

开放性创新实践——硬件报告 *

姓名: 王庚霖 班级: 崇新学堂 学号: 202020120236

June 28, 2022

需求简述

基本功能:

- 1) 当前的年、月、日、时、分、秒显示
- 2) 设置年、月、日、时、分、秒
- 3) AM、PM 设置与显示
- 4) 农历年月日、二十四节气显示
- 5) 闹钟时间设置，闹钟时间到进行声光报警提示
- 6) 温湿度检测与显示

高级功能:

- 1) 闹钟音乐选择（采用 TF 卡或录音芯片存储音乐数据）
- 2) 手机通过蓝牙或 WiFi 网络连接该万年历，通过手机 APP 可以完成基本功能中的所有设置功能
- 3) 通过编写 Android 手机程序获得网上天气预报数据，并将该数据通过蓝牙或 WiFi 网络传递给音乐万年历，万年历可以完成天气信息显示
- 4) 火灾检测与报警，报警形势可以是短信。

实现步骤:

- 1) 利用国产立创 EDA 设计原理图。
- 2) 使用面包板进行基础功能仿真测试
- 3) 利用 Altium Designer 设计该系统的 PCB 板图，该板图能够进行厂家加工
- 5) 焊接元器件进行实物调试
- 6) 编写设计报告，包含相应的软硬件设计思路描述，设计效果展示

*

硬件设计原理图

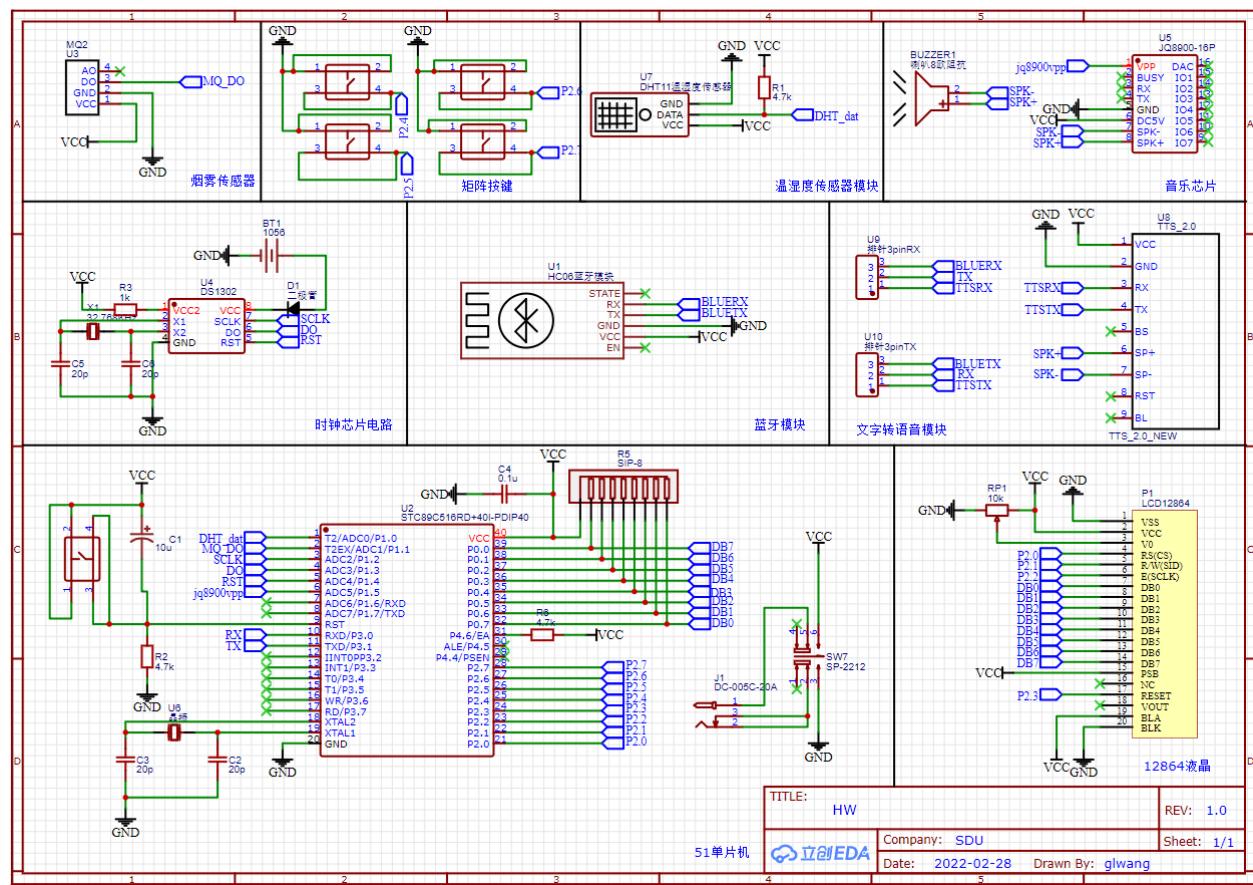


Figure 1: 原理图概览

单片机系统: 采用独立元器件构成复位电路和晶振电路，通过 DC-005 接口与外部开关电源适配器连接进行供电，无需考虑供电稳压和最大供电电流 (HUAWEI 电源适配器供电电压 5V，最大电流 2A)。

DS1302 时钟系统: 采用独立元件和时钟芯片构成电路：芯片管脚 1 接电源 VCC，串联电阻降低电压冲击；芯片管脚 2,3 接晶振稳频电路，含 32.768 无源晶振，两枚 20p 稳频电容，保证起振平稳；芯片管脚 8 通过二极管接独立电源进行涓滴电流供电，保证断电不掉时钟数据；芯片管脚 5,6,7 作为控制引脚分别接单片机 I/O 口 P1.4,P1.3,P1.2。

RX/TX 串口通信: 单片机串口可以通过跳帽选择与 TTS 模块（文字转语音）或者 HC-05 蓝牙模块相连，连接蓝牙模块可以实现手机与单片机的通信，而 tts 模块主要功能为文字转语音，是对 JQ8900 音乐模块的补充。

JQ8900 音乐模块: JQ8900 模块采用一线控制模式，其 VPP 管脚连接单片机 I/O 口 P1.5，通过外接导线连接 8W0.5Ω 参数喇叭。

LCD12864 显示屏: 12864 采用并行输入模式，DB0 7 分别与单片机 PO.7 0 连接，RS, RW, E,RESET 分别接单片机 IO 口 P2.0, P2.1, P2.2, P2.3, V0 接电位器触电，控制屏幕对比度。

