**移动端温度控制设计**

年级专业：16 级软件工程

方向：嵌入式软件与系统

组员：宋晓彤 16340192

苏依晴 16340196

熊秭鉴 16340259

叶梓豪 16340277

张金亮 16340288

完成日期：2018/12/7

**【设计思路】**

通过使用单片机上的热阻温度检测模块对温度进行检测，然后通过单片机数码管显示；内置变量模拟温度设定，可通过手机端/单片机按钮模块进行调整，温度达到设定目标或倒计时走完后单片机进行提示；手机端与单片机通过wifi模块进行连接；外接液晶屏显示日期等信息。

**【实验功能概述】**

1. 使用单片机上的热感电阻进行室温检测
2. 扫描单片机键盘输入，通过单片机通信设定温度
3. 手机端使用wifi模块连接单片机，远程设定温度
4. 液晶屏显示日期等信息
5. 数码管切换显示【当前温度，设定温度】以及倒计时信息
6. 当倒计时结束/温度达到设定温度时进行LED灯闪烁提示

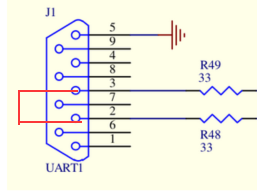
**【主要知识点应用及拓展】**

1. 单片机通信
2. RTC时钟
3. 键盘扫描
4. 温度模块、wifi模块、液晶屏模块、LED灯模块应用

