



山东大学

崇新学堂

2023—2024 学年第二学期

电子音乐万年历报告

课 程 名 称： 开放性创新实践 I —硬件
课 程 部 分： 硬件报告

专 业 班 级： 崇新 22.1
学 生 姓 名： 庞开升
学 生 学 号：
报 告 时 间： 2024. 7. 3

目录

| | |
|--|-----------|
| 1 项目功能要求 | 3 |
| 1.1 基本功能 | 3 |
| 1.2 高级功能 | 3 |
| 2 实现步骤 | 3 |
| 3 原理图绘制 | 4 |
| 3.1 设计 PCB 板 | 4 |
| 3.1.1 第一版 PCB | 4 |
| 3.1.2 第二版 PCB | 6 |
| 3.2 焊接调试 | 6 |
| 3.2.1 第一版焊接 | 6 |
| 3.2.2 第二版焊接 | 7 |
| 3.2.3 第一、二次 PCB 比较 | 7 |
| 4 3D 外壳制作 | 8 |
| 5 功能完成情况 | 9 |
| 5.1 年、月、日、时、分、秒的显示与设置（基本功能 1） | 9 |
| 5.2 年、月、日、时、分、秒的显示与设置（基本功能 2） | 10 |
| 5.3 农历年月日、二十四节气、星期显示（基本功能 1、3） | 10 |
| 5.4 闹钟时间设置，闹钟时间到进行声光报警提示（基本功能 4） | 11 |
| 5.5 温湿度检测与显示（基本功能 5） | 12 |
| 5.6 闹钟音乐选择（高级功能 1） | 12 |
| 5.7 手机 APP 可以完成基本功能中的所有设置功能 | 13 |
| 5.8 事件提示 | 13 |
| 5.9 通过蓝牙完成天气信息显示 | 14 |
| 5.10 火灾检测与声光报警，报警形势可以是短信 | 15 |
| 6 代码 | 16 |
| 6.1 Main.c | 16 |
| 6.2 12864.c | 16 |

| | |
|----------------------------------|-----------|
| 6.3 Ds1302.c | 16 |
| 6.4 Dht11.c | 17 |
| 6.5 jq8900.c | 17 |
| 6.6 keyscan.c | 17 |
| 6.7 Lunar.c | 18 |
| 6.8 Bluetooth.c | 18 |
| 7 软件设计 | 19 |
| 7.1 蓝牙连接程序逻辑 | 19 |
| 7.1.1 Screen1 初始化 | 19 |
| 7.1.2 列表选择框：选择蓝牙准备选择 | 20 |
| 7.1.3 列表选择框：选择蓝牙选择完成 | 20 |
| 7.1.4 按钮：断开连接被点击 | 20 |
| 7.2 时间、闹钟设置蓝牙发送阵列 | 20 |
| 7.3 事项提醒逻辑 | 20 |
| 7.4 天气逻辑 | 21 |
| 7.4.1 初始化变量 | 21 |
| 7.4.2 按钮：获取天气 -> 被点击 | 21 |
| 7.4.3 Web 客户端 1 -> 获取文本 | 21 |
| 7.4.4 调用 蓝牙客户端 1 . 发送单字节数字 | 21 |
| 8 总结与心得 | 22 |

1 项目功能要求

1.1 基本功能

- 年、月、日、时、分、秒、星期的显示与设置
- AM、PM 设置与显示
- 农历年月日、二十四节气显示
- 闹钟时间设置，闹钟时间到进行声光报警提示
- 温湿度检测与显示