其他: 均采用成熟模块和厂家原理图。

PCB 制板策略

第一次制版：

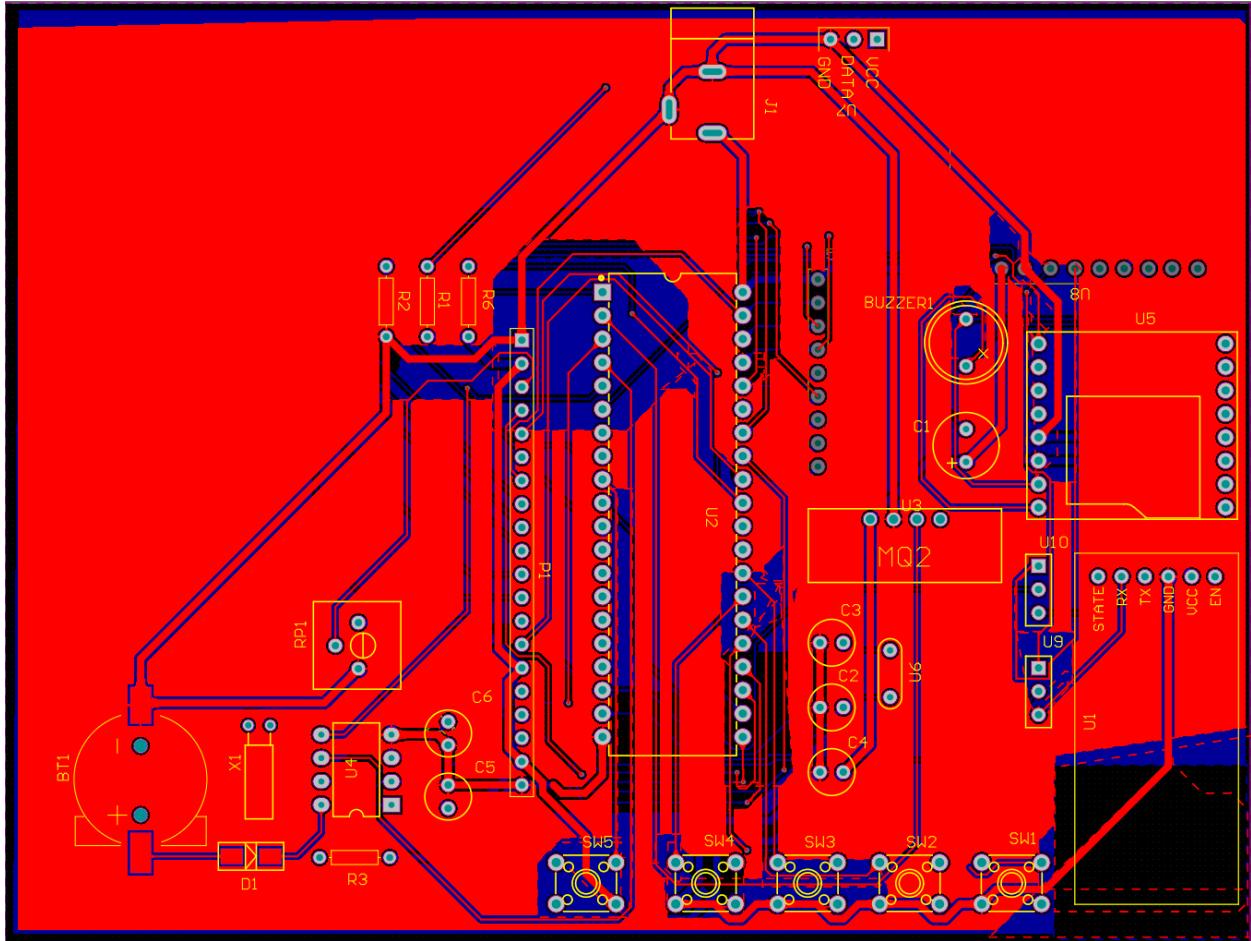


Figure 2: 终版-PCB 正面

板长 13cm，宽 10cm。

lcd12864 位于板卡左侧，出于节约空间的考虑，我将时钟电路安排在 lcd12864 下方，保障其稳定性，避免接触影响 DS1302 走时。

板卡右侧分布着 MQ-2、JQ8900、TTS 模块、蓝牙等模块和元器件，供电口位于板卡上方，DC005 直接与外部电源适配器相连。

不足之处在于：

- 1) lcd12864 距离 P0 口太远，需要过孔穿过单片机上方，可能会使得屏幕并行显示不够稳定。
- 2) 对 MQ-2 的引脚考虑不周，妨碍到 JQ8900 的安装，并且碍于美观。
- 3) 缺少电源开关，调试需要重复拔插电源，造成调试上的困难。
- 4) 大量空间浪费，不够紧凑，超过嘉立创 10x10 免费打样限制，提高了成本。
- 5) 使用 AD 自动布线功能，可能会有未知的 bug，已经出现独立按键的检测问题。