**一、物理逻辑**

**【串口连接】**

由于使用是单台单片机进行，因此需要将单片机上的UART中的发送串口（TX，图中3）和接收串口（RX,图中2）连接起来

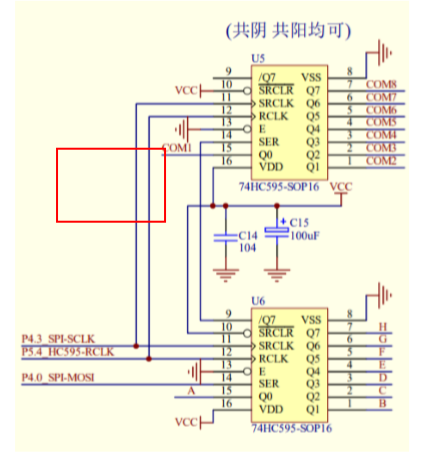


**【将串口数据通过USB输出】**

将实验板的P16与P30、P17与P31分别连接，使得串口1收发的数据可以转换到烧录使用的USB口，使得通信可以使用电脑端的串口助手进行观察。

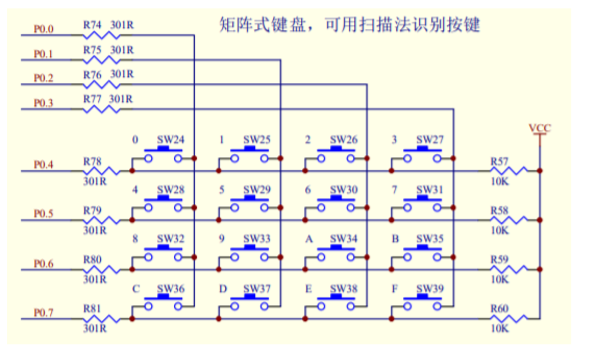
**【数码管显示部分】**

数码管显示时分秒，使用按键可以调节时和分，连接如图：



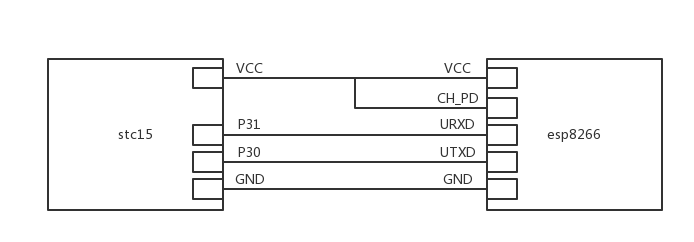
**【按键部分】**

按键部分使用扫描法识别按键的按下，并根据按键来进行对应的操作



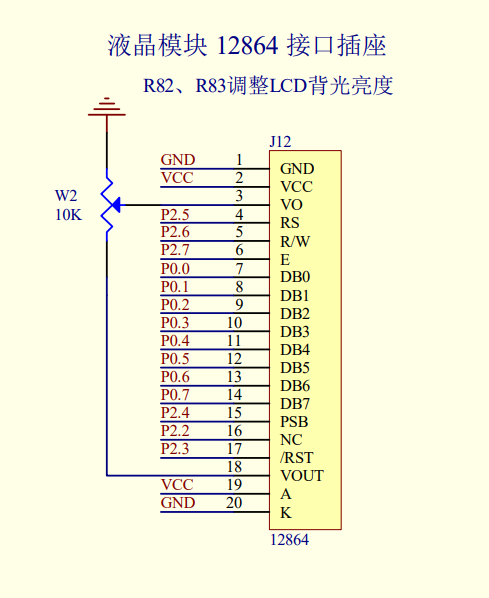
**【wifi模块部分】**

电路图（本来esp8266的VCC和CH\_PD应该接3.3V的，不过没在我们板上找到5V转3.3V的，就先直接插VCC。）

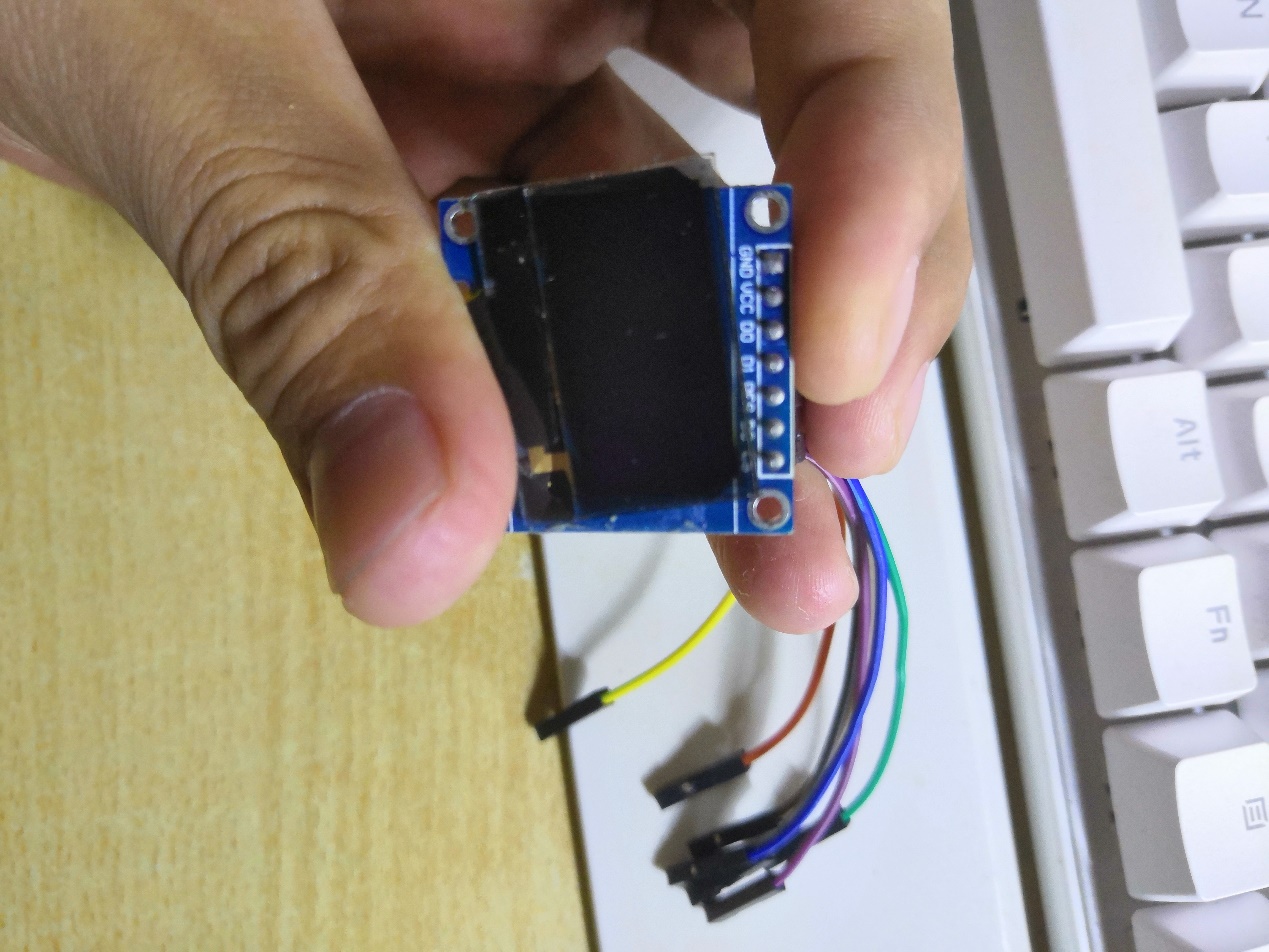


