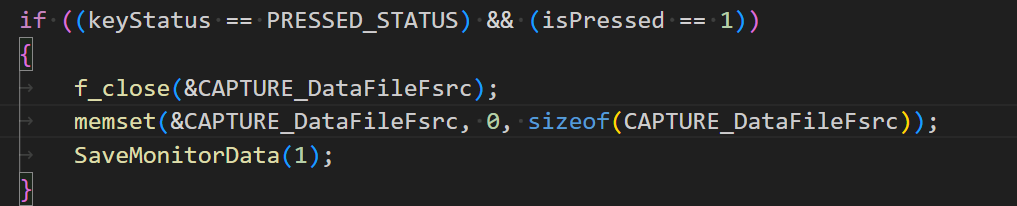
api\_PowerDownWaveRecord

2、这相当于只检测了2次



3、掉电时是不是应该也这么处理下?（已修改）

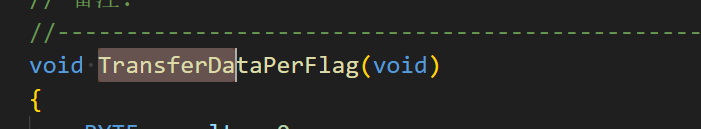




4、编程规范

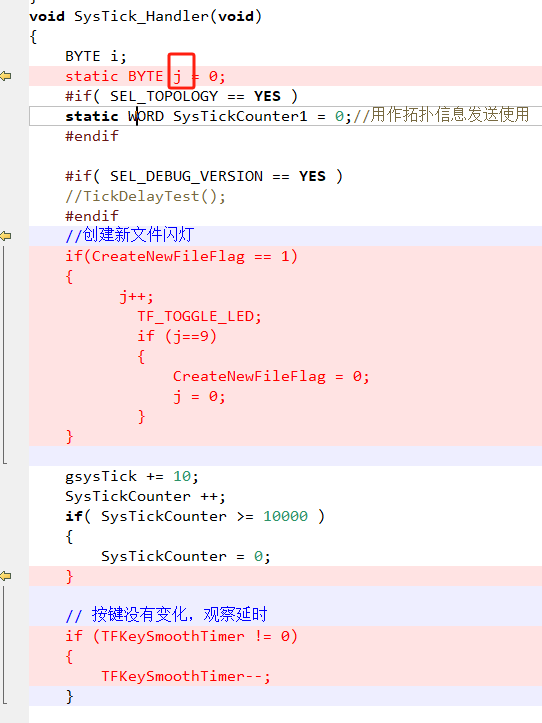
（1）（已修改）

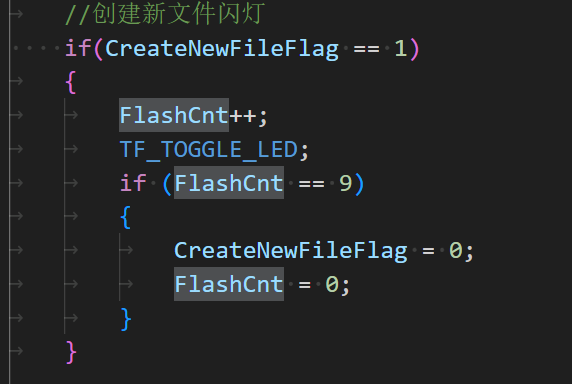




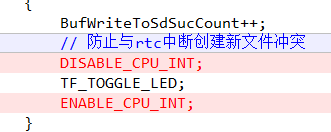
（2）一般全局变量不定义为j，容易与局部变量混淆（已修改）

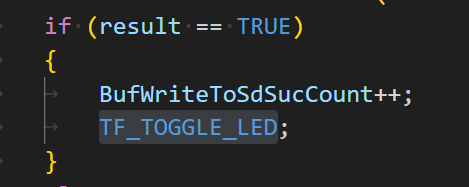
（3）tab对齐（已修改）



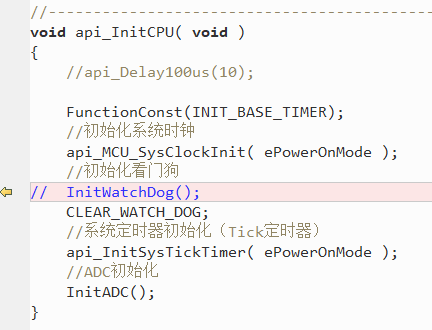


5、这个亮灯不用关中断（已修改）



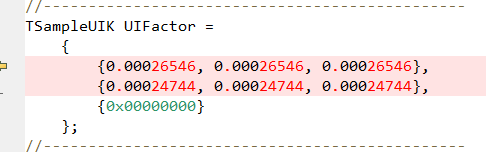


6、这个要关闭看门狗吗？（已开）

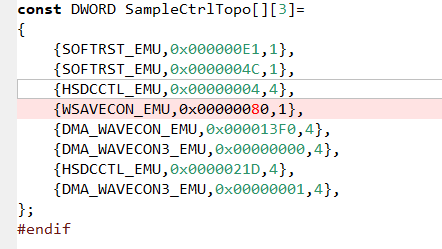


7、这个是临时的？（临时，校准后存到eeprom）

新的硬件，这个系数应该是需要校准的

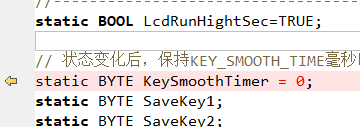


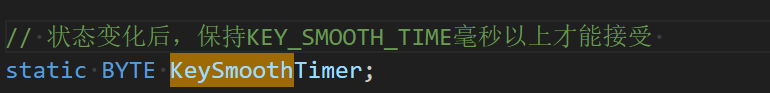
8、增加注释（已修改）





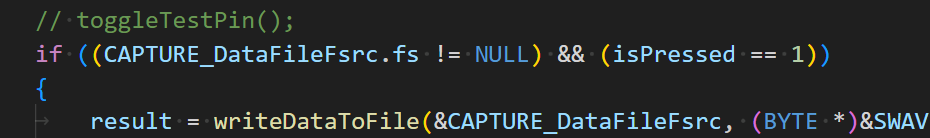
9、没用到的不要改（已修改）





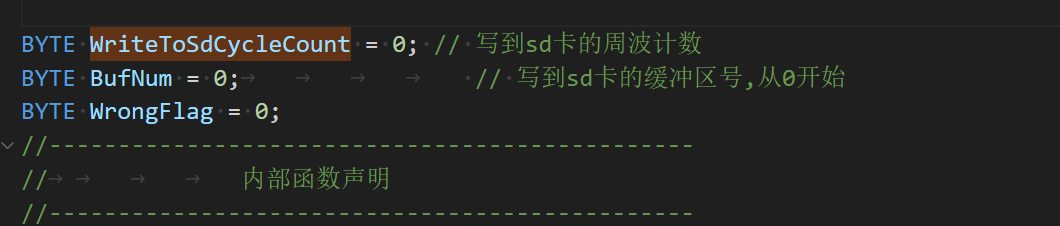
10、写文件，最好再结合isPressed的合法性判断（已增加）





11、这些全局变量都需要吗？(已删除不需要变量)