第二次制版：

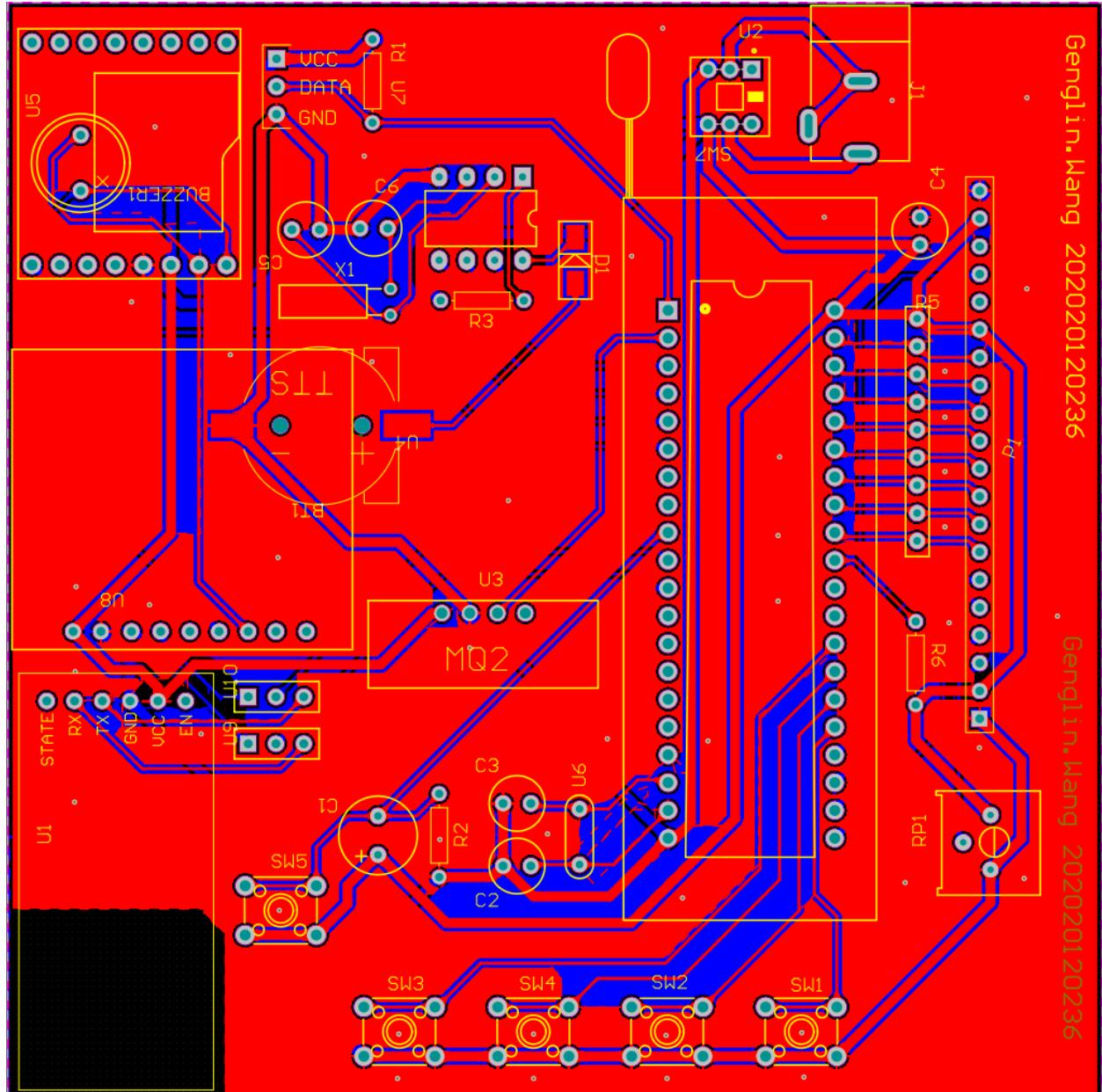


Figure 3: 终版-PCB 正面

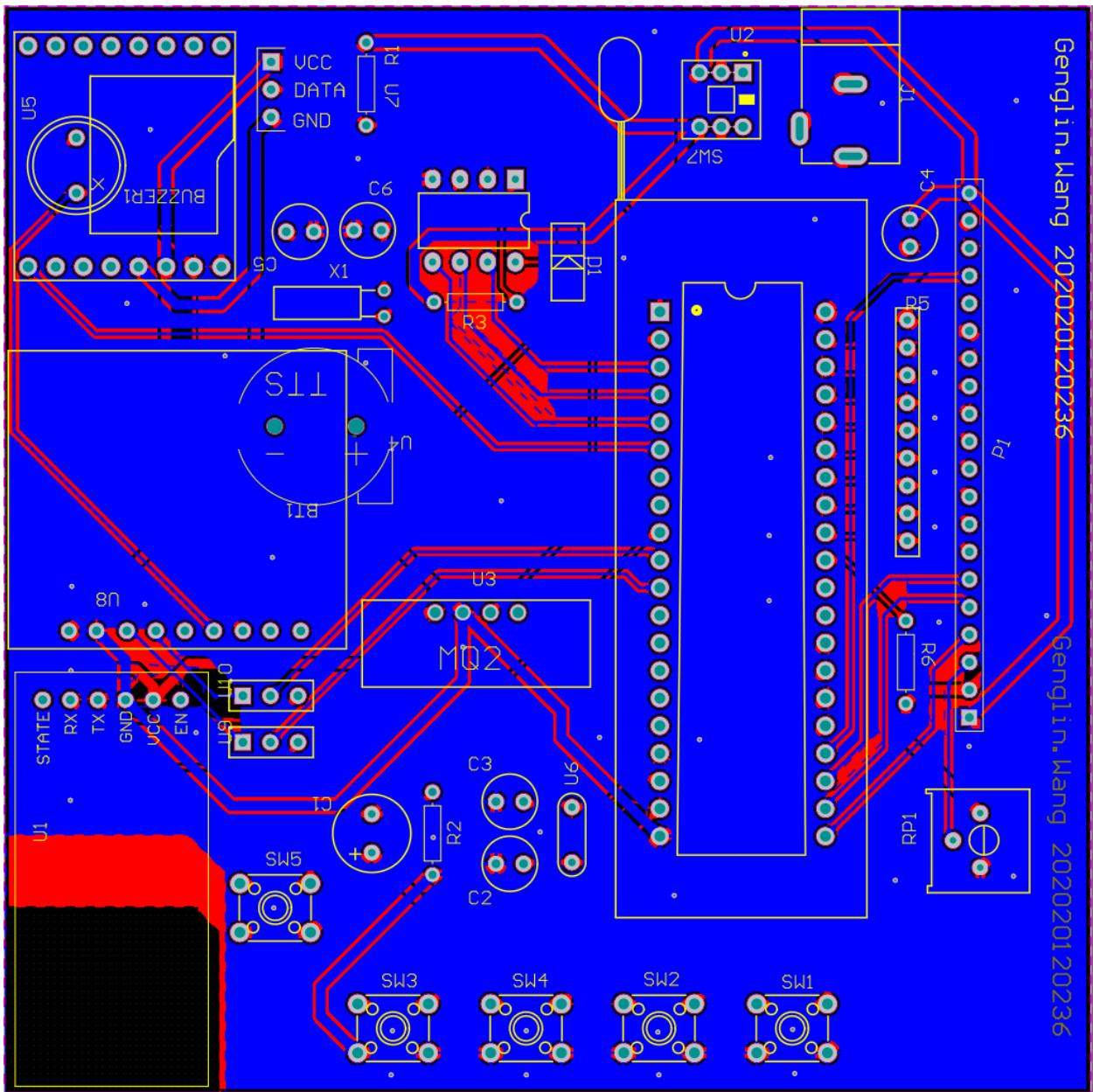


Figure 4: 终版-PCB 反面

第二版板长 10cm，宽 10cm。

总体上各种模块基本集中在板卡左侧，出于节约空间的考虑，我将时钟芯片电路的电池部分埋在 TTS 模块下方。

我认为第二次的设计相当精良，主要体现在：

- 1) 成片美观，各种元器件井然有序，版面空间利用得当。
- 2) 充分考虑元器件的稳定性问题：将晶振和稳频电路远离时钟；lcd12864 靠近并行输入 P0 口；大面积铺铜提高稳定性和降低加工难度；增加过孔以促进回流等；为蓝牙留出挖空铜皮，尽可能避免电磁屏蔽保证蓝牙通信稳定性。
- 3) 本次布线由我自己进行布线，布线优于第一版自动布线：TOPlayer 和 BOTTOM-layer 的布线尽可能垂直分布，P0 口和 12864 靠的比较近，这些都有利于稳定性。

实物图

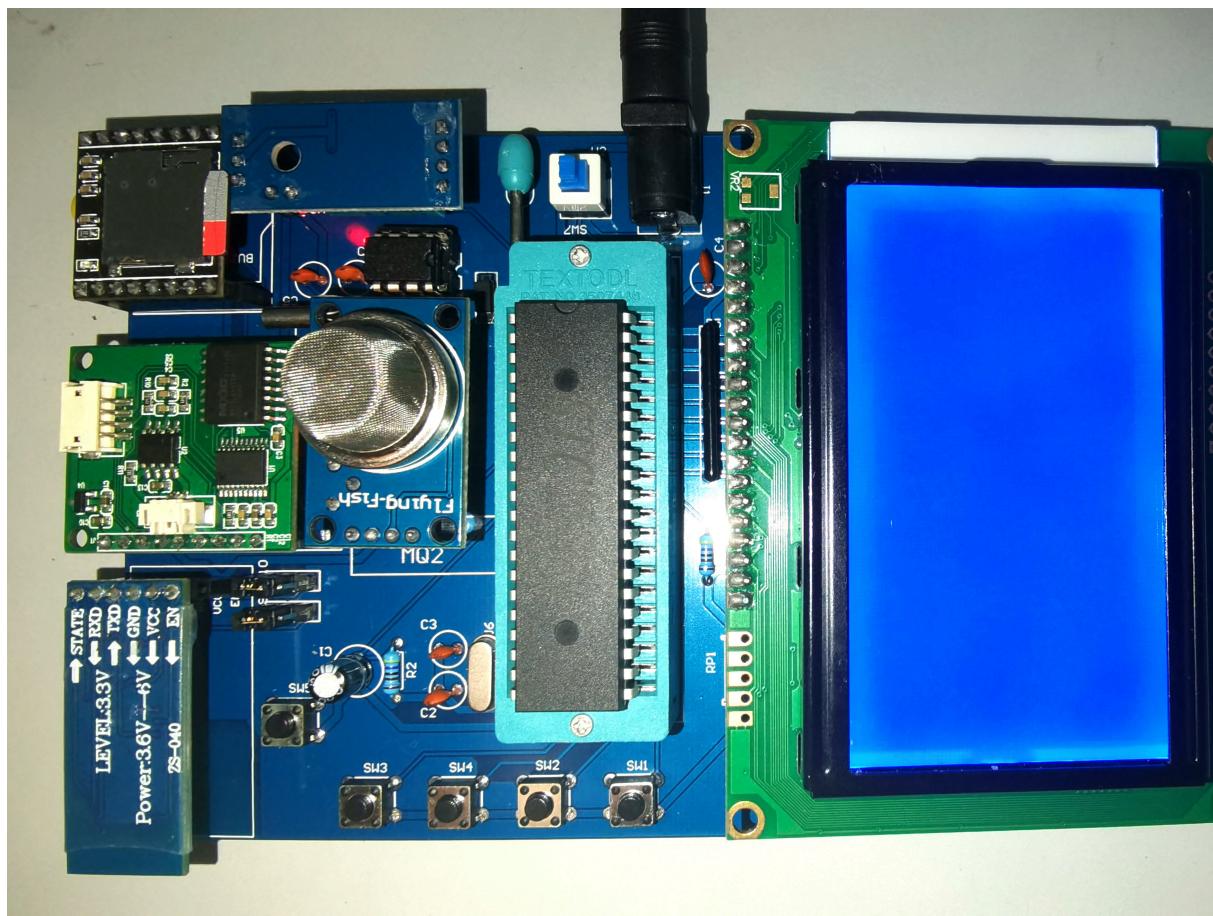


Figure 5: 实物图

keil 代码结构和设计思路

总结结构：

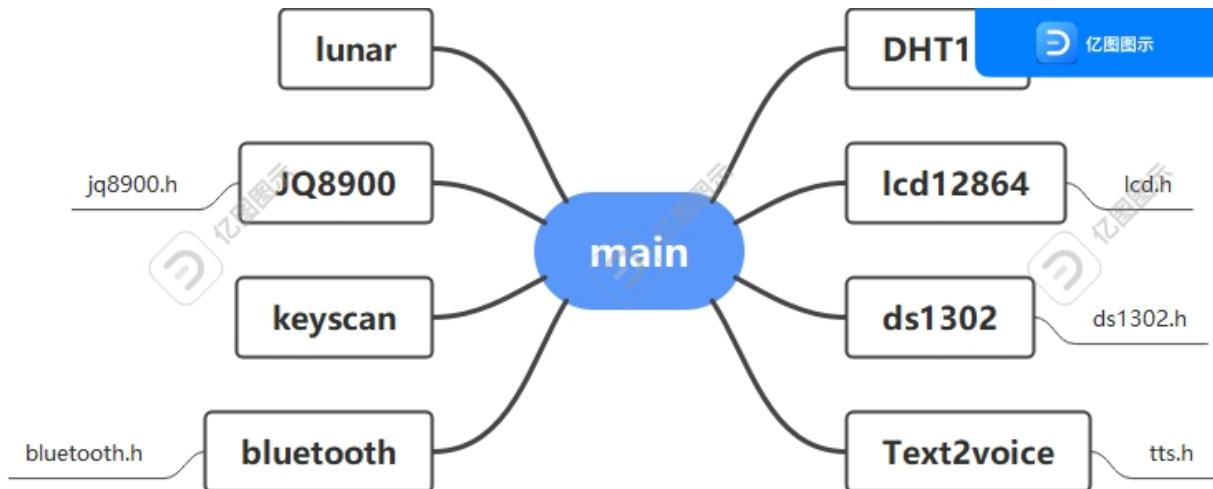


Figure 6: 总结构图

main.c

含 main 函数，菜单呼出子函数，基础功能显示子函数，天气获取子函数。

main 函数中涉及部分基础显示功能，查看闹钟时间功能，并负责检测闹铃时间和烟雾报警器；菜单呼出子函数负责切屏并呼出其他单元；基础功能显示子函数包含了农历年月日显示，阳历年月日显示功能；天气获取子函数和 App 配合使用可以在手机上解析 json 天气数据并在单片机上显示。

12864.c

含 LCD12864 所有驱动程序：包括显示，忙检测等；此外还有八位数据交换子函数，主要是由于 DB0 7 与 P0.7 0 分别连接（这样硬件会优化），序号相反，而网络上一般驱动大多是 DB0 7 与 P0.1 7 相连，因此使用该函数交换 0-7 的数据符合硬件设计。

ds1302.c

含 DS1302 所有驱动程序，时间信息格式化显示子函数，修改时间子函数，设置闹钟时间子函数，闹钟检测子函数，蓝牙设置时间处理子函数。

DS1302 驱动程序含：DS1302 写时间、读单字节时间、读时间数组、初始化函数；时间信息格式化显示子函数将 DS1302 全局变量 TIME 数组格式化为 LCD12864 可以显示的字符数组；修改时间子函数负责建立临时变量存储被修改的时间并在返回前写时间；设置闹钟时间子函数负责建立临时变量存储闹钟时间并在返回前将闹钟信息存储到全局变量中；闹钟检测子函数读取当前时间并和闹钟时间比对，以检测闹钟是否到来；蓝牙设置时间处理子函数被蓝牙中断子程序调用，直接对用以修改当前时间和闹钟时间。

dht11.c

含温湿度数据获取及格式化驱动。

lunar.c

含获取农历月、日子程序，以及存储了农历信息大数组。

JQ8900.c

含 JQ8900 数据发送驱动程序和闹钟音乐选择子函数。

闹钟音乐选择子函数实现在 6 种音乐中自由切屏选择。

keyscan.c

单点模式按键检测子函数，函数返回键值可以调用控制模式，切屏，选择确定等。

bluetooth.c

含串口初始化函数，串口中断数据处理子函数，字符发送子函数。

其中串口中断数据处理子函数接收手机发送的单字节数字并根据该数字进行 switch 条件分支选择功能；字符发送函数用来向手机发送单字节数字，主要服务于 MQ-2 火灾短信信息发送功能。