**【液晶屏模块部分】**

这是参考资料中给出的电路示意图，我们的0.96寸OLED显示屏需要和这个进行连接，然后我们需要在Keil上实现代码对显示屏进行显示控制。



我们所购买的OLED显示屏是7线的，上面分别是GND，VCC，D0，D1，RES，DC，CS



GND和VCC分别连接对应的接口就行，D0对应时钟，D1对应数据，RES复位，剩下的DC是数据命令选通端。

**【手机端部分】**

前端：使用html+css+js 实现前端界面的搭建。

通讯：使用websoket与Wi-Fi模块进行通讯。

**二、软件实现及原理分析**

**【功能原理】**

单片机启动时先初始化串口、按键等数据，然后进入循环：

1.检测是否收到新的设定温度数据/新的检测温度并进行处理。

2.按照周期刷新数码管的显示

3.按照周期扫描行列键盘，判断是否有按键操作；如果有键按下，则通过串口发送按键的代码

4.接收方接收到按键码放入缓存RX1\_Buffer，回到步骤1

**【使用方法】**

**单片机端：**

按下所有按键都会通过串口发送键位的号码，但只有0~3会被使用

按下单片机行列键盘的0：倒计时设定的小时+1

按下单片机行列键盘的1：倒计时设定的小时-1