12、

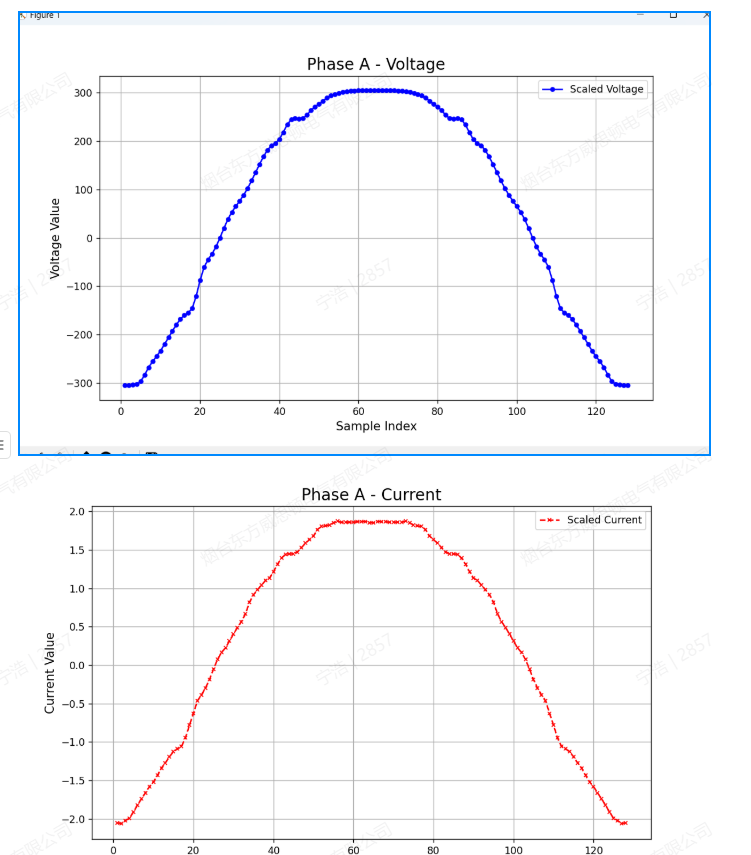
NILM\_GDF415\_QJZ\_RT-Thread\_20250110\_1812\_Origin\_768K\_rtt\_1phase\_25Submit.rar