Text2voice.c

含串口初始化函数，字符发送子函数，通过标准 io 库 stdio.h 之 printf 发送语音数据并播报。

App 结构与设计思路

组件设计

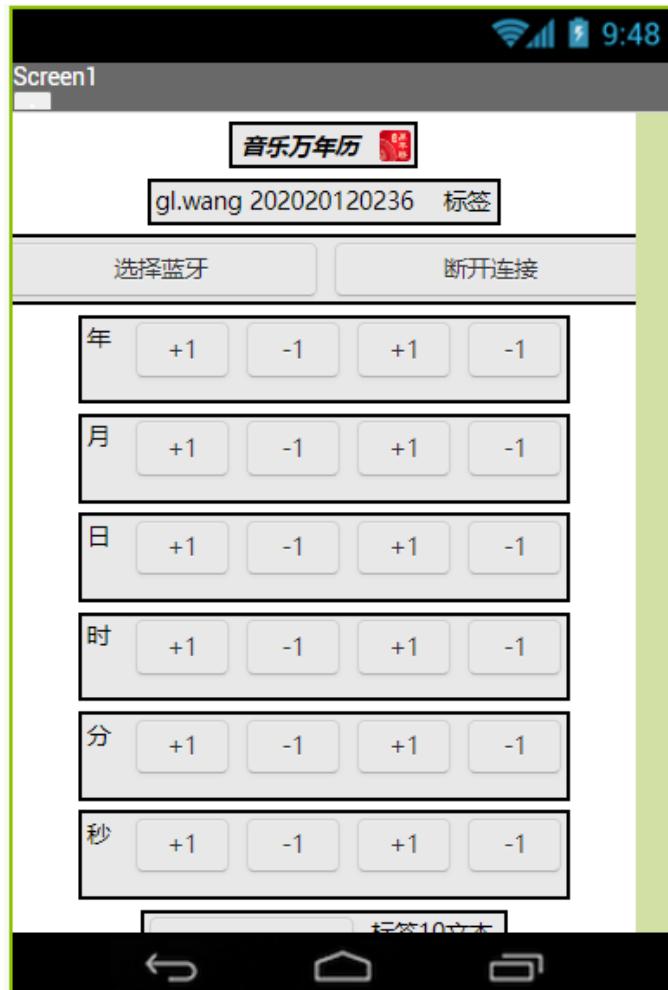


Figure 7: App 界面

我使用 Appinventor 进行简易 app 开发。App 界面设计较为简单：界面顶端为“音乐万年历”，并附个人信息和显示蓝牙连接情况的标签。

下方为蓝牙连接的列表选择框和断开蓝牙连接的按钮。初始条件下“选择蓝牙”是启用状态，而断开蓝牙是灰色即未启用状态。单击“选择蓝牙”即可显示蓝牙列表框，选择并连接成功后“选择蓝牙”变为灰色即未启用状态，断开蓝牙可以启用。

页面主体为年月日时分秒的按键设计，左方 +1, -1 为日历时间的增减按钮，右方 +1, -1 为闹钟时间的增减按钮（位于右侧的年月日 +- 并没有实际意义）。

页面最底端为“获取天气”按钮和用来显示天气信息的标签文本。按下“获取天气”按钮：如果单片机未处于“获取天气界面”，则会在标签文本处显示当前时间，天气，温度和风向；如果单片机处于“获取天气界面”，那么不仅仅手机 App 标签处显示当前时间，天气，温度和风向，还会在万年历 LCD12864 上显示当前天气。在手机上爬取当前时间，天气，温度和风向属于额外 bonus 功能。