按下单片机行列键盘的2：倒计时设定的分钟+1

按下单片机行列键盘的3：倒计时设定的分钟-1

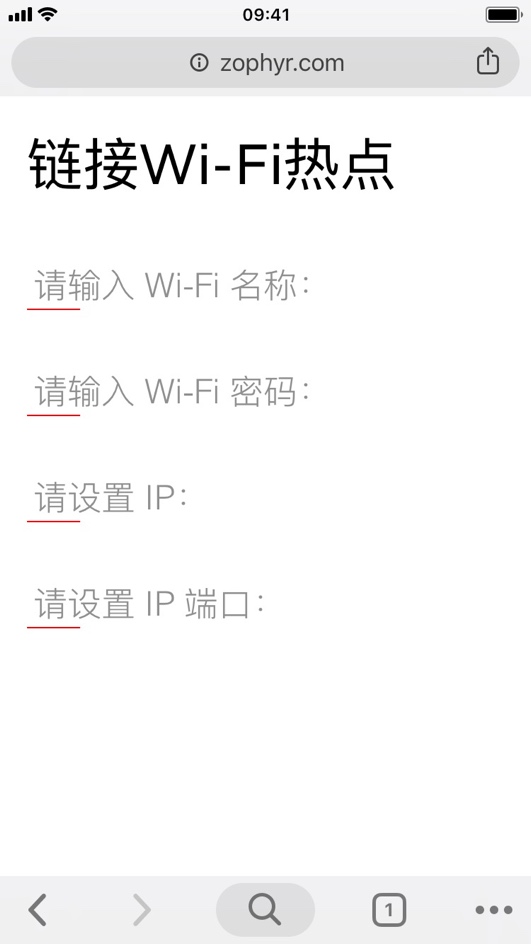
按下单片机行列键盘的4：将倒计时重置

按下单片机行列键盘的7：进入倒计时设定功能，此时数码管切换显示停止，时钟倒计时停止

按下单片机行列键盘的11：完成倒计时设定，数码管重启切换显示，倒计时开始

**手机端：**

1. 打开网页



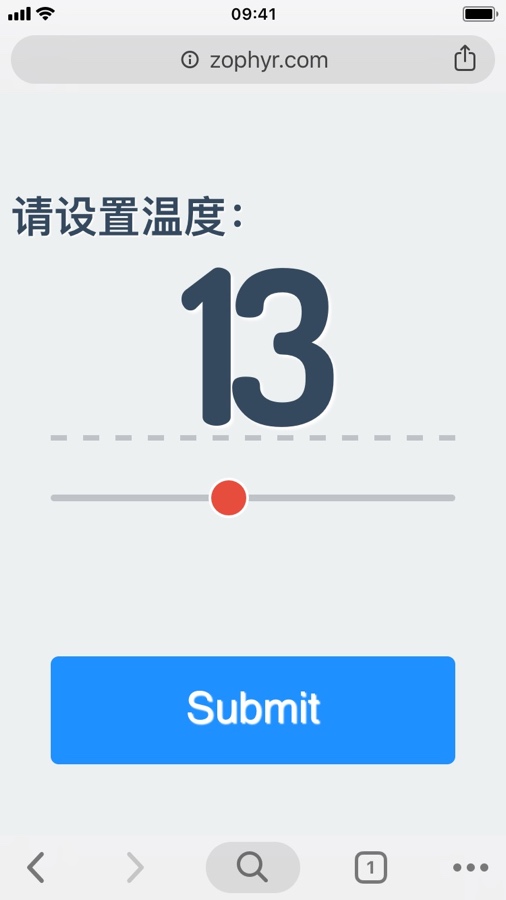
1. 输入相应的Wi-Fi名称、密码、设置访问IP、IP监听端口



1. 点击链接，进入温控界面。



1. 拖动滑块，调整设置的温度。

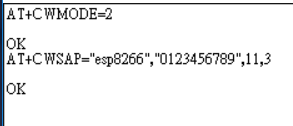


1. 点击‘submit’提交设置的温度。



**【wifi端】**

设置接入点名称为esp8266，密码为0123456789，通道号为11，加密模式3（WPA2\_PSK）



模块的ip为192.168.4.1

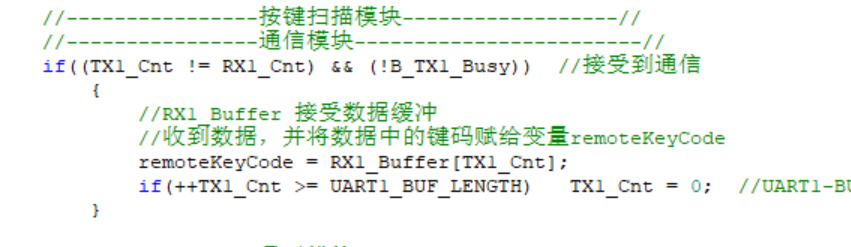


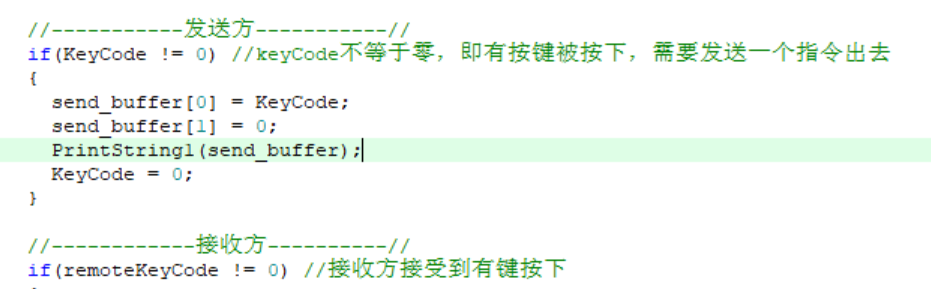