以后代码的压缩包，建议使用这种方式，产品名称，硬件平台，日期等信息

13、造个异常的情况，看看用前面半波补的对不对 （已测试）

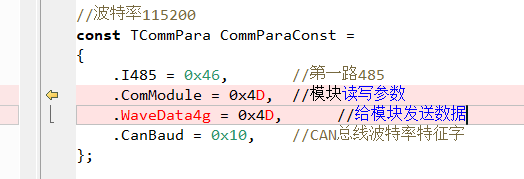




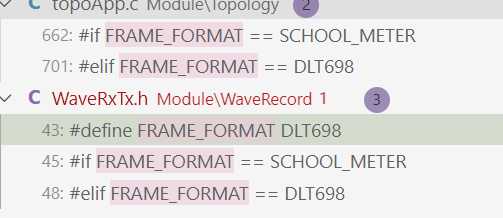


# 2025.4.30

1、对齐，加注释是多少波特率



2、加括号



3、对齐

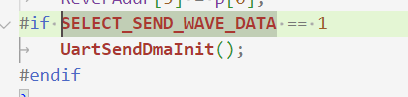


对齐，加注释





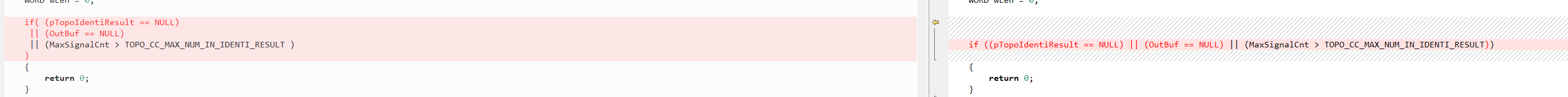
加括号，尽量用YES，NO替代0,1



3、用的时候，现转换，否则全局变量冲掉后，地址一直是错误的



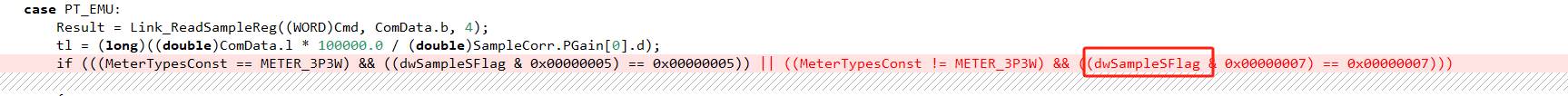
4、这个文件里之前都写的是对的，为什么都改为同一行显示了？



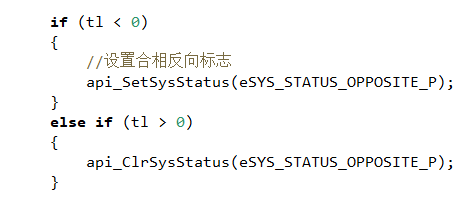
5、4路step需要独立



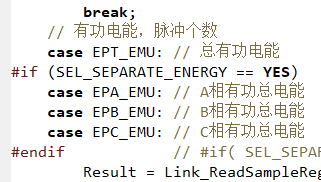
需要区分4路dwSampleSFlag



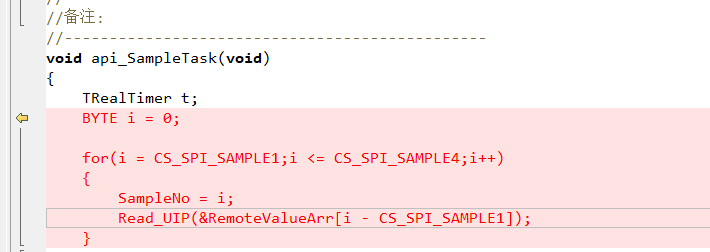
正负号等



电能可以不读取



建议使用4回路LTU的方式校准，获取瞬时量数据



6、这个没用



7、枚举定义加e



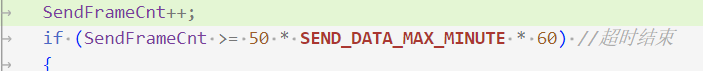
8、充电结束默认30分钟，除非不到30分钟，可以由充电结束检测来结束

BeginSendData

9、先加后减



10、加括号



# 2025.5.15

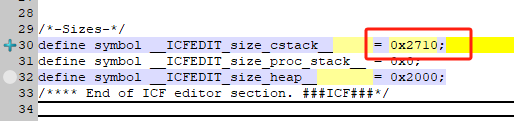
1、需要放在大循环里检查吗？



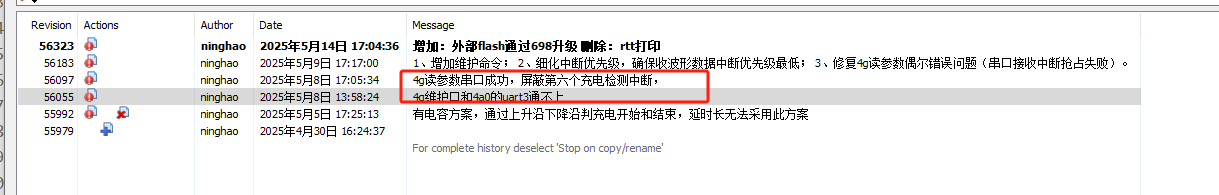
2、



3、为什么定义0x2710

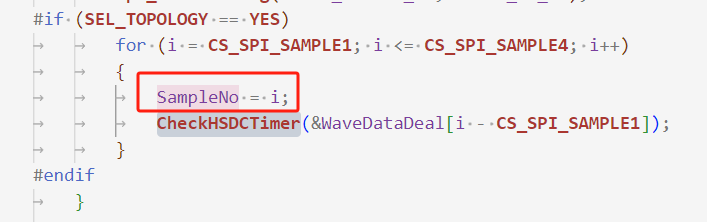


4、注释太少

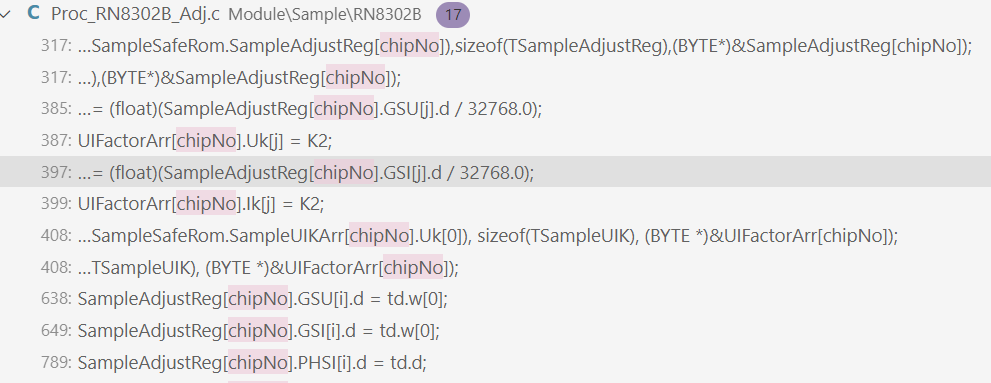


5、

（1）需要吗？



（2）用chipNo和的地方，都容易被冲导致值超限，；建议参考4回路LTU的方式