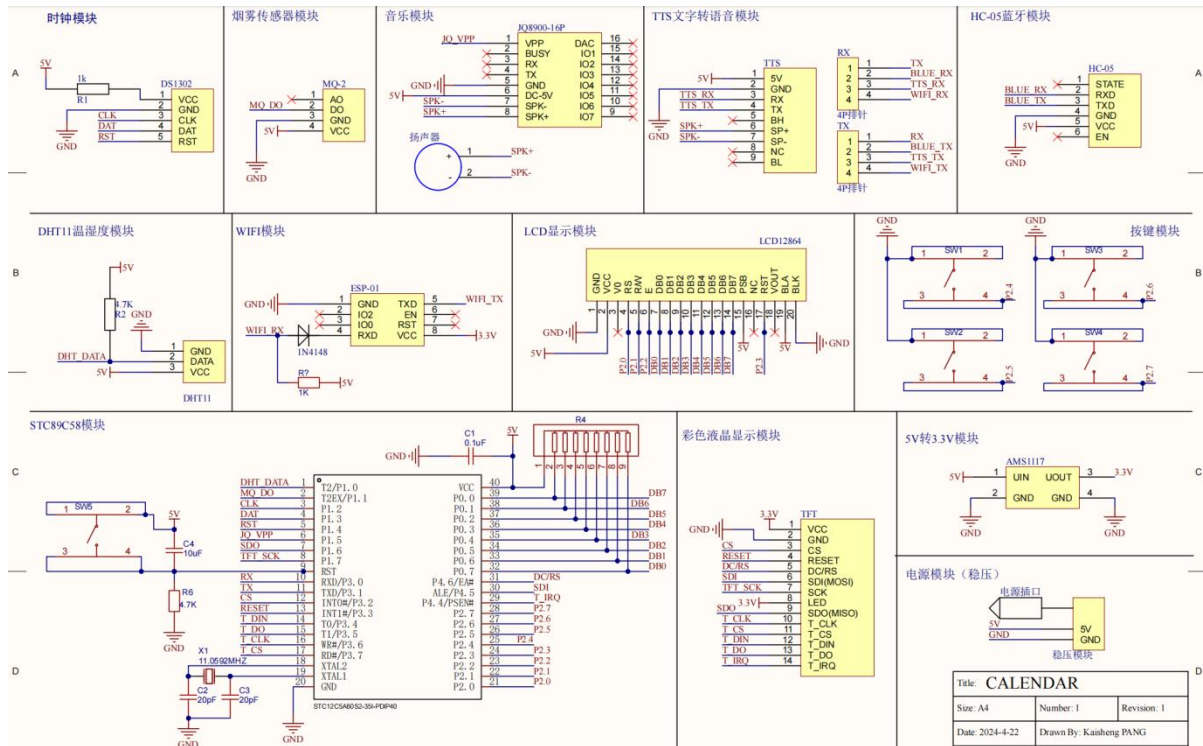
1.2 高级功能

- 闹钟音乐选择（采用 TF 卡或录音芯片存储音乐数据）
- 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
- 事件提示设置，可以通过手机或电脑将重要事件信息发送给系统，系统在事件发生前一日通过字幕提醒（比如某某人的生日，某某会议等）
- 通过编写 Android 手机程序或者将万年历连接到电脑上，获得网上天气预报数据，并将该数据通过蓝牙或 WiFi 网络传递给万年历，万年历可以完成天气信息显示
- 火灾检测与声光报警，报警形势可以是短信。

2 实现步骤

- 利用 Altium Designer 设计该系统的原理图
- 利用 Altium Designer 设计该系统的 PCB 版图，并委托制板厂家加工，板上要有学号姓名文字标识
- 焊接元器件进行实物调试
- 进行 3D 打印，配套机壳，做成能实用的万年历！
- 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

3 原理图绘制



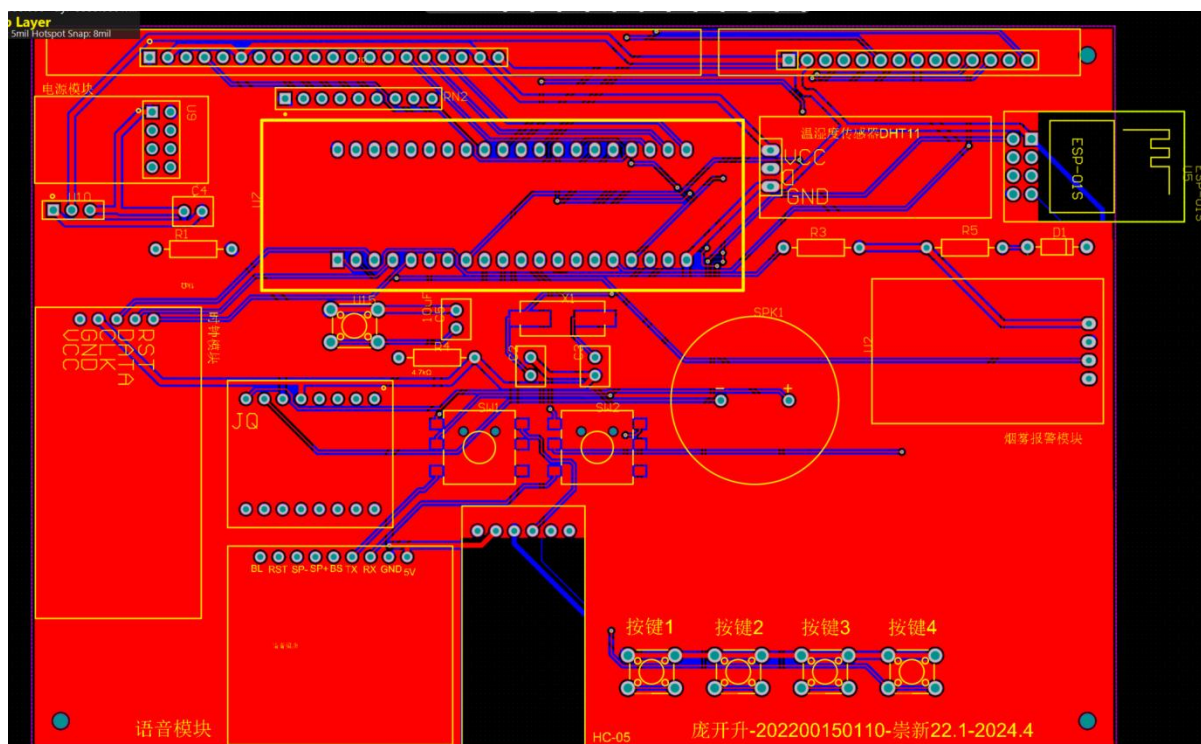
根据设计要求，设计了第一版原理图，根据老师提示，在 WIFI_RX 处添加电平转换设计。

| 模块名称 | 具体选择 |
|-----------|--------------|
| MCU | STC12C5A60S2 |
| 时钟 | DS1302 |
| 烟雾传感 | MQ-2 |
| 音乐 | JQ8900-16P |
| 文字转语音 | TTS |
| 蓝牙 | HC-05 |
| 温湿度传感 | DHT11 |
| WIFI | ESP-01 |
| LCD 显示 | LCD12864 |
| 彩色液晶显示 | TFT |
| 5V 转 3.3V | AMS1117 |

3.1 设计 PCB 板

3.1.1 第一版 PCB

根据原理图设计 PCB，因为显示屏占据空间较大，选用 15*10 尺寸的 PCB 板，供电和接地线宽为 30mil，其余线宽为 10mil。这是第一版的 PCB，但经过测试，发现存在很多问题。