实现效果：



**【具体功能实现】**

1.通信功能（单片机串口）实现

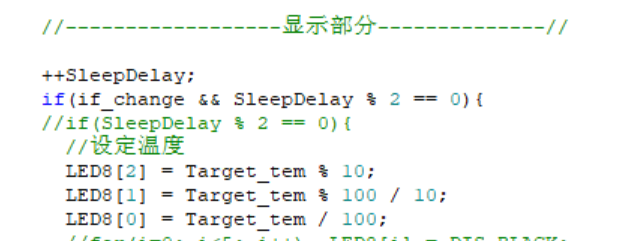




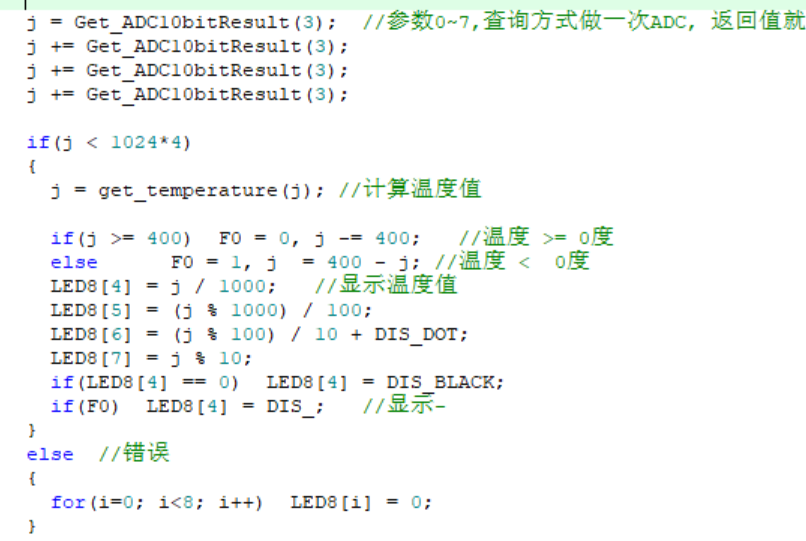
2. 键盘扫描模块（使用XY查找4x4键的方法）



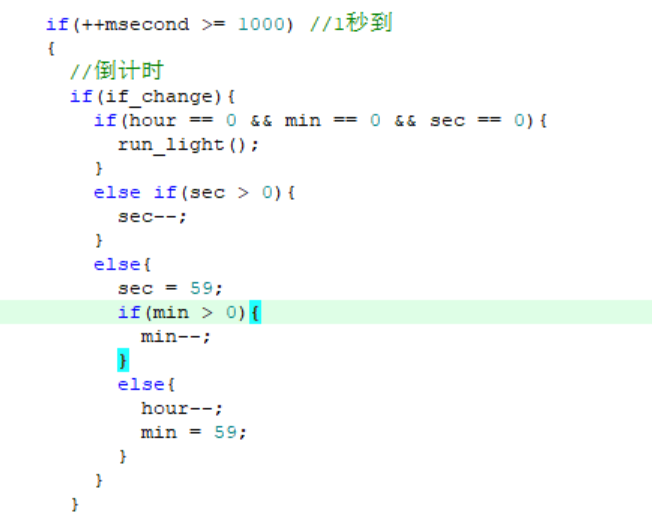
3.LED显示模块（设定温度）



4.实际温度检测并显示



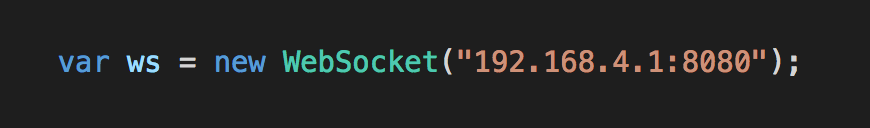
5.倒计时计算



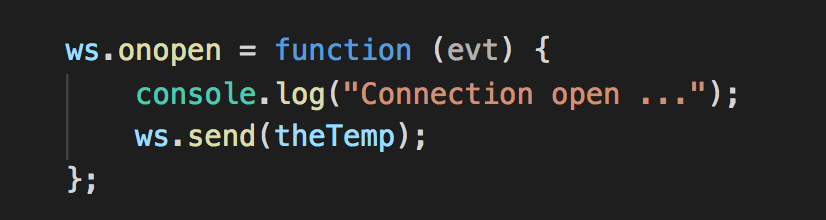
1. 前端与Wi-Fi模块通信

与Wi-Fi模块通信，本质上是TCP链接的建立。我们可以使用websocket来非常简单的实现。

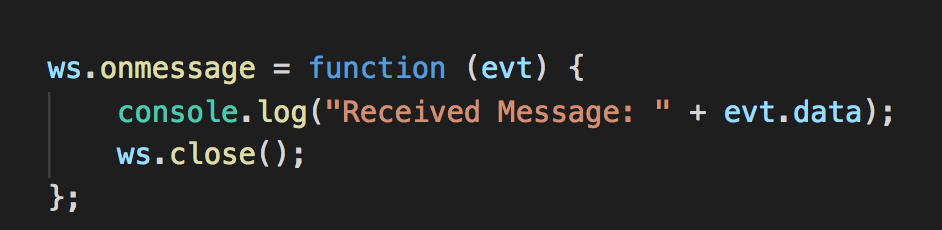
建立 socket 连接：



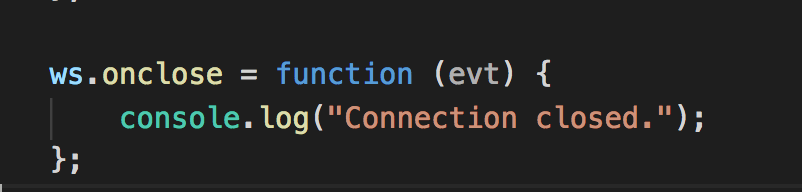
当与服务器通信时，log出提示并且发送温度设置：



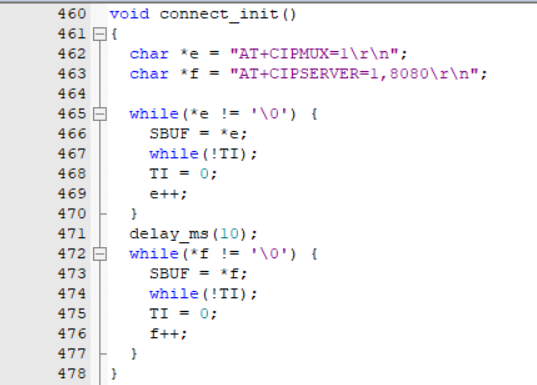