逻辑设计

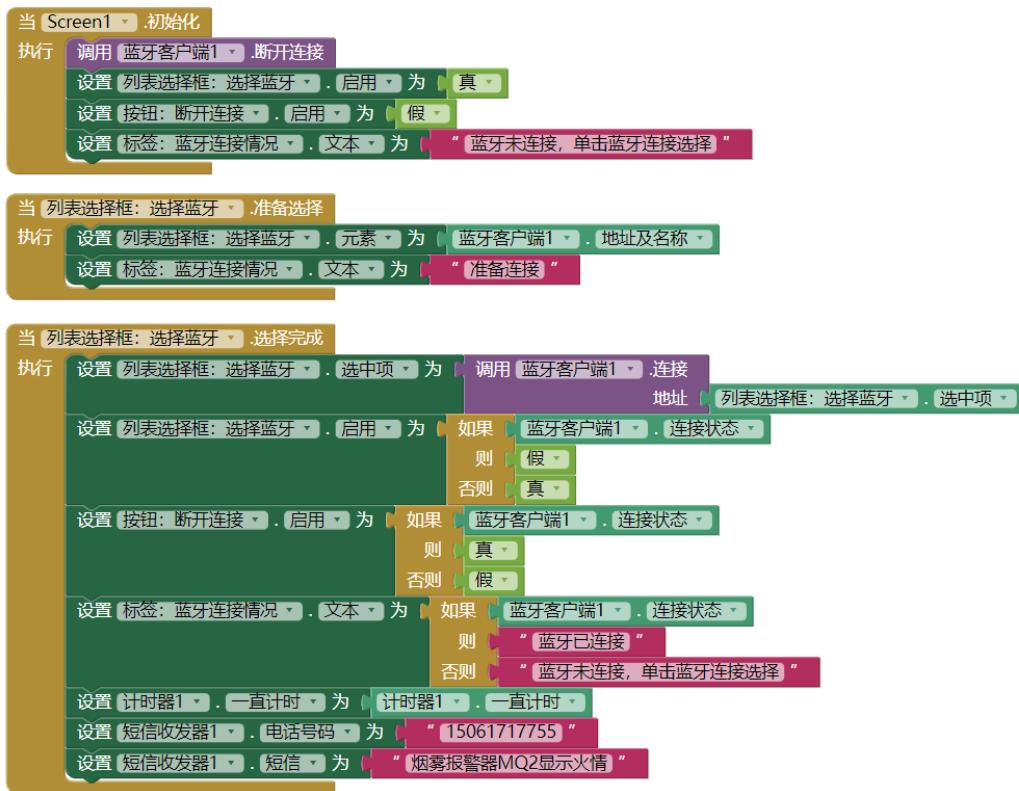


Figure 8: 蓝牙连接逻辑及短信报警设计



Figure 9: 基本设置功能概览

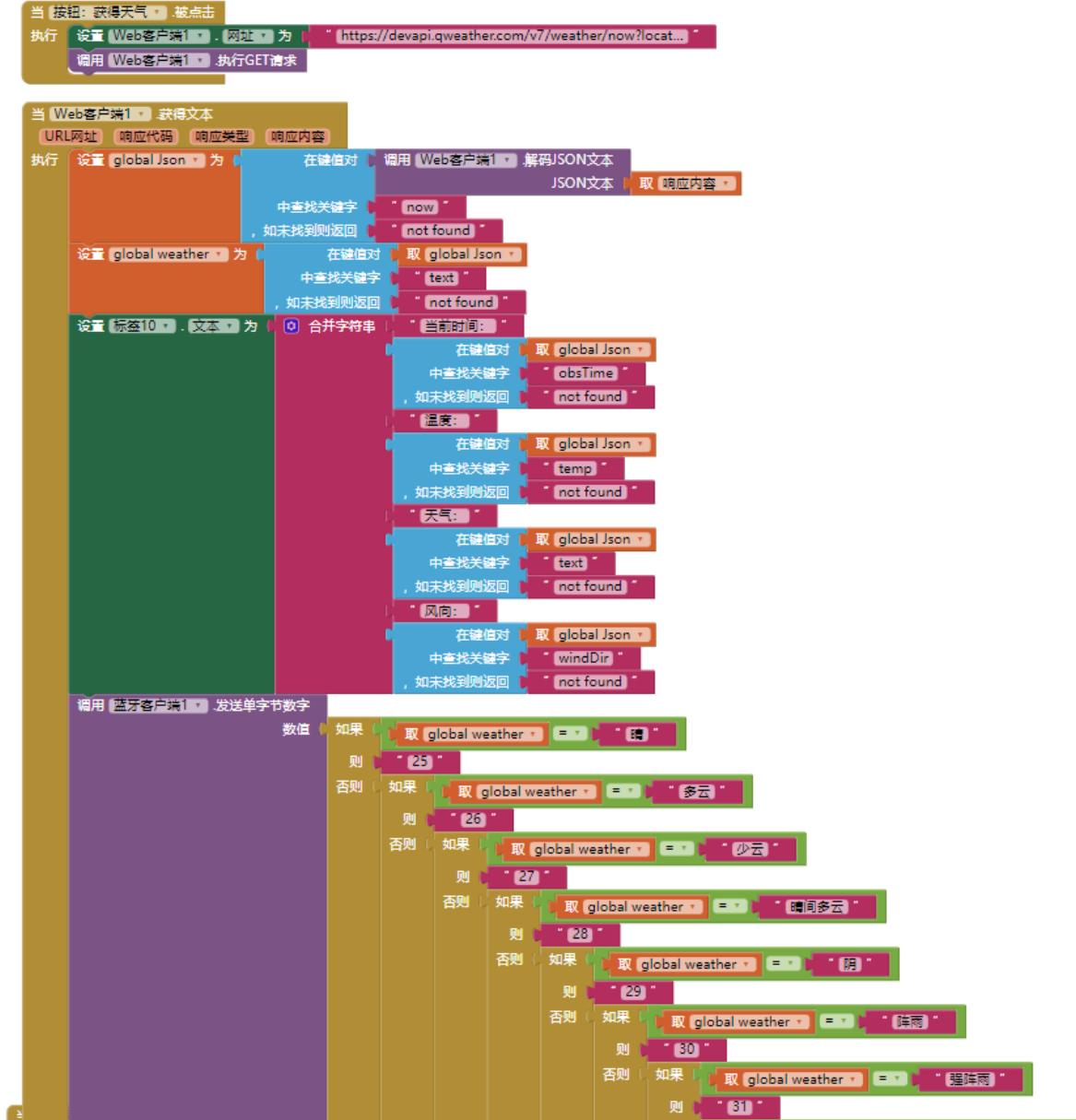


Figure 10: 天气 API 调用及数据拼接发送

蓝牙连接逻辑 见 Figure 8

应用组件: 蓝牙客户端, 按钮, 计时器

在 App 界面被初始化时, 我们启用选择蓝牙的列表并提示蓝牙未连接。一旦连接上蓝牙, 蓝牙选择列表被禁用, 这时我们可以点击取消连接断开蓝牙。

在连接成功后, 我们开启定时器, 定时器的功能是定时接收来自单片机的火情数据并根据数据判断是否发送活在短信信息。并设置发送的电话号码为我本人的手机号, 短信内容为“烟雾报警器 MQ-2 显示火情”。

基本设置功能 见 Figure 9

应用组件: 蓝牙客户端, 按钮

当某一按键被点击, App 向单片机发送单字节无符号整数, 单片机根据该数字进行 switch 条件分支判断符合某一条件, 并完成具体设置。如单击年右侧第一个 +, 那

么 App 向单片机发送单字节无符号整数 0x01，单片机接收到 0x01 判断应该调用子程序 blue2time(1, 6)，即上文所述蓝牙设置时间处理子函数，该函数的作用是对时间数组 TIME 年这一位进行 +1 操作，并调用 DS1302 驱动之写时间函数将新时间写进 DS1302，从而完成 App 时间设置。

天气 API 调用及数据拼接发送 见 Figure 10

应用组件：Web 客户端，蓝牙客户端，按钮，计时器

所使用的天气信息来自和风天气 API，信息返回为 json 格式，如下示例是我在 6 月 27 日下午 15:52 获取的 json 信息（时间会有约 10-20 的延时）：

```
{  
    "code": "200",  
    "updateTime": "2022-06-27T15:42+08:00",  
    "fxLink": "http://hfx.link/2xc1",  
    "now": {  
        "obsTime": "2022-06-27T15:36+08:00",  
        "temp": "27",  
        "feelsLike": "29",  
        "icon": "104",  
        "text": "阴",  
        "wind360": "225",  
        "windDir": "西南风",  
        "windScale": "3",  
        "windSpeed": "15",  
        "humidity": "94",  
        "precip": "0.0",  
        "pressure": "992",  
        "vis": "8",  
        "cloud": "89", "dew": "24"},  
    "refer": {  
        "sources": ["QWeather", "NMC", "ECMWF"],  
        "license": ["no commercial use"] }  
}
```

首先我们设置两个 global 全局变量。json 和 weather，当按钮“获得天气”被点击，Web 客户端地址被设置为我所申请的和风天气 API 网址，并执行 GET 请求。当 App 的 Web 客户端从 API 获得文本，我们将 json 设置为键值对中的 now 字段，继而对 now 进行解析取其中天气字段 text。然后我们对 json 下关键字 obsTime, temp, text, winDir 进行拼接，获得最终字符串显示在标签上。