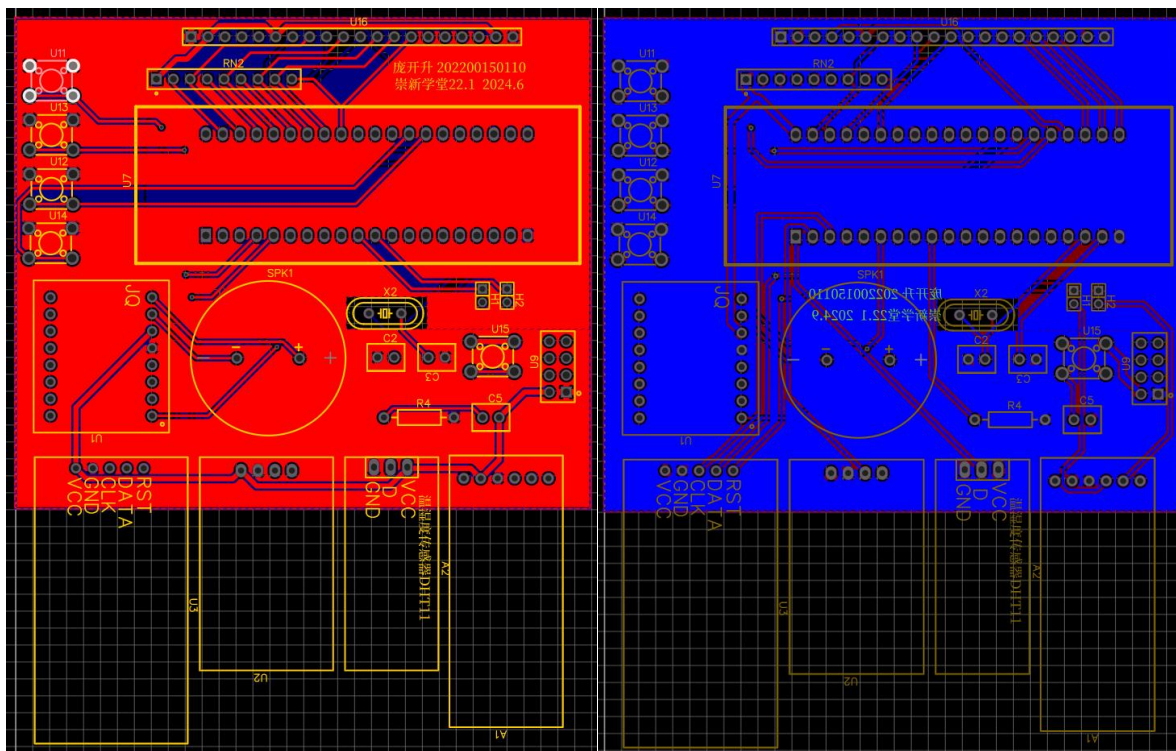


问题汇总及修正解决:

| 序号 | 问题描述 | 解决方案 |
|----|---|---------------------|
| 1 | PCB 设计的模块方向和大小造成空间浪费 | 改进 PCB 布局与设计 |
| 2 | 同时使用 ESP-01S WIFI 和 HC-05 冲突且意义不大 | 去掉 ESP-01S |
| 3 | TX、RX 分别要连接三个模块，所以使用了一款五个方向的开关，可是这款开关会自动复位！ | 换成排针形式，将 TX、RX 各自引出 |
| 4 | TFT 显示内容太占据内存，C58 不能胜任 | 去掉 TFT 显示屏 |
| 5 | DC 供电电压偏差太大 | 加入稳压模块 |

3.1.2 第二版 PCB

解决了第一版存在的所有问题，并且将板大小从 **15*10 减小到了 8.6*7.4** 的大小，**免费打板且精巧。特色：考虑了 3D 打印的按钮问题，巧妙地从背面焊接按钮，搭配 3D 打印，效果良好！**



3.2 焊接调试

3.2.1 第一版焊接

焊接 PCB，并进行连通性测试，但是发现 LCD12864 的显示不稳定，PCB 接触不良，在进行通讯时极其复杂。



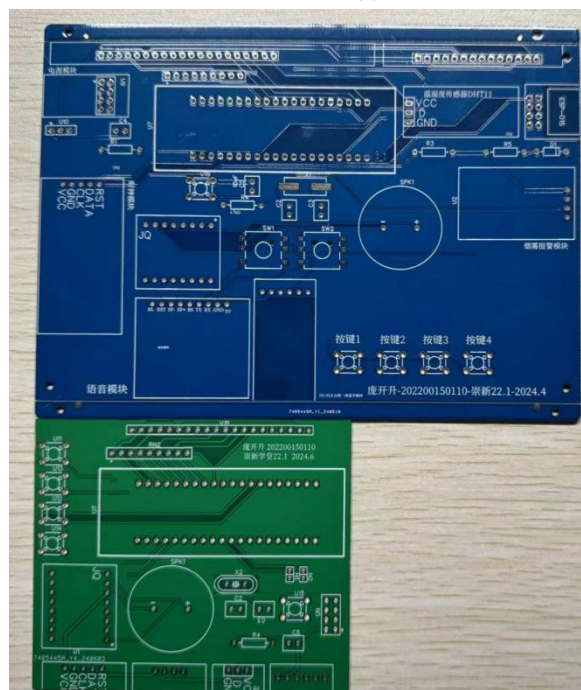
3.2.2 第二版焊接

这次焊接效果良好，大大节约了空间，充分利用板外空间，蓝牙通讯也恢复正常。



3.2.3 第一、二次 PCB 比较

很直观地看到了 PCB 制版的差别，提升显著。



4 3D 外壳制作

使用 SolidWorks 建模，制作了一个外壳以及一个后盖，将显示屏、按钮、烟雾传感器进行挖孔，将器件从里面穿出，便于显示、操控与检测。

步骤如下：

1.创建基本外壳

- 在前视平面上绘制一个矩形。
- 使用拉伸凸台/基体功能将矩形拉伸成一个长方体，这将成为外壳的基本形状。

2. 添加开口和孔

- 在合适的平面上绘制所需的开口和孔的轮廓。
- 使用切除拉伸功能来切除这些开口和孔。可以根据实际需要调整开口和孔的位置和大小。在前视平面上绘制一个矩形，用于显示屏开口，然后使用切除功能切出这个开口。
- 在侧视平面上绘制两个小矩形，用于按键开口，然后使用切除功能切出这些开口。