回调函数，现实接受的信息：

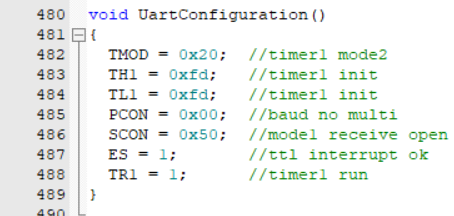


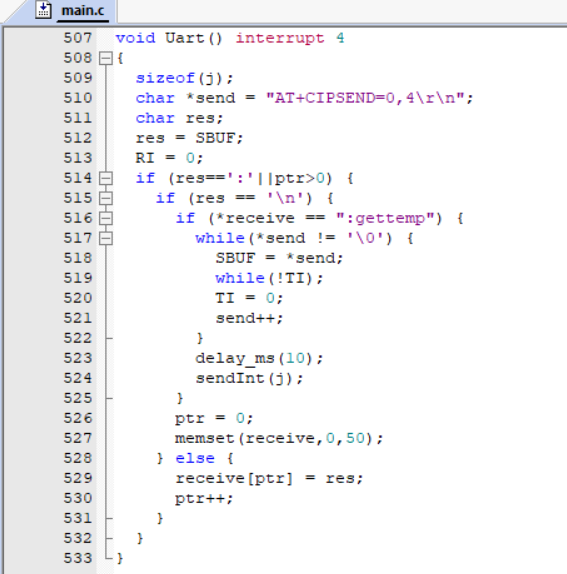
结束通行，say goodbye：



7.WiFi模块实现

开启多连接模式和服务器模式，端口号8080  


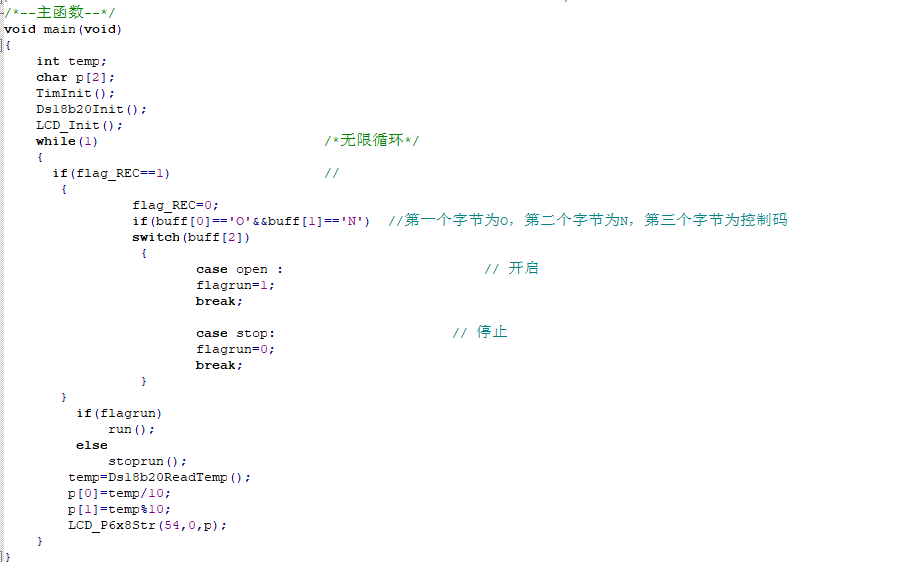
串口配置，配置内容如注释  


中断接受，接受内容格式如图：  
  
有用的信息只有冒号后的东西，如果冒号后是gettemp，就传送温度过去。  


8.液晶屏模块（单独实现）

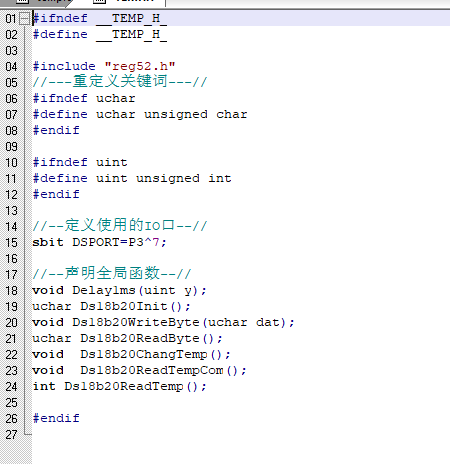
代码实现：

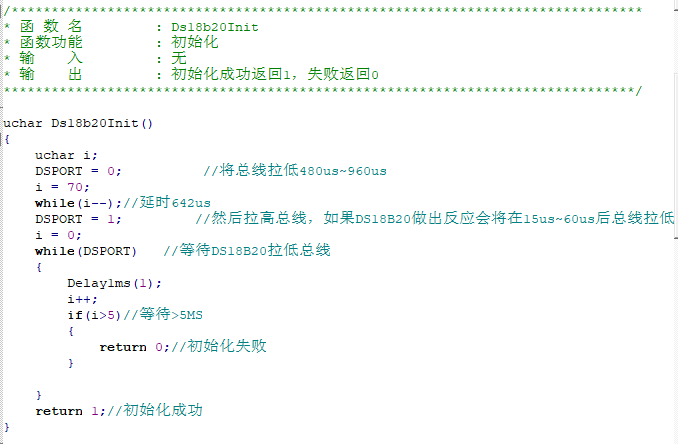
主函数首先需要调用多个初始化函数，对晶振、波特率还有我们的显示屏进行初始化。然后进行不停的循环，在循环中我们根据复位信号的变化对显示屏进行相应处理。buff是一个数组，用于接受缓冲字节。然后下面的LCD\_P6x8Str()是一个显示6\*8一组标准ASCII字符串 显示的坐标（x,y），y为页范围0～7.