继而我们调用蓝牙客户端，对不同的天气我们进行判断，并发送对应单字节数字。如当天气为“晴”，发送单字节数字“25”，在单片机进行判断并显示。

功能测试与结果

- 1) 当前的年、月、日、时、分、秒显示
- 6) 温湿度检测与显示



Figure 11: 显示年、月、日、时、分、秒

- 2) 设置年、月、日、时、分、秒
- 3) AM、PM 设置与显示

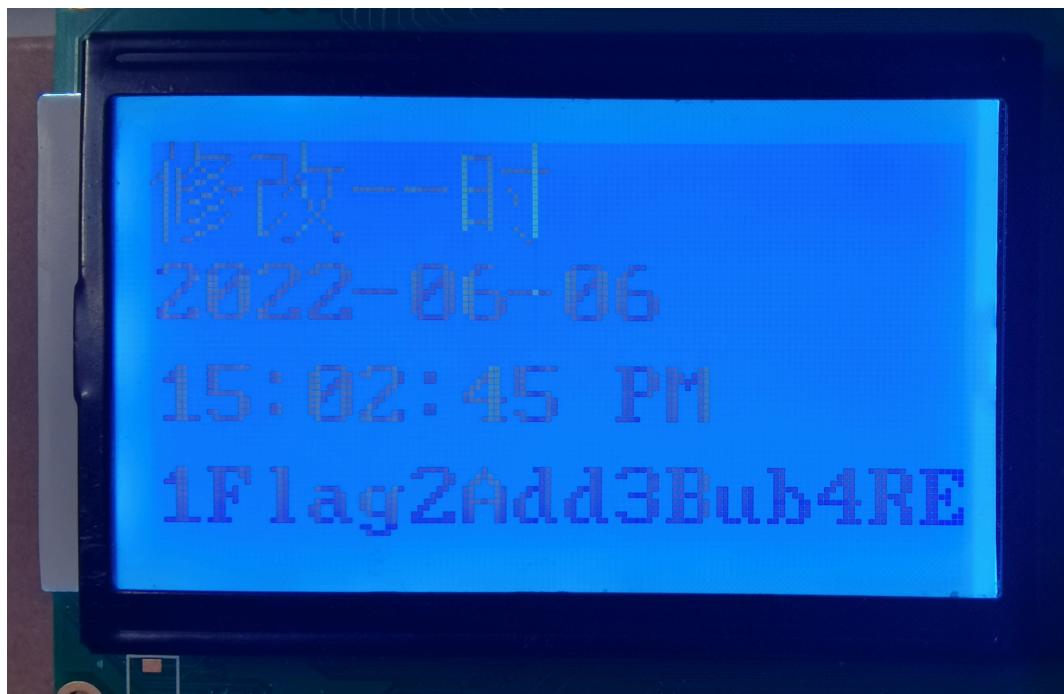


Figure 12: 设置年、月、日、时、分、秒、AM、PM

4) 农历年月日、二十四节气显示

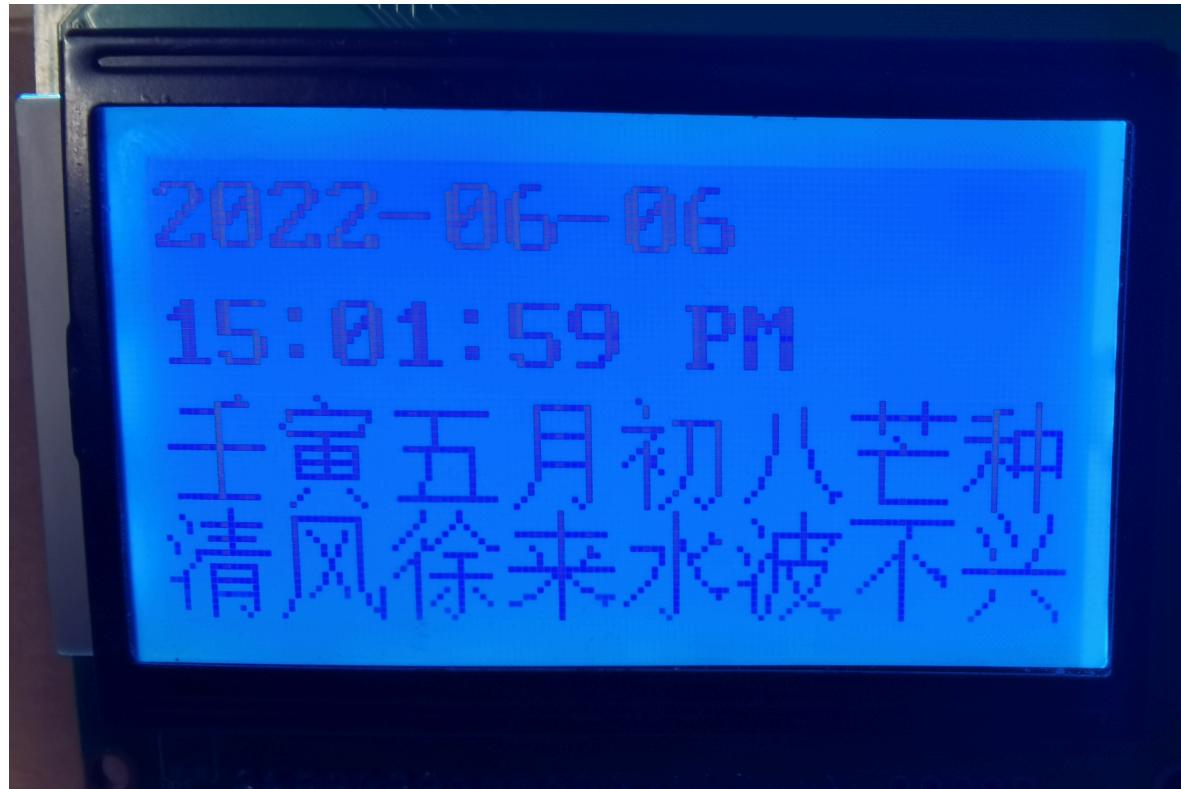


Figure 13: 农历年月日、二十四节气

5) 闹钟时间设置，闹钟时间到进行声光报警提示

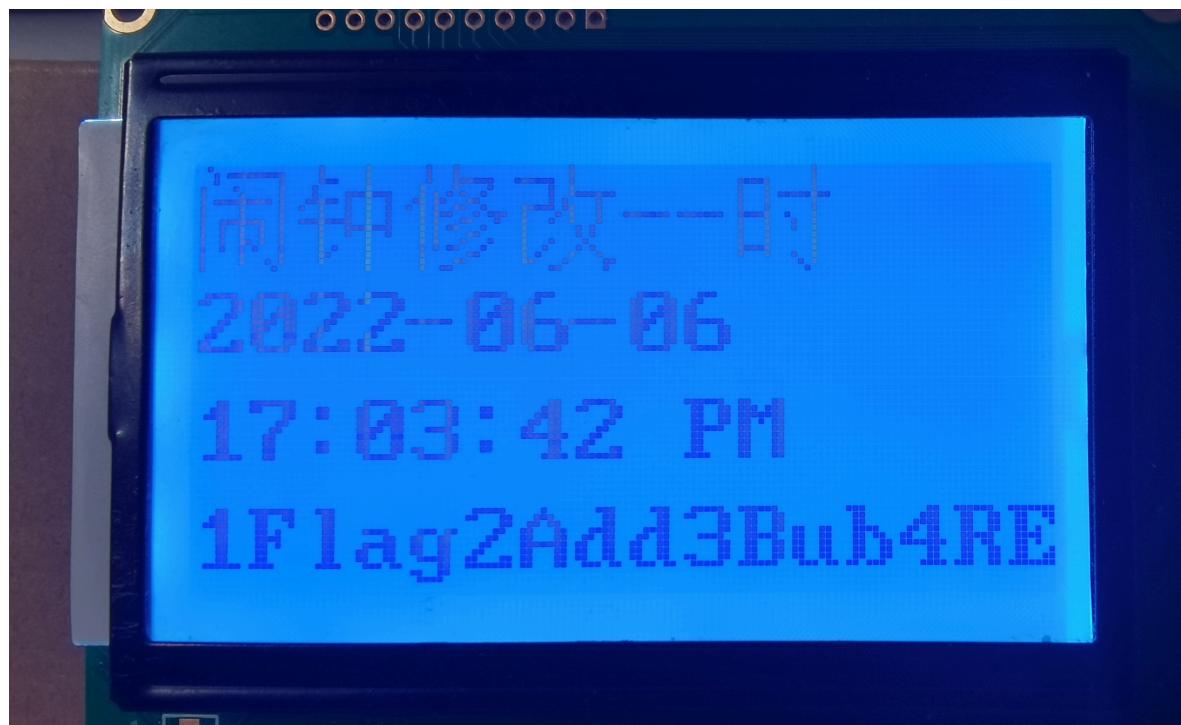


Figure 14: 闹钟时间设置

高级功能：

- 1) 闹钟音乐选择（采用 TF 卡或录音芯片存储音乐数据）

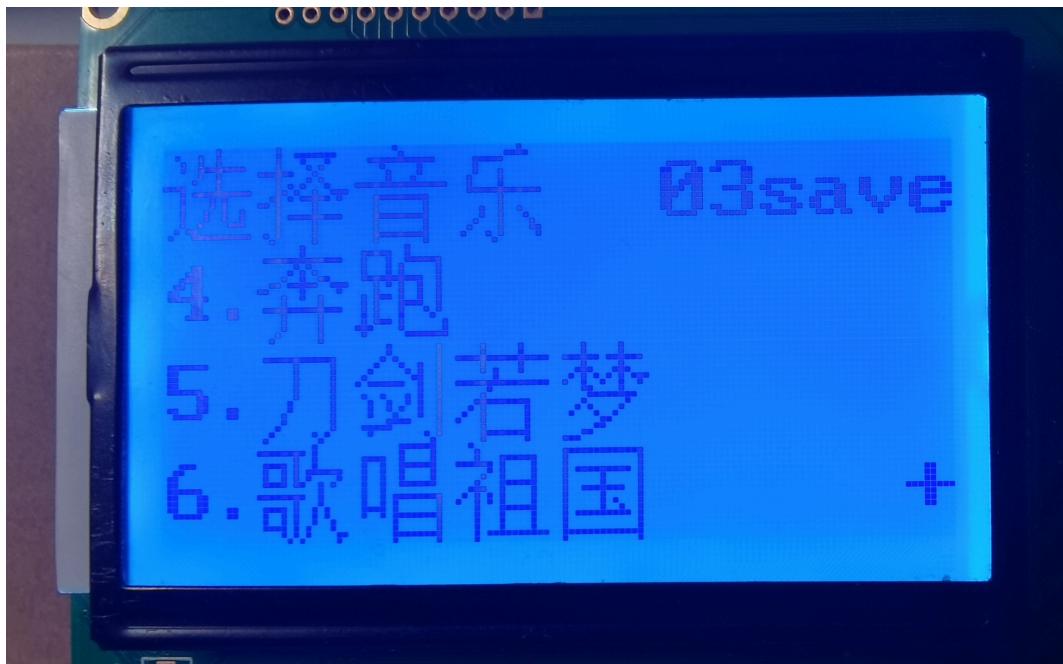


Figure 15: 闹钟音乐选择

- 2) 手机通过蓝牙或 WiFi 网络连接该万年历，通过手机 APP 可以完成基本功能中的所有设置功能

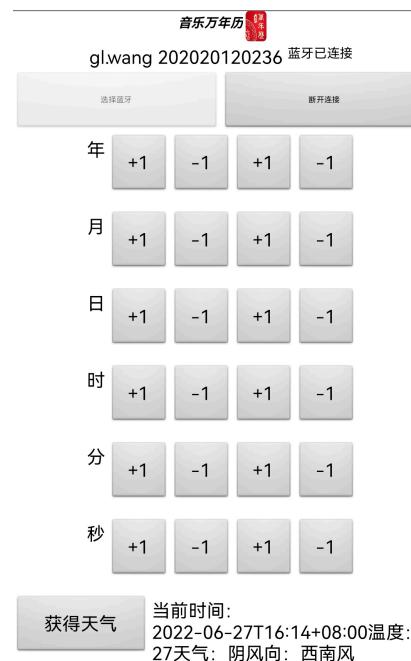


Figure 16: APP 基本功能设置

3) 通过编写 Android 手机程序获得网上天气预报数据，并将该数据通过蓝牙或 WiFi 网络传递给音乐万年历，万年历可以完成天气信息显示

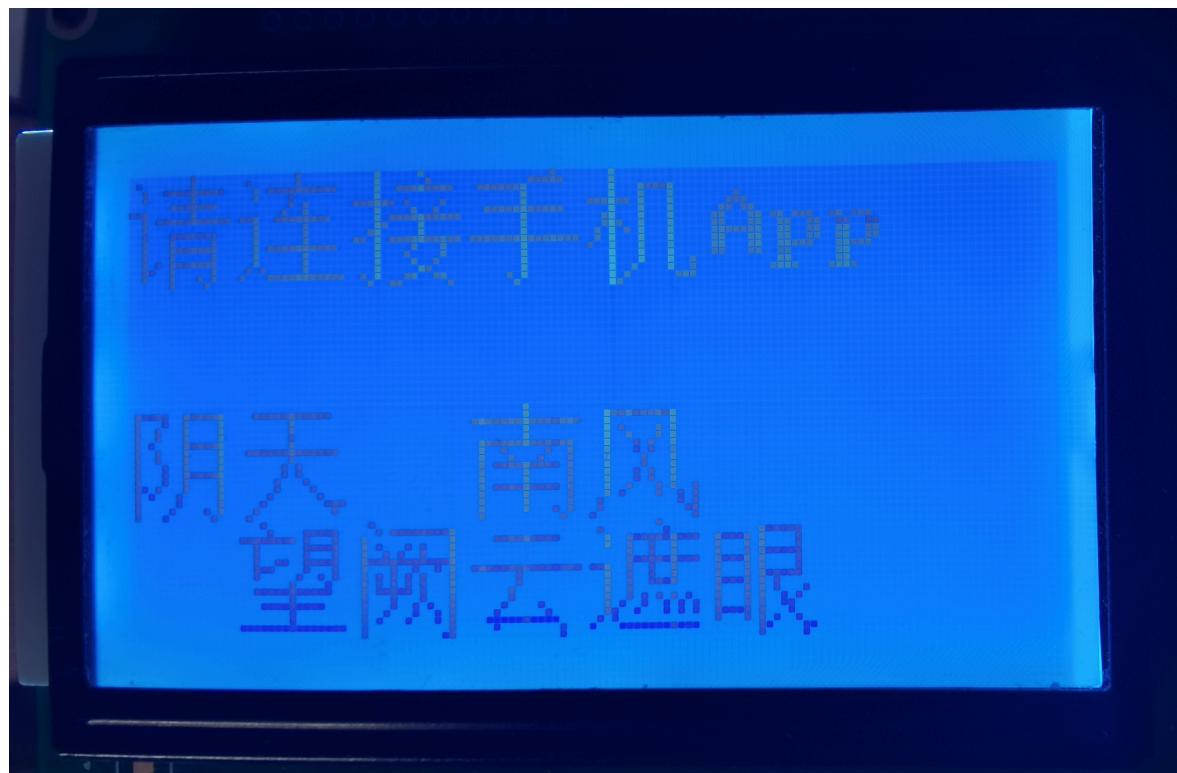


Figure 17: 万年历天气信息显示

4) 火灾检测与报警，报警形势可以是短信。

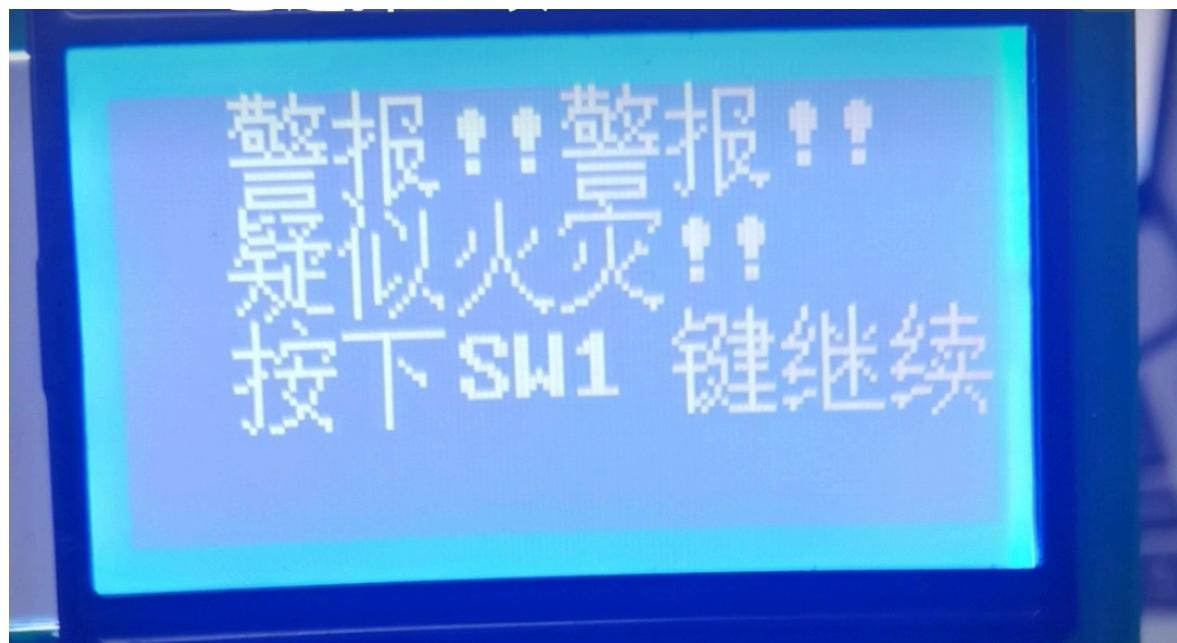


Figure 18: 火灾检测与报警

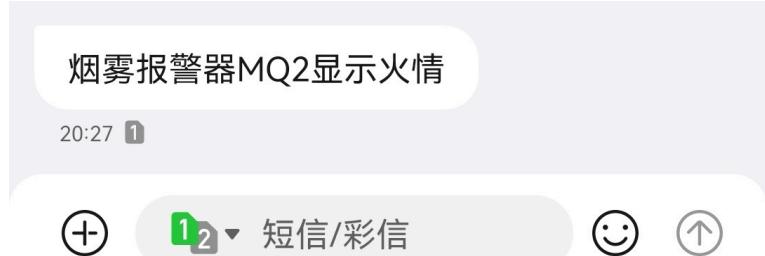


Figure 19: 火灾短信

总结与展望

硬件实现过程不算顺利，四月开始在面包板上实现基础功能，其间显示屏偶尔出现乱码，但是并没有在意；硬件进行到第六步的时候农历年依然没有实现，究其原因是农历年显示经常出现乱码，当时是使用 `strcpy` 和指针数组进行字符创拼接显示，没有及时改换思路，后来临近交期采用 `for` 循环赋值和字符数组进行农历年拼接，大功告成。应该说，实现农历年比我其他部分加起来的时间都要多，如果我在一开始选择比较灵活的 lcd 驱动程序，如果能知晓指针数组和 `string.h` 中 `strcpy` 和 `strcat` 的 bug，应该也是可以很快完成，这是让人很郁闷的事情。

第二个比较麻烦的事情是关于 P0 口作屏幕数据并行口的问题，因为是和 DB 反接，需要对数据进行反置。很多程序都是将无符号整数倒过来，但是在程序中均不成功。我认为不成功的原因是由于指令周期导致的时序错误。后来我在黑 51 论坛找到通过定义 sbit 进行翻转的程序，效果拔群，解决了该问题。

做这个小东西的时候一共熬了这么几次夜，第一次是第二次画板，设计 PCB 到兴头上；第二次是验收前几天，把 App 设计完成。总结收获是积攒了一些 c51 的代码量和经验，对这种小玩具的逻辑有了一些实践。在调试软硬件时，如果你感觉软件和硬件没有问题，那么你可能恰恰犯了轻实践的错误。实际上太多的问题不是我们在设计的时候一锤子、一棒子就能想到的，在实际操作的时候会出现很多实际的 bug，bug 的调试需要很多功夫，怒气值也会上升，但是毕竟代码、原理图和 PCB 要落在实处才有用吧？列宁说“理论是灰色的，生活之树是长青的”，但把理论变成可以应用的实物是需要大量脑力和体力的，说到底这是一个实践的过程，所以我们上的课是“**开放性创新实践**”，很有道理。

附录

Report written by latex, free distribution in www.overleaf.com

参考 main.c 代码

其他所有源码位于 sourceCode 文件夹下。

```
1 # include <reg52.h>
2 # include <string.h>