3. 添加文字

- 在前视平面上使用文本工具添加所需的文字。
- 调整文字的位置和大小，然后使用切除拉伸功能将文字凹陷进去。

4. 调整厚度

- 使用壳工具将外壳的壁厚调整为所需的厚度。选择壳工具，然后选择要保持厚度的面，输入所需的厚度值。

5. 添加圆角

- 使用圆角工具在外壳的边缘添加圆角，增加外壳的美观性和结构强度。

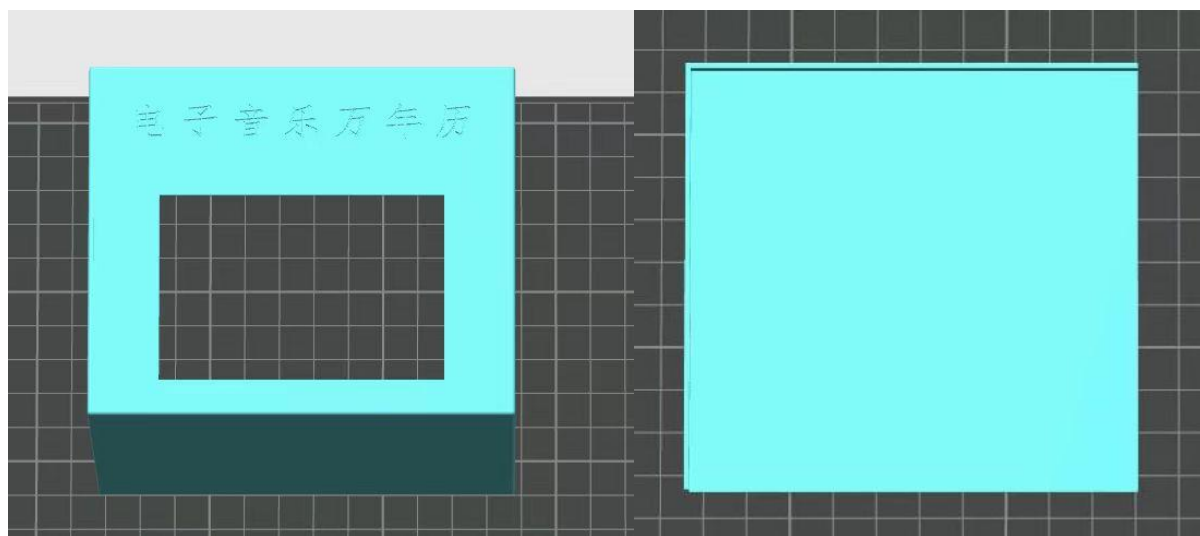
6. 检查和调整

- 旋转和查看 3D 模型，确保所有开口、孔和文字都在正确的位置。
- 根据需要进行调整，以确保外壳符合设计要求。

7. 保存文件

- 完成设计后，保存 SolidWorks 文件（.sldprt），并根据需要导出为 STL 用于 3D 打印或其他制造工艺。

建模图：