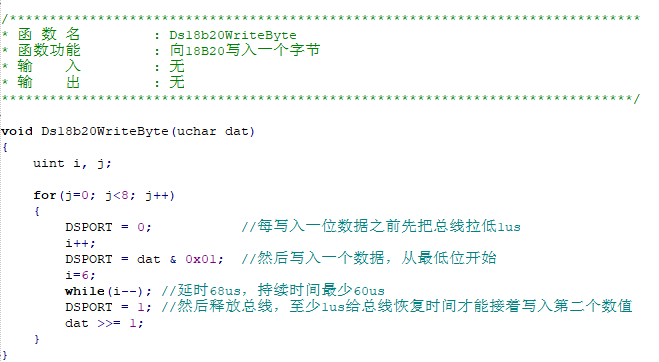


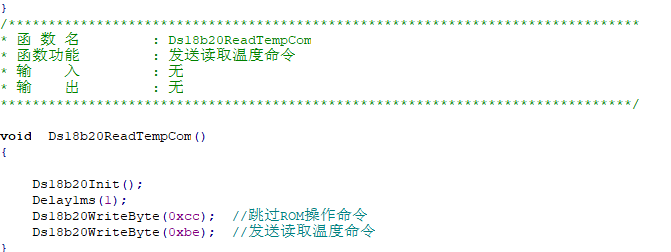
除了主函数我还实现了头文件lcd和temp以及i2c。首先介绍头文件temp。

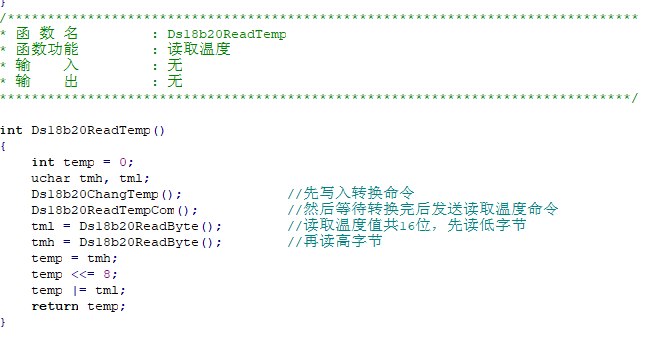
下面是temp.h的定义，定义使用的IO口以及几个全局函数。主要是和温度模块进行交互，能够读取温度和写入温度