6、

gMaxRecWaveTime

1. 增加平时的校验机制
2. 初始化赋默认值

7、工程里每次编译都会变的文件不要上传到SVN



8、#if 加括号，其他类似



9、这个要多判断几次，万一有一次误判

IsChargeFinished

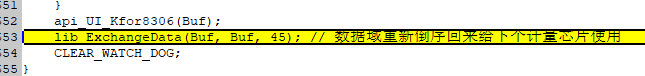
10、反转完直接就用了，不用定义全局变量



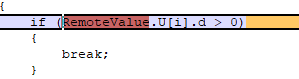
11、



12、这个操作放在规约层处理



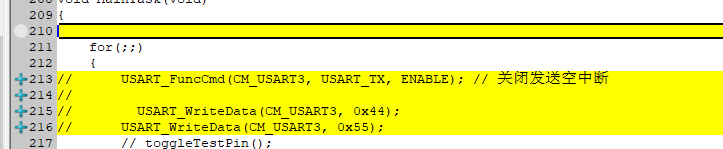
13、不用的变量都屏蔽，查看是否还有没改全的地方

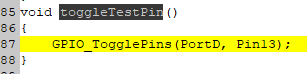


14、UIFactorArr和gMaxRecWaveTime一样，上电和每1小时检查ram校验，异常时从eep恢复，eep读取错误，则使用默认值



15、删掉测试代码



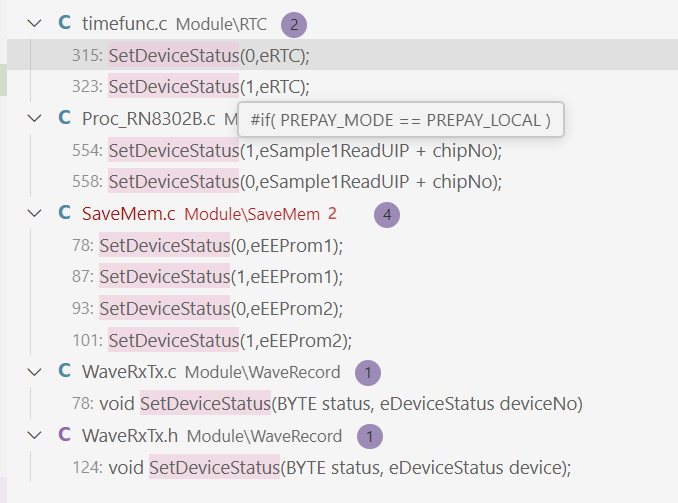


16、测试时打开，出厂正式版时关闭



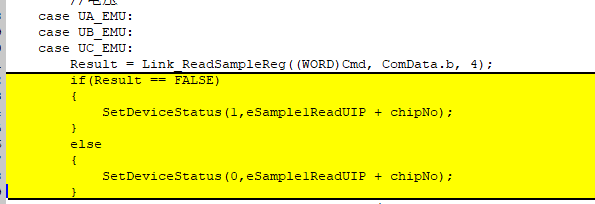
17、异常的每个bit的位置，扩展规约可以自定义；

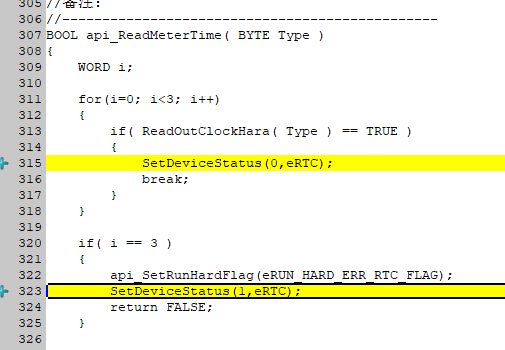
异常置位，清零的地方参考费控导轨表；



已有的异常不要自己改置位，清零的位置，比如：

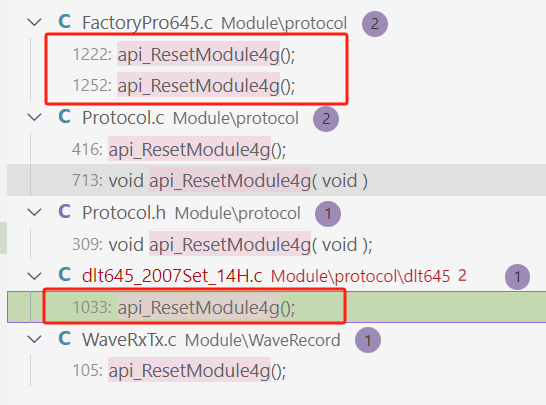






SetDeviceStatus函数名改下，这个只是Set

18、4G复位命令怎么有三个？



19、

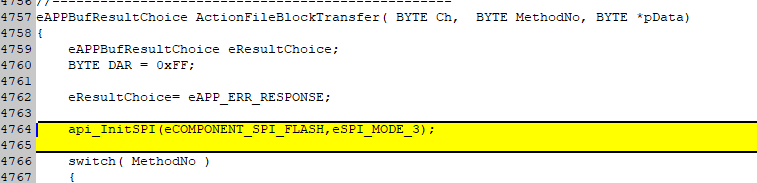
*CROverTime*

参考健亮程序，上电赋初值

20、无掉电检测，进不去



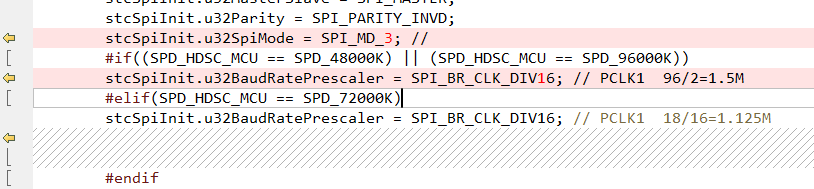
21、不应该出现在这里

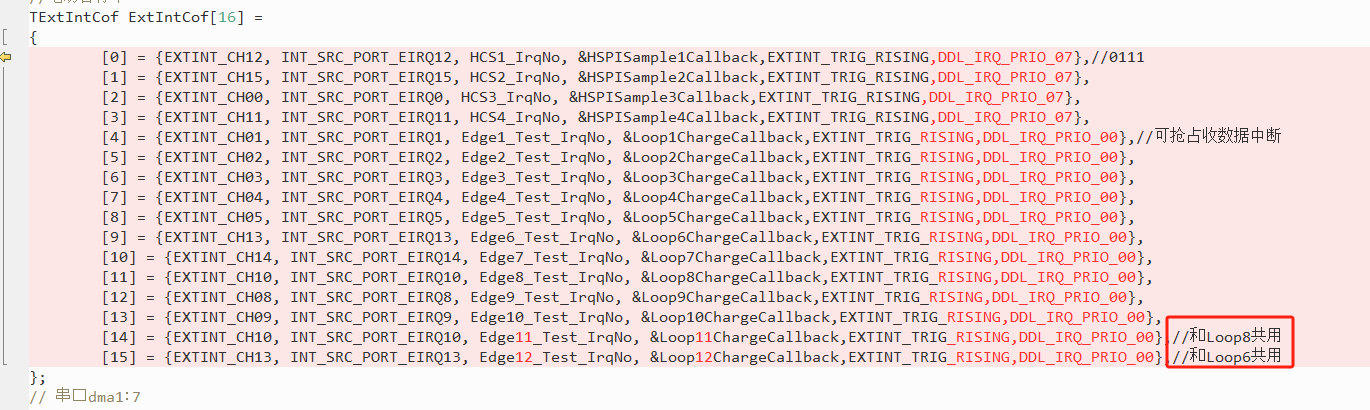


22、后面的都屏蔽掉

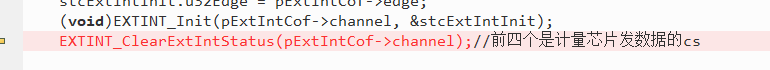


