# 模块文件夹的命名：

1. 目录采用全部大写的形式，里边的.c、.h文件全采用小写模式





2、sys.c文件中，有一些端口的地址的宏声明，如果有define要记得加上头文件

# 各个驱动模块的测试

1. 各个模块的单独测试文件都在对应h头文件中

使用模块时要调用初始化函数，延时1秒是有问题的，采用500ms进行叠加即可

AT24C02：ok

KEY:ok

Key0、1、2、3对应了PF 9、8、7、6

LED:ok

LED0、1、2、对应了PE 3、4 、PG9

BEEP:ok

beep对应了PG7

exit:ok

RTC：ok

IWDG:ok

TIM3:ok

初始化之后会马上进入中断服务函数---待解决

DHT11：ok

接ds18b20的中间接口

光敏：ok

所接的接口是PF3 的ADC3通道9 ，在主函数中要调用channel9不是channel5

pwm待使用，打算控制电机

ESP8266模块：先开wifi在上电，这里也要做处理

注意：ESP8266在连接的时候不对其返回数据判断，最后收尾时在尝试，现在无法解决数据接收的问题，直接采用直接连接的方式

在连接的时候需要给模块初始化的时间，初始化之后再继续通过串口发送指令！否则会出现错误，可能是没有初始化好，直接发送指令，导致有问题出现（第四天早上解决2020-1-26）

出现概率性连接不上：通过连接热点的时间

用户在控制台点击太快会造成数据无法进行传输，不能点击太快，需要检测心跳包的处理，结合看门狗进行处理，将系统重启

# 功能初始化函数命名

1. 采用的命名规则是：LED\_init()

# 进度

在esp8266接收启动时候的数据接收不全，用数组接收不行，以及用串口1发送也存在着问题，

网络差，无法开发

# Tlink的功能：

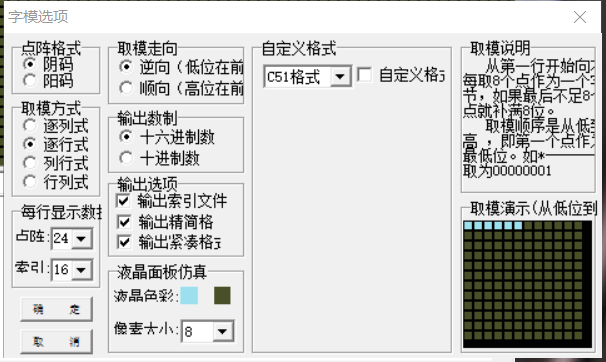
1. 点击之后立马更新状态，获得最新的数据
2. 是否开启智能模式
3. 手动控制窗帘的开与关，会自动关闭智能模式

注意：需要在60秒内把数据上传上去否则数据无法实时更新，可能需要手动去刷新上

# LCD模块：

效果：LCD要用合适的图片显示信息：温度湿度、气体、光照、窗帘的状态，以及光照强度设定的阈值，是否开启了智能模式，以及简易的使用说明，添加一些字体滑动的效果，以及要制作开机界面，以及推出界面

1. LCD的背光是PF10



字模软件的设置：字体大小采用24x24，占用了 高24/8 \* 宽24 = 72字节

1. FSMC(灵活的静态存储控制器)笔记

FSMC 总共管理 1GB 空间，拥有 4 个存储块（ Bank），STM32F4 的 FSMC 存储块 1（ Bank1）被分为 4 个区

CS： TFTLCD 片选信号。

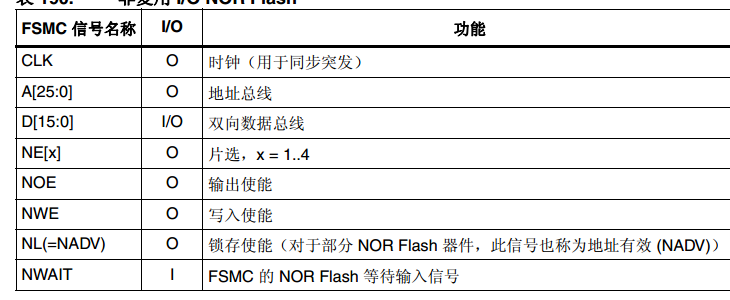
WR：向 TFTLCD 写入数据。

RD：从 TFTLCD 读取数据。

D[15： 0]： 16 位双向数据线。

RST：硬复位 TFTLCD。

RS：命令/数据标志（ 0，读写命令； 1，读写数据）。



FSMC\_A是地址线

FSMC\_D是数据线

FSMC 自动控制了 WR/RD/CS 等这些信号

# 七、优化

1、汉字能否用更直观的方式调用，不通过输入数字

2、LCD\_DisplayChinese\_one函数中需要优化---lcd.c（可能不能优化处理）

3、服务器下行的指令可以设计得稍微复杂，加上物联网的校验码

进度：在显示汉字字符串的函数还在修改

# 流程简述

1. 首先进行与tlink服务器的对接（大约14秒），先发送指令，同时开启10秒的定时器，10秒后开启5秒的看门狗，在这5秒内，需对接上服务器会马上喂狗操作，否则就将系统重新启动。

14秒---喂狗之后(5秒后)开启定时器4秒，在4秒的时候

看门狗检测连接的稳定性是行不通

定时器3、4最长的定时在6秒左右

1. 连接之后定时器与看门狗会被重新设置成检测心跳包，用户点击快有可能会导致连接中断

通过第一个心跳包延迟5秒来开启一个TIM2定时器10秒，刚刚用来检测是否有心跳包，没有的话先 发送+++退出透传模式，然后看门狗再去重启系统，不发+++，无法重新发送AT指令，继而无法重新连接服务器

TIM4负责喂狗，开启看门狗后无法关闭，芯片设计的事

TIM3负责esp8266段的成功启动，才能喂狗

通过打印来看程序的执行情况

Bug：

在接收15个心跳包之后，看门狗就复位了，检测的时间有问题

解决方式：通过每收到一个心跳包，开启TIM2的12秒定时器，在10秒后再次收到心跳包，将定时器的计数值清零(TIM2->CNT=0)，从而不会进入中断服务函数，如果没有心跳包，将会进入中断服务函数，不喂狗，让系统重启

2020-1-29打通板子到tlink服务器的通路