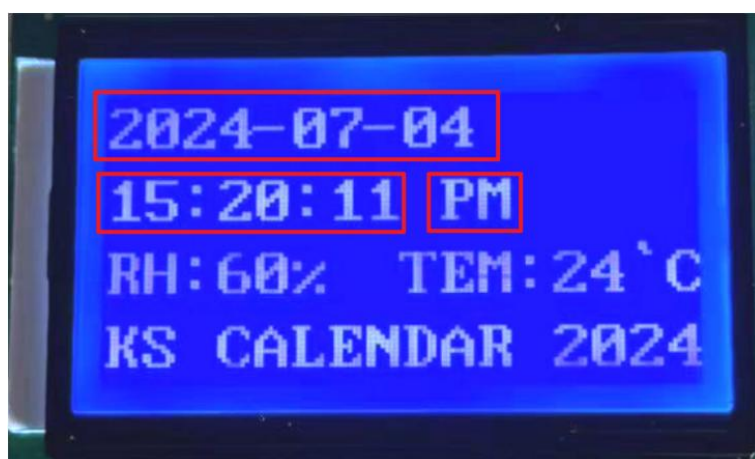
效果图：

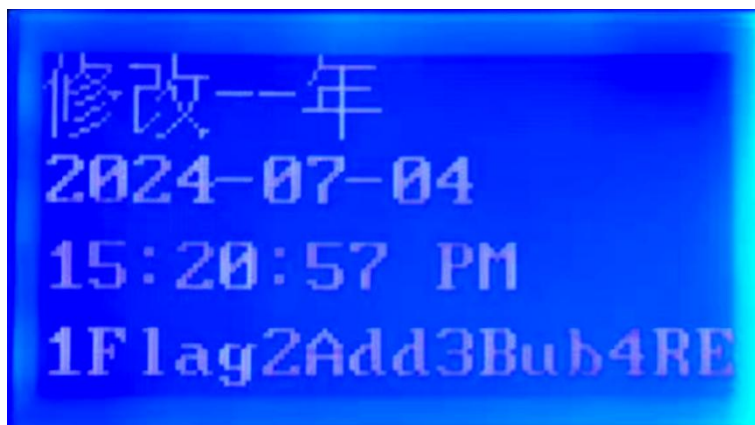


5 功能完成情况

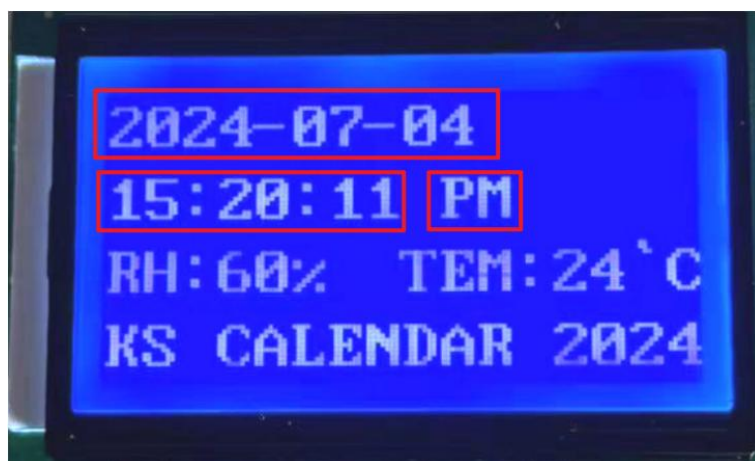
所有功能全部完成，共十项，基本功能高级功能各五项。

5.1 年、月、日、时、分、秒的显示与设置（基本功能 1）

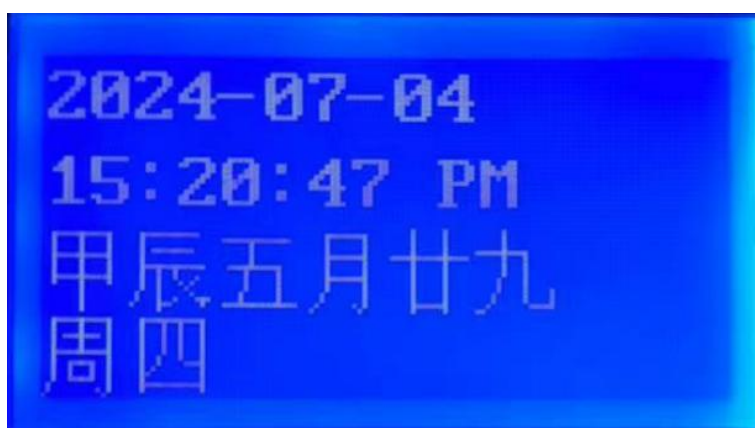




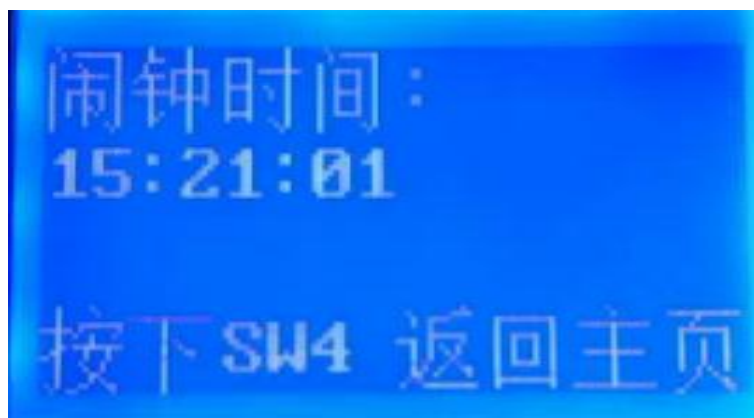
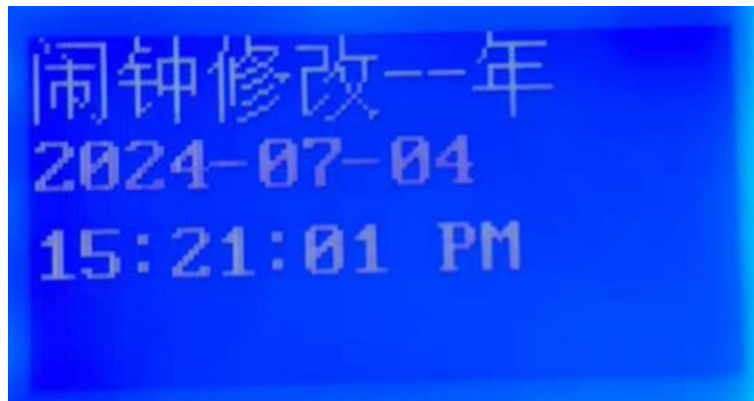
5.2 年、月、日、时、分、秒的显示与设置（基本功能 2）