23、实际是多少的频率，改下注释





这个注释加在文件前面数组里吧ExtIntCof



没用删掉



24、

这个函数WriteFactoryExtPro对应的操作应该保存自由事件记录

25、

**POWER\_DOWN\_VOL\_VALUE**

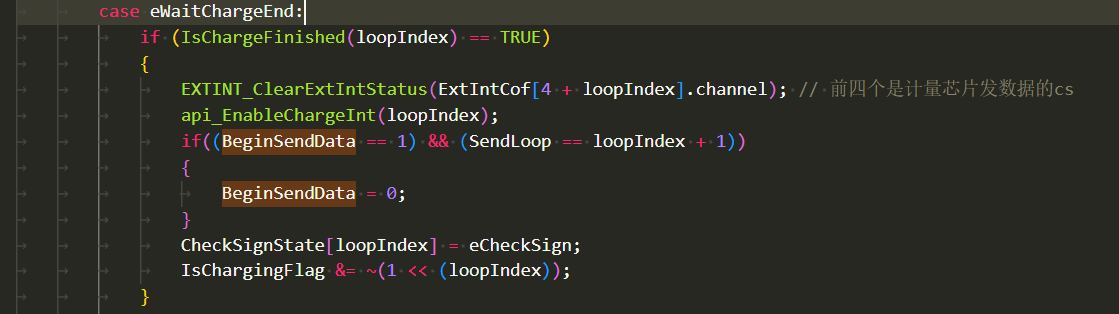
这个最好也做成参数

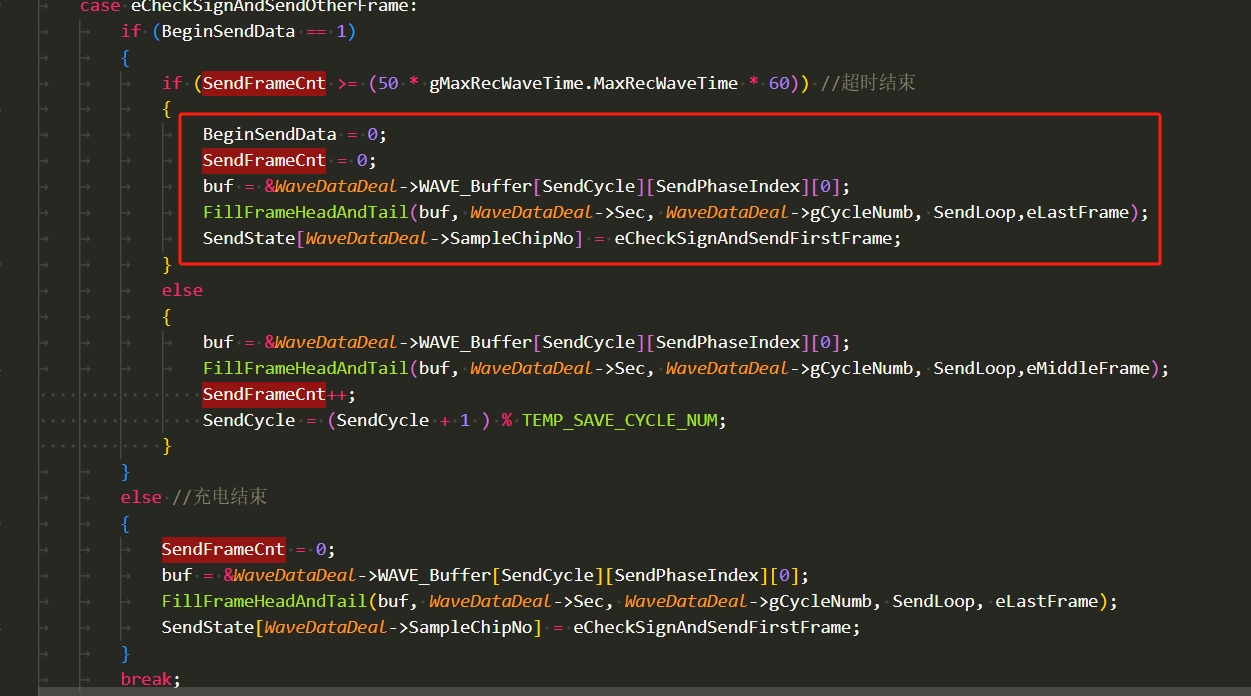
26、只在一个文件用的，不用extern

加上注释



27、检查到结束和超时结束的处理目前不一致





# 2025.5.16

1、宏定义



2、结束时将SendLoop置成非法值，比如0xff



3、超时结束时，需要将CheckSignState等状态归位

4、一回路检测到充电已经在发数据，另外一路检测到充电开始后，一直等待结束，是不是也要有超时处理1



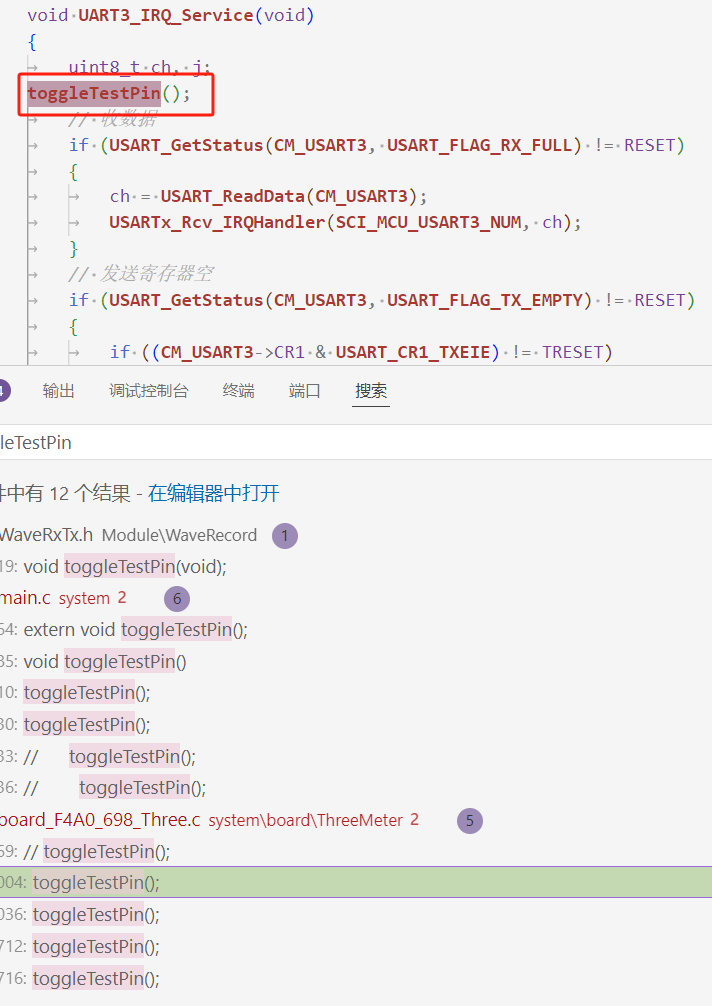
5、不需要定义为全局

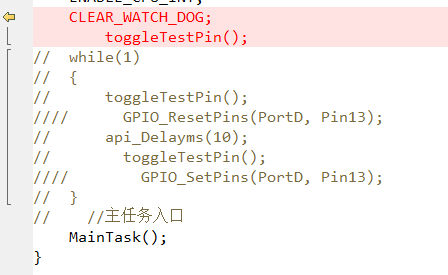
SendCycle

SendPhaseIndex

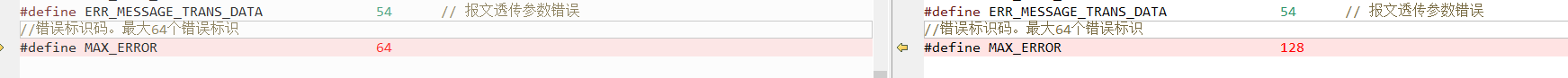
# 2025.5.21

1、测试引脚怎么还到处都是，且没对齐

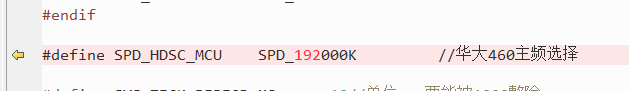




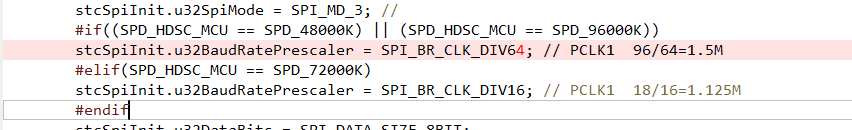
2、为什么改为128？ 注意注释



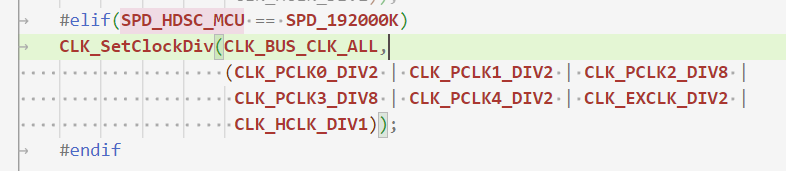
3、之前主频不够吗？改主频模拟测试速率部分都要重新测试



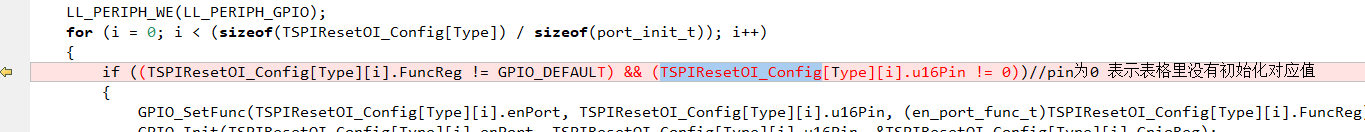
主频改为192M后，这里对吗？注意注释

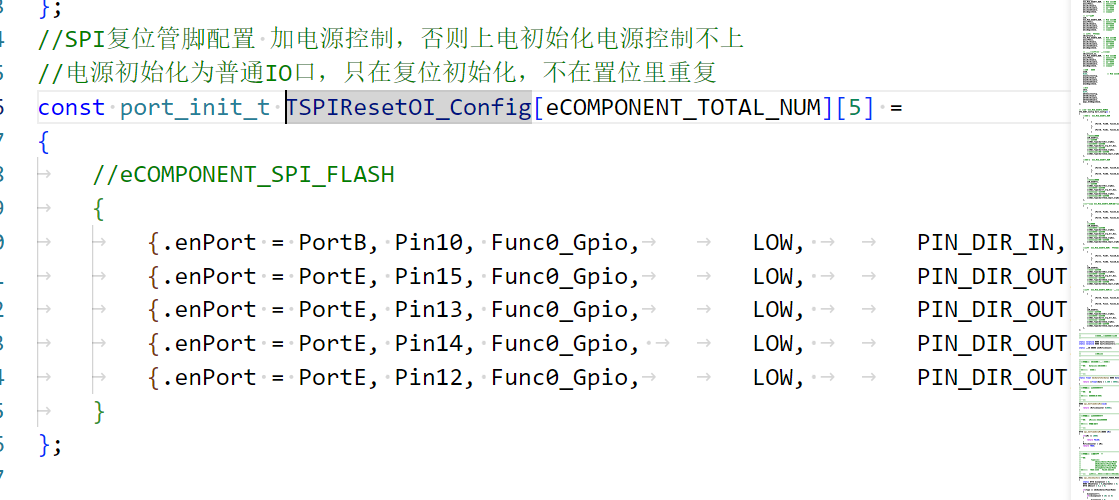


影响到系统时钟了吧？



4、这个是参考哪个改的？



原代码没有小于五个引脚的情况

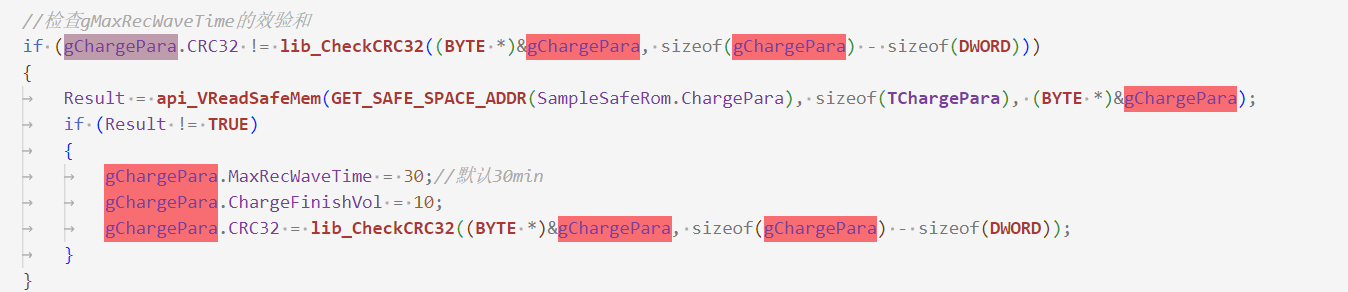
5、这部分写成函数，checkxxpara， **api\_PowerUpPara调用，para.c里的每小时调用检查**