3 # include "jq8900.h"
4 # include "tts.h"
5 # include "lcd.h"
6 # include "ds1302.h"
7 # include "bluetooth.h"
8 sbit MQ2 = P1 ^ 1;
9 uchar key_scan();
10 void callMenu();
11 void Delay5Ms();
12 //dht11 相关驱动
13 uchar* DHT11_receive();
14 void Delay30us();
15
16 //农历年相关驱动
17 void turn_lunar_calendar(uint year, uchar month, uchar day, uchar *lunar)
18 uchar judgeSolar(uint year, uchar month, uchar day);
19
20 void getweather();
21
22 uchar code TianGan[20] = "甲乙丙丁戊己庚辛壬癸";
23 uchar code DiZhi[24] = "子丑寅卯辰巳午未申酉戌亥";
24 uchar code month_lunar[24] = "正二三四五六七八九十冬腊";
25 uchar code first_day[6] = "初十廿";
26 uchar code second_day[20] = "一二三四五六七八九十";
27 uchar code Solarterms[96] = "小寒大寒立春雨水惊蛰春分清明谷雨立夏小满芒种";
28 void main()
29 {
30     Uart_Init_blue();
31     Lcm_Init();
32     Ds1302Init();
```

```

33     while(1)
34     {
35         uchar key;
36         uchar clockvo[16] = " ";
37         dataProcess();
38         Display_String(1, displayData);
39         Display_String(2, displayData2);
40         Display_String(3, DHT11_receive());
41         Display_String(4, "清风徐来水波不兴");
42
43         key = key_scan();
44         if (key == 1)
45         {
46             callMenu();
47         }
48         else if (key == 2)
49         {
50             SendData(0x0a);
51             SendData(song);
52             SendData(0x0b);
53         }
54         else if (key == 3)
55         {
56             SendData(0x0a);
57             SendData(0x13);
58             SendData(0x0b);
59         }
60         else if (key == 4)
61         {
62
63             Lcm_Init();
64             while(1)
65             {
66                 uchar key;
67                 clockvo[0] = clocktime[2] / 16 + '0';
68                 clockvo[1] = clocktime[2] % 16 + '0';
69                 clockvo[2] = ':';
70                 clockvo[3] = clocktime[1] / 16 + '0';