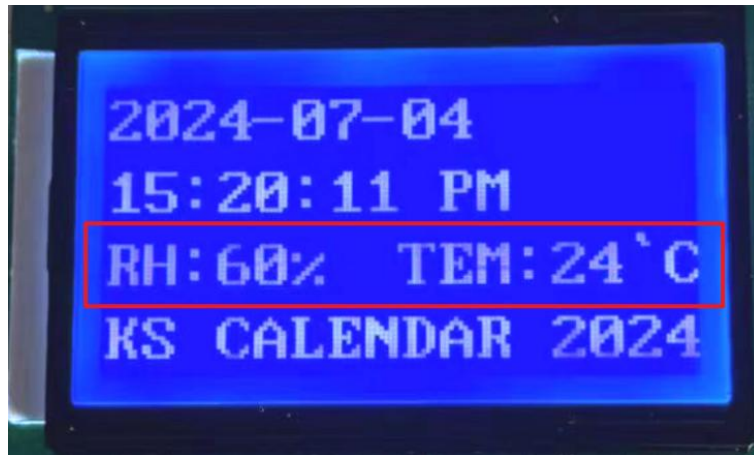
5.3 农历年月日、二十四节气、星期显示（基本功能 1、3）



5.4 闹钟时间设置，闹钟时间到进行声光报警提示（基本功能 4）



5.5 温湿度检测与显示（基本功能 5）



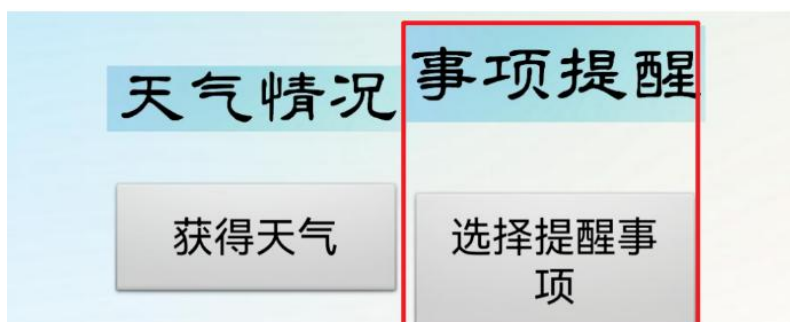
5.6 闹钟音乐选择（高级功能 1）

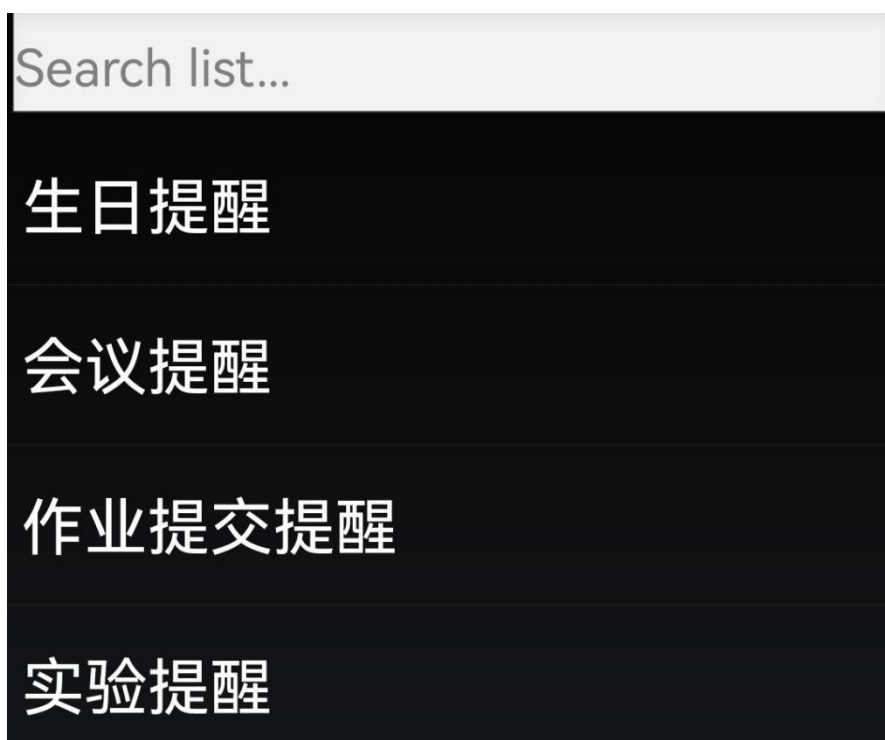


5.7 手机 APP 可以完成基本功能中的所有设置功能

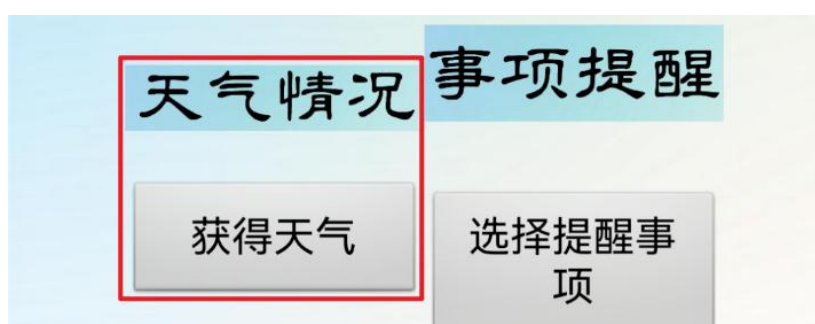


5.8 事件提示

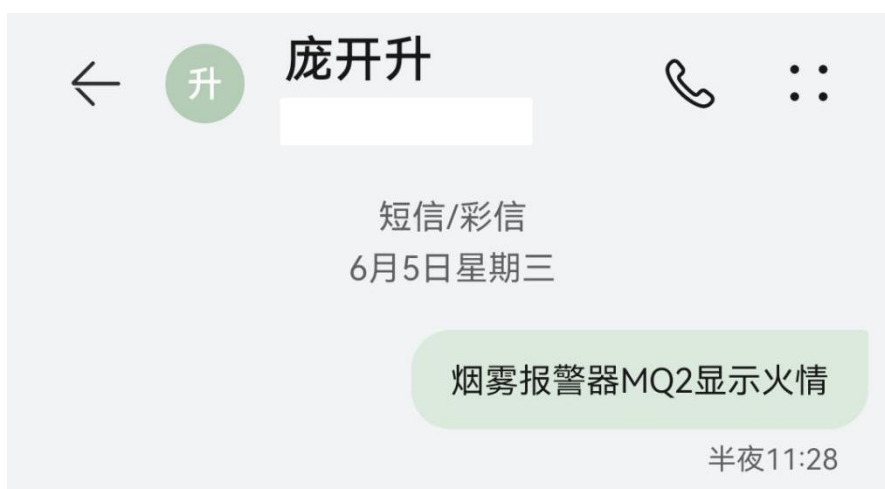




5.9 通过蓝牙完成天气信息显示



5.10 火灾检测与声光报警，报警形势可以是短信



6 代码

6.1 Main.c

| 函数名 | 功能介绍 |
|-------------------------|---|
| main | 初始化蓝牙、屏幕和时钟；进入一个无限循环，显示时间、DHT11 传感器数据、以及按键检测和处理；处理闹钟提醒和火灾警报检测 |
| Basic | 显示基本的日期和农历信息；通过按键返回主菜单 |
| calculateWeekDay | 计算给定日期的星期几 |
| callMenu | 显示并处理主菜单，用户可以选择进入不同的功能模块 |
| getweather | 获取并显示天气信息 |
| control | 显示提醒事项，如生日、会议等 |