FactoryPro645.c里的读写也写为函数，放在para.c里，FactoryPro645.c调用

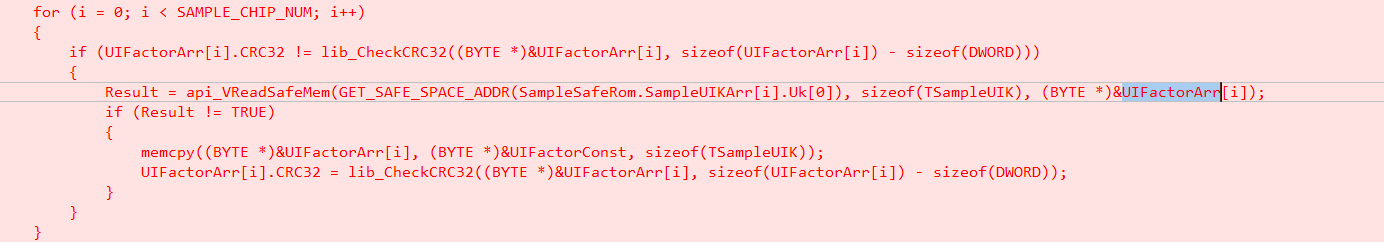
增加初始化函数，放在para.c，参考其他参数的初始化；

全局变量无需初始化

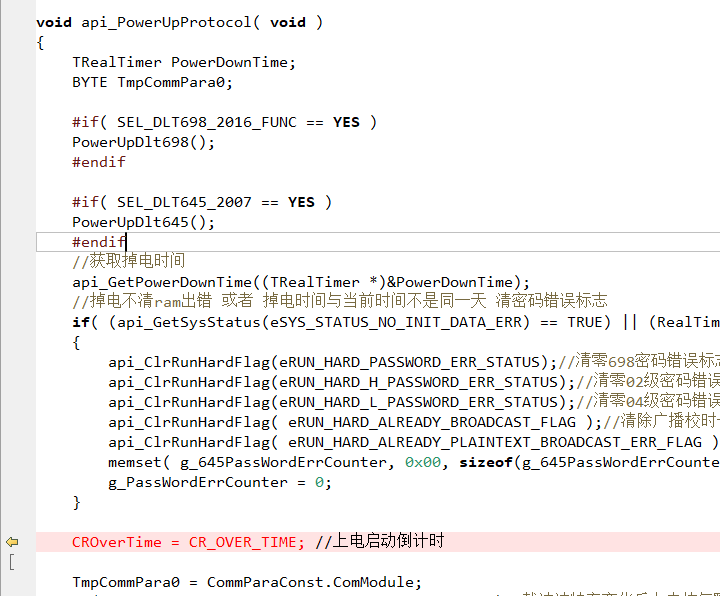




UIK系数类似

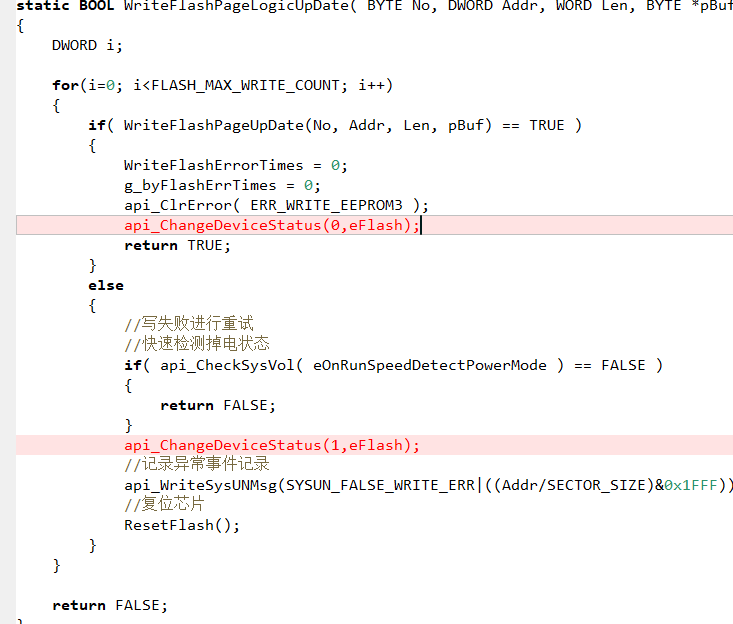


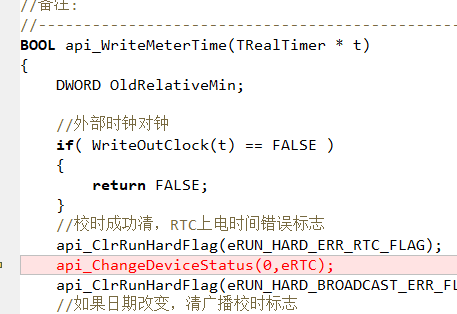
6、费控导轨表这里没有赋值



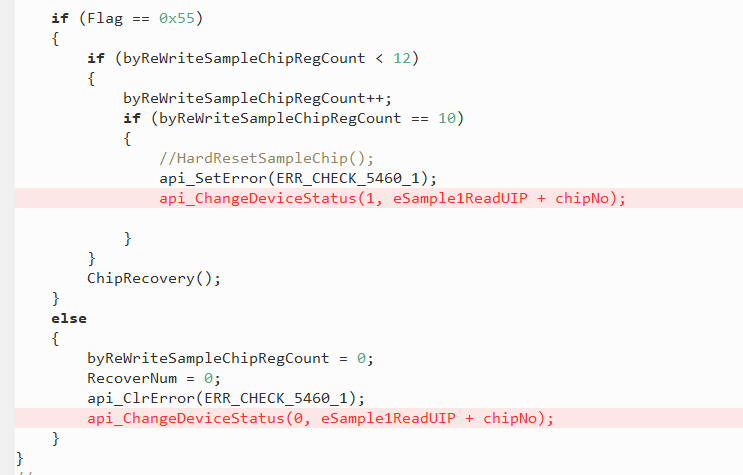
7、时间，eep，flash，计量芯片等异常，参考健亮645协议读取运行状态字

api\_ChangeDeviceStatus 函数可以删掉









8、删掉



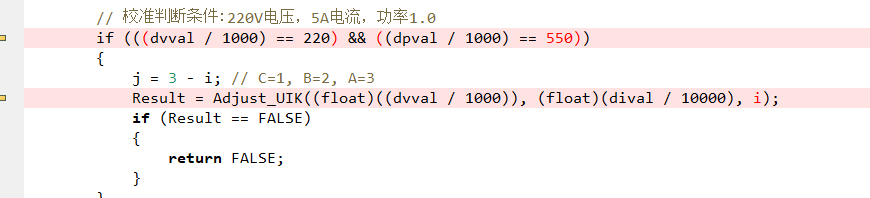
9、计量芯片没刷新，读3次和读1次效果一样



10、目前按照5/20A校准的？



11、这个是550w吗？和注释不对应



12、校园表校准UIK系数是通过扩展规约校准的，

api\_UI\_KforF415

13、不用的话，整个函数屏蔽掉

**api\_CalPaverage**

14、要区分回路