```

```

71         clockvo[4] = clocktime[1] % 16 + '0';
72         clockvo[5] = ':';
73         clockvo[6] = clocktime[0] / 16 + '0';
74         clockvo[7] = clocktime[0] % 16 + '0';
75         Display_String(1, "查看闹钟时间：");
76         Display_String(2, clockvo);
77         Display_String(3, "应先设置闹钟");
78         Display_String(4, "单击SW4 返回主页");
79         key = key_scan();
80         if (key == 4)
81         {
82             break;
83         }
84     }
85 }
86
87 /////////////////////////////////////////////////////////////////// 闹钟时间检测
88
89 if(alarmDetect())
90 {
91     SendData(0x0a);
92     SendData(song);
93     SendData(0x0b);
94 }
95
96 if(MQ2 == 0)
97 {
98     Delay5Ms();
99     if(MQ2 == 0)
100    {
101        SendData(0x0a);
102        SendData(0x07);
103        SendData(0x0b);
104        //随便发一个就行
105        PostChar(0x01);
106        Lcm_Init();
107        while(1)
108        {

```

```

109         Display_String(1, " 警报!!警报!!    ");
110         Display_String(2, " 疑似火灾!!      ");
111         Display_String(3, " 按下SW1 键继续");
112         key = key_scan();
113         if(key == 1)
114         {
115             SendData(0x0a);
116             SendData(0x13);
117             SendData(0x0b);
118             break;
119         }
120     }
121 }
122 }
123 }
124 }

125 void Basic(void)
126 {
127     uchar key, i;
128     uint year_t;
129     uchar month_t, day_t;
130     uchar TianGan_t, DiZhi_t;
131     uchar lunar[2];
132     uchar lunarinfol[16] = "          ";
133     uchar month[2] = "月";
134     uchar result;
135     Lcm_Init();
136     while(1)
137     {
138         //dataprocess 用于读取TIME和生成displaydata
139         dataProcess();
140         year_t = 2000 + TIME[6] / 16 * 10 + TIME[6] % 16;
141         month_t = TIME[4] / 16 * 10 + TIME[4] % 16;
142         day_t = TIME[3] / 16 * 10 + TIME[3] % 16;
143
144         TianGan_t = (year_t - 3) % 10;
145         DiZhi_t = (year_t - 3) % 12;
146

```

```

147     if(TianGan_t==0)TianGan_t=10;
148     if(DiZhi_t==0)DiZhi_t=12;
149
150     turn_lunar_calendar(year_t, month_t, day_t, lunar);
151     result = judgeSolar(year_t, month_t, day_t);
152     for (i=0; i<2; i++)
153     {
154         lunarinfo[i] = TianGan[2 * (TianGan_t - 1) + i];//天干
155         lunarinfo[i + 2] = DiZhi[2 * (DiZhi_t - 1) + i];//地支
156         lunarinfo[i + 4] = month_lunar[2 * (lunar[0] - 1) + i];//月数
157         lunarinfo[i + 6] = month[i];//“月”
158         if(lunar[1] <= 10)
159         {
160             lunarinfo[i + 8] = first_day[i];//初十廿
161             lunarinfo[i + 10] = second_day[2 * (lunar[1] - 1) + i];//1-10
162         }
163         else if(lunar[1] <= 19)
164         {
165             lunarinfo[i + 8] = first_day[i + 2];//初十廿
166             lunarinfo[i + 10] = second_day[2 * (lunar[1] - 10 - 1) + i];//1-10
167         }
168         else if(lunar[1] <= 29)
169         {
170             lunarinfo[i + 8] = first_day[i + 4];//初十廿
171             lunarinfo[i + 10] = second_day[2 * (lunar[1] - 20 - 1) + i];//1-10
172         }
173     }
174     if(result)
175     {
176         lunarinfo[12] = Solarterms[4 * result];
177         lunarinfo[13] = Solarterms[4 * result + 1];
178         lunarinfo[14] = Solarterms[4 * result + 2];
179         lunarinfo[15] = Solarterms[4 * result + 3];
180     }
181     else
182     {
183         lunarinfo[12] = ' ';
184         lunarinfo[13] = ' ';

```

```

185         lunarinfo[14] = ' ';
186         lunarinfo[15] = ' ';
187     }
188     Display_String(1, displayData);
189     Display_String(2, displayData2);
190     Display_String(3, lunarinfo);
191     Display_String(4, "清风徐来水波不兴");
192
193     key = key_scan();
194     if(key == 4)
195     {
196         return;
197     }
198 }
199
200 }
201 void callMenu()
202 {
203     uchar flag = 1;
204     uchar test = 0x01;
205     uchar key0;
206     Lcm_Init();
207     while(1)
208     {
209         Delay30us();
210         key0 = key_scan();
211         if (key0 == 1)
212         {
213             if (flag == 1)
214             {
215             }
216             else if (flag == 2)
217             {
218                 Basic();
219             }
220             else if (flag == 3)
221             {
222                 dataChange();

```

```

223     }
224     else if (flag == 4)
225     {
226         setClock();
227     }
228     else if (flag == 5)
229     {
230         Musicselect();
231     }
232     else if (flag == 6)
233     {
234         getweather();
235     }
236     else if (flag == 7)
237     {
238         return;
239     }
240     else
241     {
242     }
243 }
244 else if (key0 == 2)
245 {
246     flag++;
247     if (flag == 8) flag = 1;
248 }
249 else if (key0 == 3)
250 {
251     break;
252 }
253 if (flag == 1)
254 {
255     Display_String(1, "SW2 选中 SW1 进入");
256     Display_String(2, "1. 基础显示      ");
257     Display_String(3, "2. 修改时间      ");
258     Display_String(4, "3. 设置闹钟      ");
259 }
260 else if (flag == 2)

```

```
261     {
262         Display_String(1, "SW2 选中 SW1 进入");
263         Display_String(2, "1. 基础显示      选");
264         Display_String(3, "2. 修改时间      ");
265         Display_String(4, "3. 设置闹钟      ");
266     }
267     else if (flag == 3)
268     {
269         Display_String(1, "SW2 选中 SW1 进入");
270         Display_String(2, "1. 基础显示      ");
271         Display_String(3, "2. 修改时间      选");
272         Display_String(4, "3. 设置闹钟      ");
273     }
274     else if (flag == 4)
275     {
276         Display_String(1, "SW2 选中 SW1 进入");
277         Display_String(2, "1. 基础显示      ");
278         Display_String(3, "2. 修改时间      ");
279         Display_String(4, "3. 设置闹钟      选");
280     }
281     else if (flag == 5)
282     {
283         Display_String(1, "1. 基础显示      ");
284         Display_String(2, "2. 修改时间      ");
285         Display_String(3, "3. 设置闹钟      ");
286         Display_String(4, "4. 设置铃声      选");
287     }
288     else if (flag == 6)
289     {
290         Display_String(1, "2. 修改时间      ");
291         Display_String(2, "3. 设置闹钟      ");
292         Display_String(3, "4. 设置铃声      ");
293         Display_String(4, "5. 获取天气      选");
294     }
295     else if (flag == 7)
296     {
297         Display_String(1, "3. 设置闹钟      ");
298         Display_String(2, "4. 设置铃声      ");
```

```
299     Display_String(3, "5. 获取天气      ");
300     Display_String(4, "6. 返回      选");
301 }
302 }
303 }
304 void getweather()
305 {
306     uchar key;
307     Lcm_Init();
308     while(1)
309     {
310         Display_String(1, "请连接手机App      ");
311         if(weather == 25)
312         {
313             Display_String(3, "晴朗天气      ");
314             Display_String(4, "即墨的天是晴朗天");
315         }
316         else if(weather == 26)
317         {
318             Display_String(3, "多云天气      ");
319             Display_String(4, "总为浮云能蔽日      ");
320         }
321         else if(weather == 27)
322         {
323             Display_String(3, "少云天气      ");
324         }
325         else if(weather == 28)
326         {
327             Display_String(3, "晴间多云      ");
328         }
329         else if(weather == 29)
330         {
331             Display_String(3, "阴天 南风      ");
332         }
333         else if(weather == 30)
334         {
335             Display_String(3, "阵雨      ");
336         }
```

```
337     else if(weather == 31)
338     {
339         Display_String(3, "强阵雨");
340     }
341     else if(weather == 32)
342     {
343         Display_String(3, "雷阵雨");
344     }
345     else if(weather == 33)
346     {
347         Display_String(3, "强雷阵雨");
348     }
349     else if(weather == 34)
350     {
351         Display_String(3, "小雨");
352         Display_String(4, "朝来寒雨晚来风");
353     }
354     else if(weather == 35)
355     {
356         Display_String(3, "中雨");
357         Display_String(4, "朝来寒雨晚来风");
358     }
359     else if(weather == 36)
360     {
361         Display_String(3, "大雨");
362         Display_String(4, "朝来寒雨晚来风");
363     }
364     else if(weather == 37)
365     {
366         Display_String(3, "暴雨");
367         Display_String(4, "暴雨倾盆水满溪");
368     }
369     else if(weather == 38)
370     {
371         Display_String(3, "大暴雨");
372         Display_String(4, "暴雨倾盆水满溪");
373     }
374     else if(weather == 39)
```

```
375     {
376         Display_String(3, "    雨          ");
377         Display_String(4, " 朝来寒雨晚来风  ");
378     }
379     else
380     {
381     }
382     key = key_scan();
383     if (key == 4)
384     {
385         return;
386     }
387 }
388 }
```