6.2 12864.c

| 函数名 | 功能介绍 |
|--|---|
| Switch_bit(uchar InChar) | 将输入的字符 InChar 进行位反转，具体是将字符的各个位交换顺序 |
| Display_String(uchar line, uchar *string) | 在指定的行 line 显示一个 16 位的字符串 string |
| Wr_Data(uchar wrdata) | 向 LCD 写入数据，使用 Switch_bit 函数进行数据位转换，然后将数据发送到 LCD |
| Wr_Command(uchar wrcommand, busyc) | 向 LCD 发送命令，如果 busyc 为真，则在发送命令前进行忙等待 |
| RDbf(void) | 忙等待函数，检查 LCD 是否忙碌，如果忙则等待 |
| Lcm_Init(void) | 初始化 LCD，包括复位和发送初始命令 |
| Delay5Ms() | 延时 5 毫秒，用于 LCD 初始化和其他需要延时的地方 |

6.3 Ds1302.c

| 函数名 | 功能介绍 |
|---|---|
| Ds1302Write(uchar addr, uchar dat) | 向 DS1302 写入命令（地址和数据），通过串行通信将数据发送到指定的地址 |
| Ds1302Read(uchar addr) | 从 DS1302 读取指定地址的数据，通过串行通信接收数据 |
| Ds1302Init() | 初始化 DS1302 时钟芯片，禁用写保护，设置初始时间，然后重新启用写保护。 |
| Ds1302ReadTime() | 读取 DS1302 的当前时间信息（秒、分、时、日、月、年），并存储在 TIME 数组中。 |
| dataProcess() | 将 DS1302 读取的时间数据转化为可显示在 12864 LCD 上的字符串格式 |
| dataProcess_new() | 类似于 dataProcess()，但使用 buffer 数组处理时间数据 |

| | |
|--|---|
| dataChange() | 修改 DS1302 中的时间，通过用户输入调整年、月、日、时、分、秒，然后将修改后的时间写回 DS1302 |
| setClock() | 设置闹钟时间，通过用户输入调整年、月、日、时、分、秒，更新 clocktime 数组。 |
| alarmDetect() | 检测当前时间是否与设定的闹钟时间一致，如果一致则返回 1，否则返回 0。 |
| blue2time(uchar flag, uchar pos) | 通过蓝牙修改 DS1302 的时间，根据 flag 和 pos 调整相应位置的时间值。 |
| blue2clocktime(uchar flag, uchar pos) | 通过蓝牙修改闹钟时间，根据 flag 和 pos 调整 clocktime 数组中的时间值。 |

6.4 Dht11.c

| 函数名 | 功能介绍 |
|--------------------------------|--|
| DHT11_delay_us(uchar n) | 微秒级延时函数，延时 n 微秒。 |
| DHT11_delay_ms(uint z) | 毫秒级延时函数，延时 z 毫秒。 |
| Delay30us() | 延时 30 微秒的函数，使用 11.0592MHz 的时钟频率。 |
| DHT11_start() | 初始化 DHT11 传感器，发送启动信号以开始温湿度数据的传输。 |
| DHT11_rec_byte() | 接收 DHT11 传感器发送的一个字节数据，通过轮询数据线的电平变化来读取每一位。 |
| DHT11_receive() | 接收 DHT11 传感器的数据，并将其格式化为可显示在 LCD12864 上的 16 位字符串格式。读取湿度和温度数据，并检查数据校验和是否正确。格式化湿度和温度数据为字符串，"RH % TEM `C"。 |

6.5 jq8900.c

| 函数名 | 功能介绍 |
|---|--|
| Delay1500us() Delay500us() Delay2ms() Delay4ms() | 这些是延时子函数，用于产生不同的延时，用于控制通信时序 |
| SendData(uint addr) | 向 JQ8900 模块发送指令，通过一线通信协议发送数据；常见指令包括重复播放（0x11）、结束播放（0x13）、清除数字（0x0a）和选曲确认（0x0b）。 |
| Musicselect() | 音乐选择子函数，用户可以通过按键选择不同的歌曲；共有六首歌曲存储在 JQ8900 的 SD 卡中，通过全局变量 song 存储选择的歌曲序号，并通过 SendData(song) 进行播放 |

6.6 keyscan.c

| 函数名 | 功能介绍 |
|-------------------|--|
| key_scan() | 按键检测函数，仅支持单点检测，不支持连点和多点检测。返回键值 1-4。检测 P2 端口的 4 个引脚状态，确定哪个按键被按下。如果有按键按下，进行去抖动处理，并返回对应的键值。 |

6.7 Lunar.c

| 函数名 | 功能介绍 |
|---|--|
| LunarCalendar(int year, int month, int day) | 计算阳历日期对应的农历日期; 根据给定的阳历年份、月份和日期, 计算出农历的年份、月份和日期, 并将结果存储在 LunarCalendarDay 中。 |
| lunar_calendar(int *solar, uchar *lunar) | 接受阳历日期数组 solar 和农历结果数组 lunar; 调用 LunarCalendar 函数, 转换阳历日期为农历日期, 并将结果存储在 lunar 中。 |
| turn_lunar_calendar(uint year, uchar month, uchar day, uchar *lunar) | 转换阳历日期为农历日期; 接受阳历年份、月份和日期, 并将结果存储在 lunar 数组中。 |
| judgeSolar(uint year, uchar month, uchar day) | 判断给定日期是否为节气日; 根据节气表 jieqi 和输入的阳历日期, 返回对应的节气编号。 |

6.8 Bluetooth.c

| 函数名 | 功能介绍 |
|----------------------------------|---|
| Uart_Isr() | 串口中断服务程序, 通过中断接收数据并进行处理; 根据接收到的数据调用 blue2time() 和 blue2clocktime() 函数来调整时间和闹钟时间; 如果接收到的值大于等于 25, 则更新天气信息。 |
| Uart_Init_blue() | 串口初始化函数, 设置串口通信的参数, 如工作模式、波特率等; 启动定时器 T1 用于生成波特率; 使能全局中断和串口中断。 |
| PostChar(uchar character) | 通过串口发送一个字符, 用于报警时发送短消息。 |