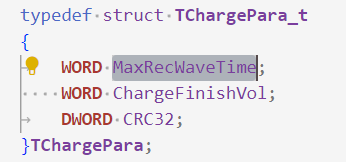


Proc\_RN8302B.c

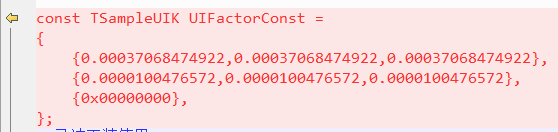
Proc\_RN8302B\_Adj.c

文件再排查一遍，看是否有未区分回路的地方

15、

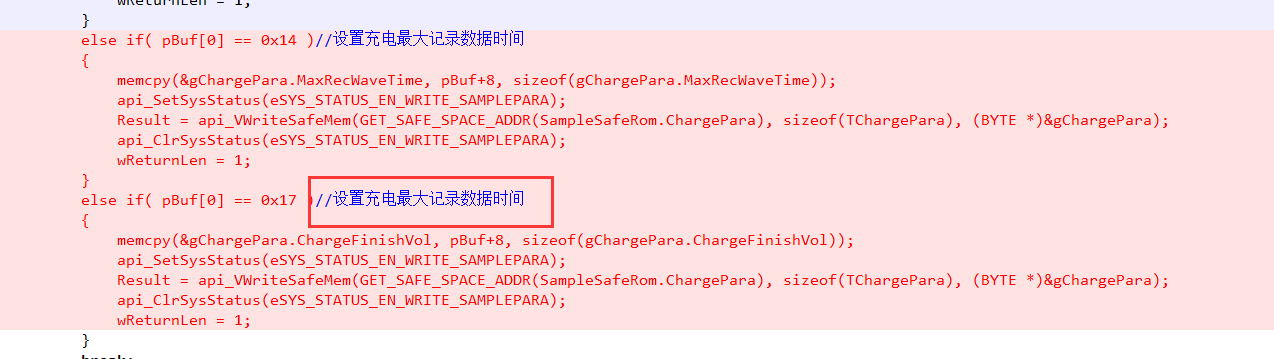
这两个也加下注释

16、

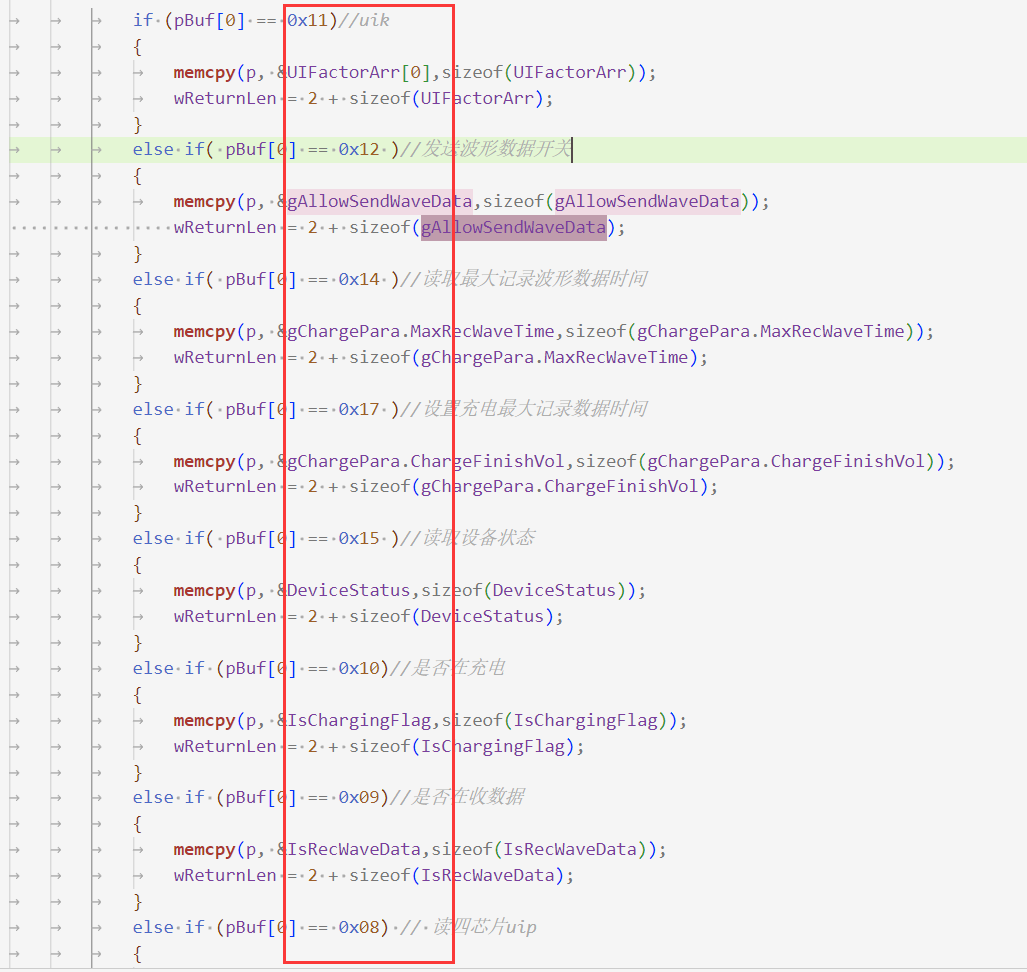
这个系数是取了几个校准后系数的平均值吗？

已修改

17、



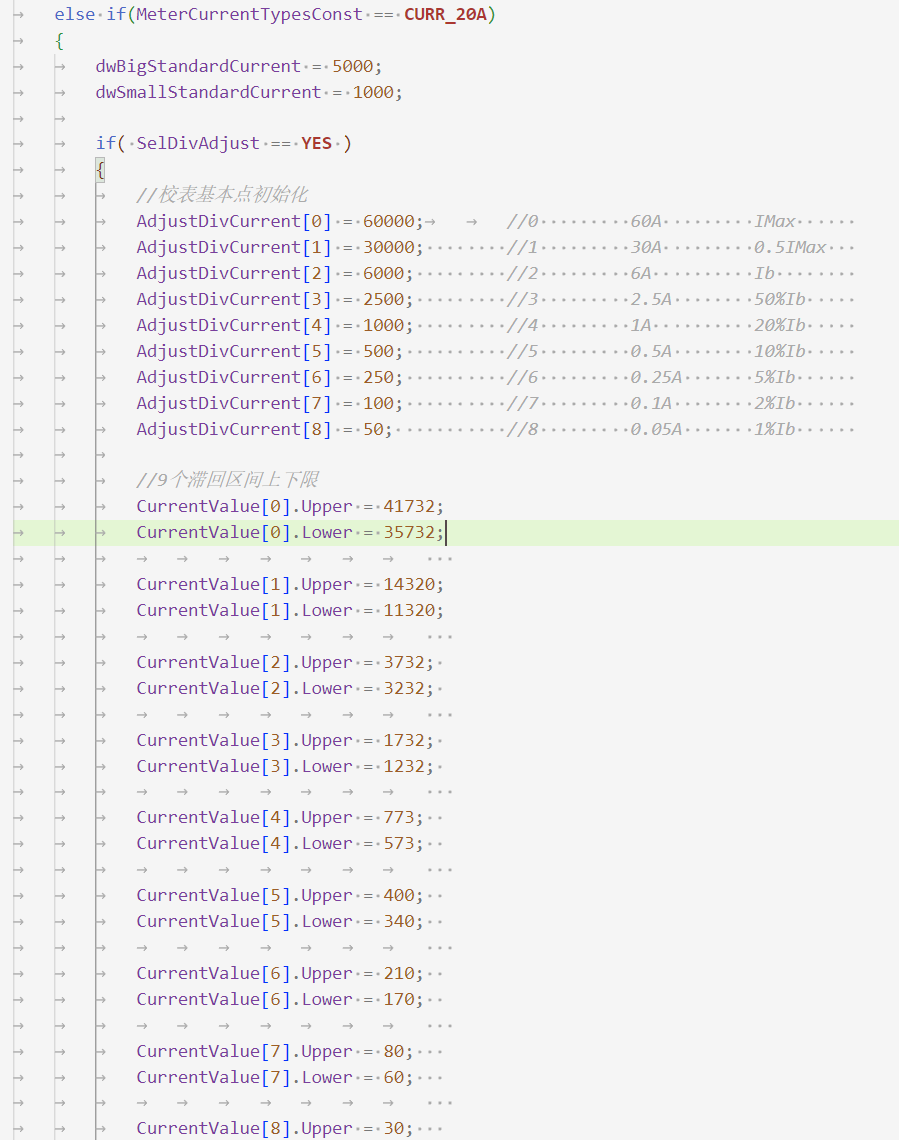


由小到大排列

18、

和高朋辉确认下

19、

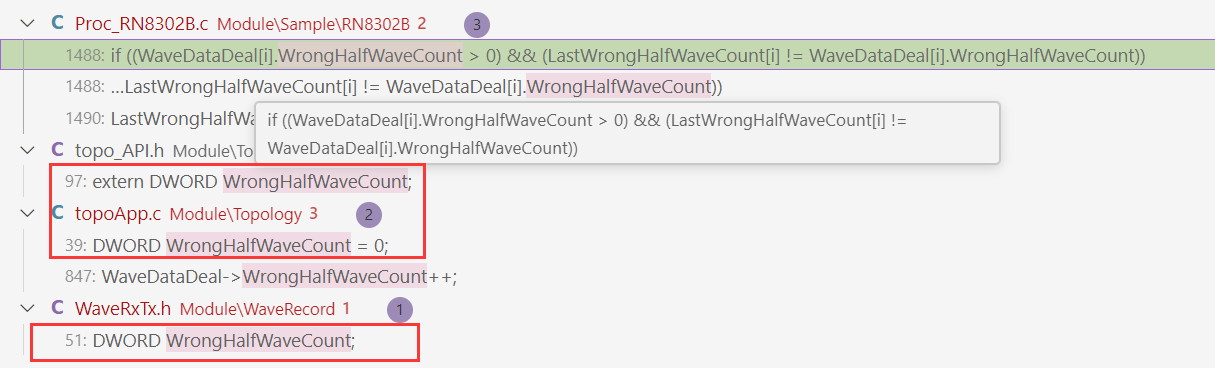


20、



参考LTU4回路

21、

删掉

# 2025.5.27

1、

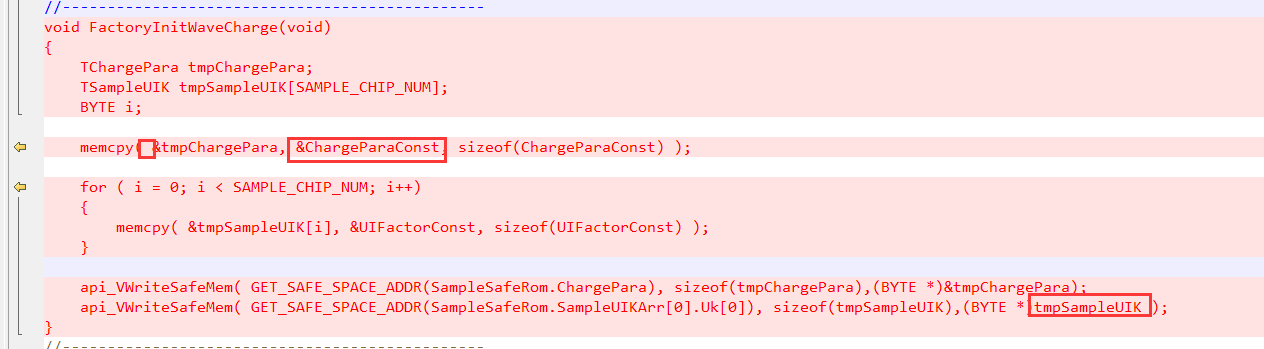
（1）校准系数，表号，通信地址等，初始化时不能恢复默认值

已修改

（2）memcpy( （BYTE \*）&tmpChargePara, （BYTE \*） &ChargeParaConst, sizeof(ChargeParaConst) );