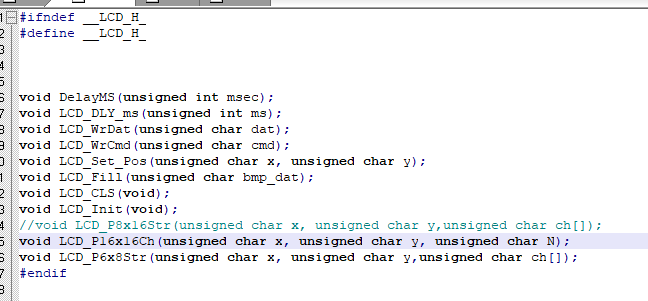




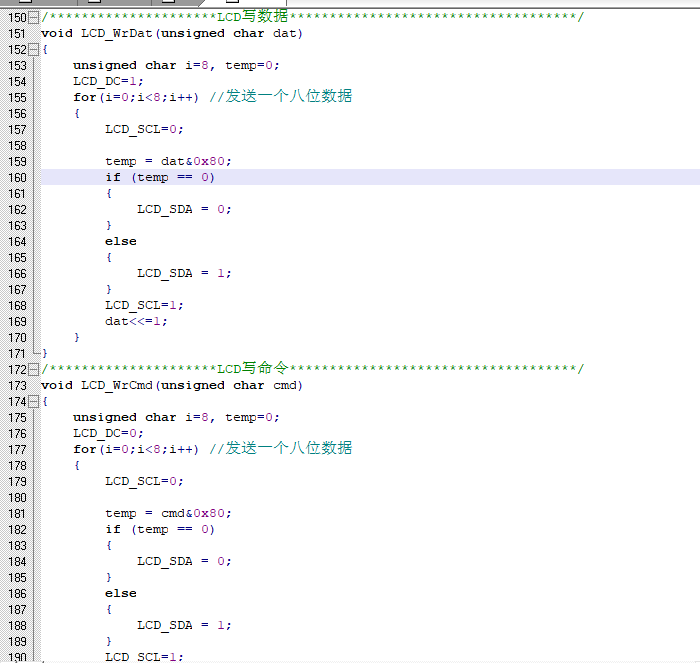




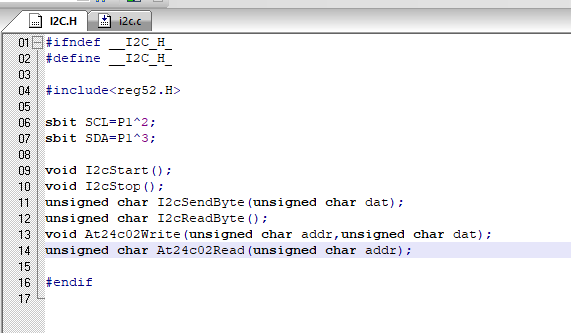
然后是头文件lcd，下面给出该头文件的定义以及比较重要的两个函数，它们分别是写数据以及写命令



一个数据是8位的，我们需要对每一位都单独进行操作。对相应的信号需要进行处理

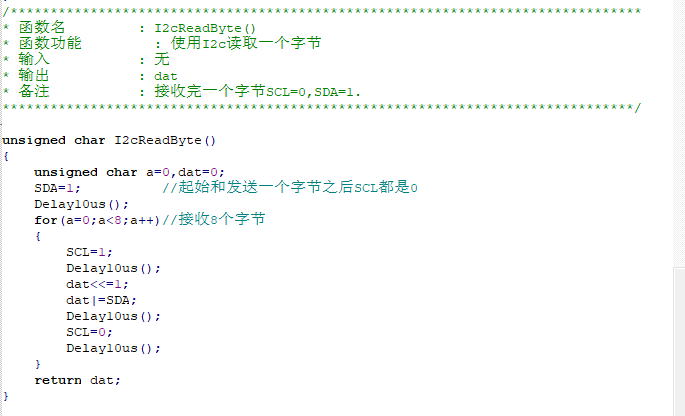


最后是i2c部分，这个部分是OLED的核心，需要以下六个函数：



I2cStart和I2cStop分别是起始函数和中止函数。I2cStart函数的功能是在SCL时钟信号在高电平期间SDA信号产生一个下降沿；I2cStop函数的功能是在SCL时钟信号高电平期间SDA信号产生一个上升沿。这两个函数比较简洁，在这里不过多叙述。

这个是读取字节的函数，我们每次接受八个字节，每接收一个都进行一定程度的延时，这是为了保证整个过程能够按照既定顺序不会发生碰撞。



**三、实验分析及思考**

1.这次试验参考了之前做过的通信和RTC时钟实验，以及stc样例，最后决定做一个具有现实应用意义的东西，也符合老师在课堂上提到的物联网的思维。

2.由于我们之前并没有对单片机以及相应的模块进行学习或研究，因此在基础的配置上（包括液晶屏的点亮、wifi模块的通信、手机端与单片机连接等部分）花费了不少的时间，也从中学到了许多关于嵌入式软件和硬件的知识。

3.由于对工作量的预估不明确，因此我们一开始想实现的功能其实是更多而且更广了，然而由于时间的限制和技术的不足（流下了没技术的眼泪.jpg），因此最后需求一再缩小，这也提示我们作为产品一定要从实际需求出发，在版本的迭代中更新功能，而不是一开始就提出大量无法实现的需求。

**四、附录**

1.实验分工

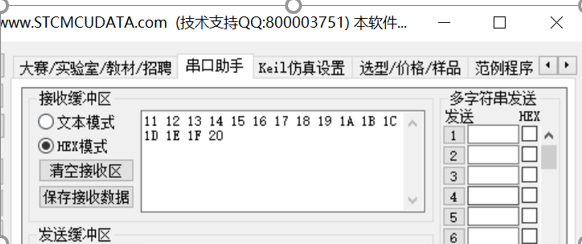
整个大作业由小组全体成员共同合作完成

2.源码及实验现象

（1）板上的源码见附件文件夹【源码】

（2）手机端的源码见github：https://github.com/Xiongzj5/embedded-design

（3）将串口输出放到USB串口后，可以通过串口助手观察到数据



（4）具体实现使用视频方式进行展示，百度云网址如下:

https://pan.baidu.com/s/16gbIrz4GJRTUhPmPRxQY\_Q

3.最初的产品说明文档：https://shimo.im/docs/NSSFMfxFUy8waVrc