已修改

（3）带数组的，一般习惯带下标(BYTE \*)&tmpSampleUIK[0]



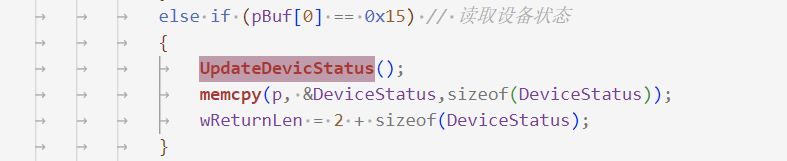
下面这个类似



已修改

2、

（1）这个不用单独update



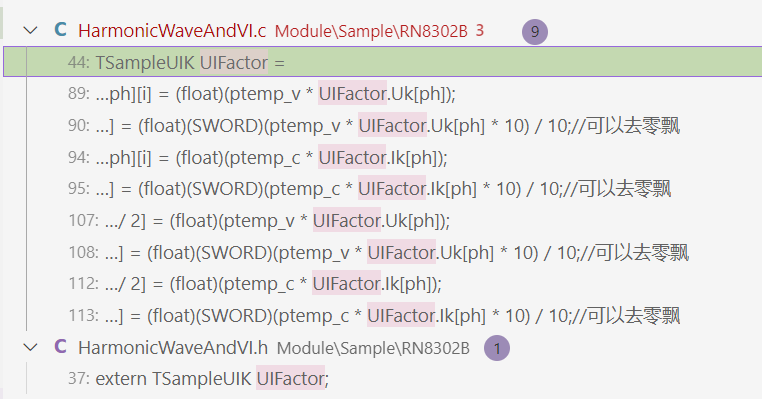
（2）可以不用这个变量，读取的地方，现调用**api\_CheckError，api\_GetRunHardFlag**

这些进行检查



已修改

3、这个没用吧



已删除谐波特征计算文件

4、还没去掉



正在模拟测试

5、上次bug分享说过，尽量少改动序号，否则上位机软件解析不对



6、高朋辉LTU用的192M主频，那就和他保持一致，再测试下时钟和延时

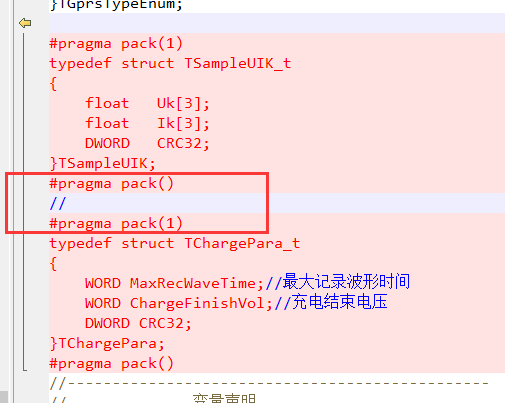
已修改

7、200使用宏定义



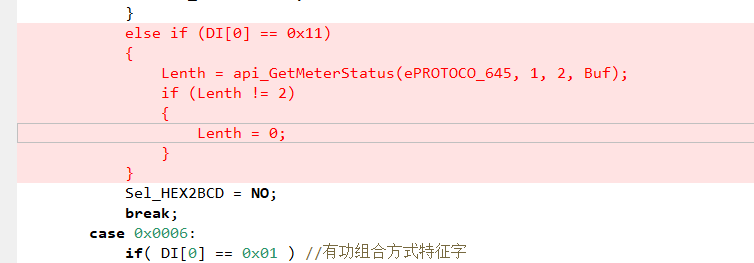
已修改

8、用一个就可以了



已修改

9、你不是有自己的扩展协议读取状态字吗？这个又合并回来了？



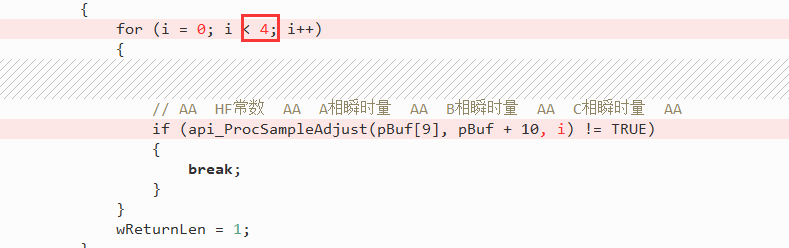
测试的时候试了下，读不出来，加上后也用不上，和扩展不冲突

10、异常时用const恢复，不用重新计算校验，有可能是读取eep异常，可以再读一次



已修改

11、宏定义



已修改

12、目前这个参数对我们没什么用

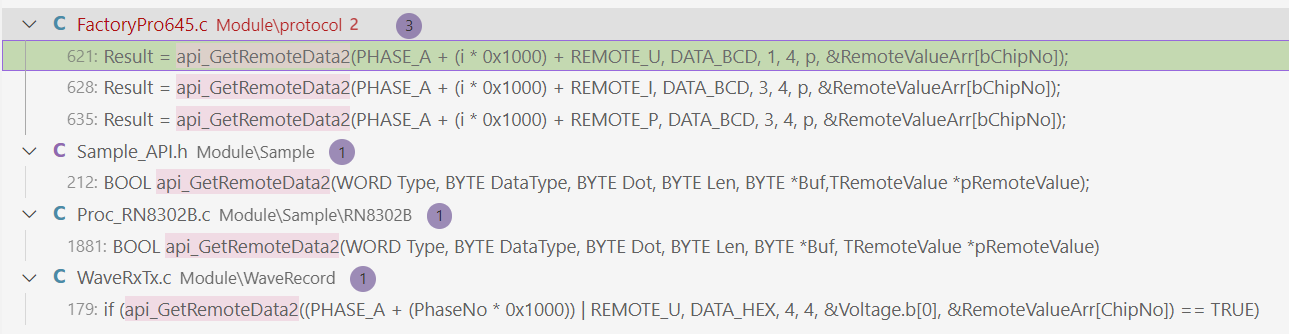


13、不用带参数，直接赋值全局变量RemoteValueArr

已修改

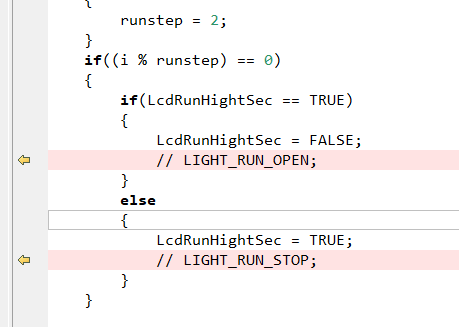


14、调用的地方不用更新全局变量，只有UIP更新（传入的是全局变量）

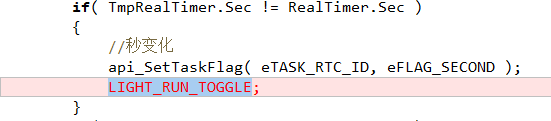


# 2025.6.3

1、



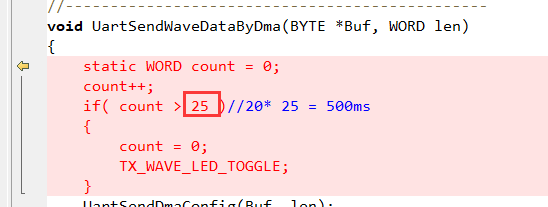
一般不在这里点灯，在上面点灯



已修改

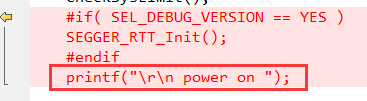
2、（1）宏定义

（2）定义与语句之间加回车分开



已修改

3、这里能打印出来吗？放在测试版里吧



已修改