7 软件设计

使用 MIT Appinventor 进行开发，UI 界面如下，通过上方进行蓝牙连接或者断开，左侧进行日期的控制，右侧进行闹钟的控制，最下面进行天气获取以及事项提醒，点击即可将信息传入单片机中。



7.1 蓝牙连接程序逻辑

7.1.1 Screen1 初始化

- 调用 蓝牙客户端 1. 断开连接：在应用启动时，确保任何可能的蓝牙连接都被断开。
- 设置 列表选择框：选择蓝牙 . 启用 为 真：启用蓝牙选择框。
- 设置 标签连接情况 . 文本 为 蓝牙未连接，单击蓝牙连接选择：更新界面标签，提示用户蓝牙未连接，需要选择连接。

7.1.2 列表选择框：选择蓝牙准备选择

- 调用 蓝牙客户端 1. 获得设备 . 设备及名称：获取可用的蓝牙设备和名称。
- 设置 列表选择框：选择蓝牙 . 元素 为 蓝牙客户端 1. 设备及名称：将获取的蓝牙设备列表显示在选择框中。
- 设置 标签连接情况 . 文本 为 准备连接：更新标签提示用户准备连接。

7.1.3 列表选择框：选择蓝牙选择完成

- 调用 蓝牙客户端 1. 连接 地址 为 列表选择框：选择蓝牙 . 选中项：尝试连接用户选择的蓝牙设备。
- 如果 蓝牙客户端 1. 连接状态：
 - 否则：如果连接失败，更新标签提示蓝牙未连接。
 - 如果 蓝牙客户端 1. 连接状态：再次检查连接状态，如果成功：
 - 设置 标签连接情况 . 文本 为 蓝牙已连接：更新标签提示蓝牙已连接。
 - 设置 定时器 1. 已启用 为 真：启用定时器，用于周期性检查状态。
 - 设置 标签 MQ2 . 文本 为 烟雾报警器 MQ2 显示火情：更新标签显示 MQ2 传感器的火情状态。

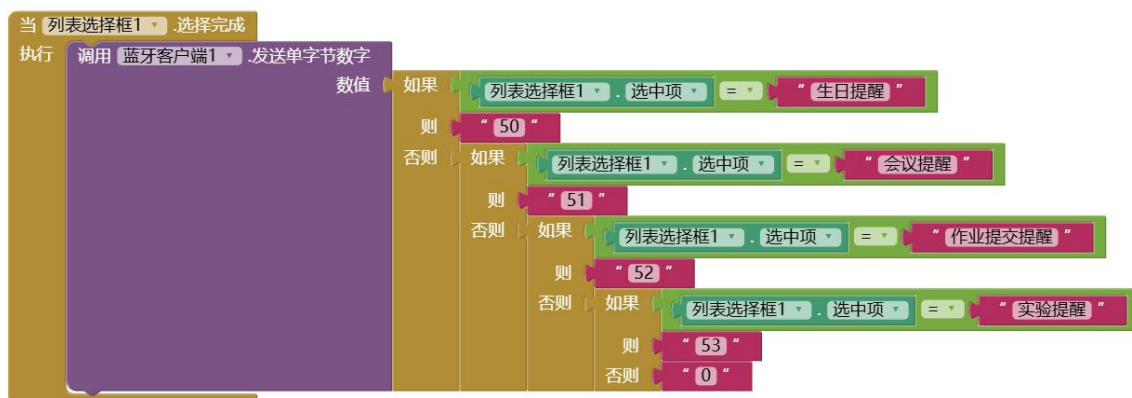
7.1.4 按钮：断开连接被点击

- 调用 蓝牙客户端 1. 断开连接：在用户点击断开连接按钮时，断开蓝牙连接。
- 设置 列表选择框：选择蓝牙 . 启用 为 真：重新启用蓝牙选择框。
- 设置 标签连接情况 . 文本 为 蓝牙未连接，单击蓝牙连接选择：更新标签提示蓝牙未连接，用户需要重新选择连接。

7.2 时间、闹钟设置蓝牙发送阵列

通过对按钮点击设置蓝牙发送单字节数字，到单片机中解码，对应每一个按钮对应的操作即可

7.3 事项提醒逻辑



列表选择框 1：选择完成

调用 蓝牙客户端 1. 发送单字节数字：发送一个单字节数字给蓝牙客户端。

数值：

如果 列表选择框 1. 选中项 = "生日提醒"

发送数值 50

否则如果 列表选择框 1. 选中项 = "会议提醒"

发送数值 51

否则如果 列表选择框 1. 选中项 = "作业提交提醒"

发送数值 52

否则如果 列表选择框 1. 选中项 = "实验提醒"

发送数值 53

否则

发送数值 0

7.4 天气逻辑

7.4.1 初始化变量

- 初始化全局变量 (Json) 为 "0"

- 初始化全局变量 weather 为 "0"

7.4.2 按钮：获取天气 -> 被点击

- 调用 Web 客户端 1. 执行 GET 请求

URL 设置为天气查询接口：

<https://devapi.qweather.com/v7/weather/now?location=101120204&key=3130034d659043408fed902ab067ca2a>

7.4.3 Web 客户端 1 -> 获取文本

- 设置全局 Json 为：

在键 "now" 中查找关键字，如果未找到则返回 "not found"

在键 "text" 中查找关键字，如果未找到则返回 "not found"

- 设置 global weather 为 全局 Json

通过解析 JSON 文本，获取当前的天气信息。

7.4.4 调用 蓝牙客户端 1. 发送单字节数字

- 根据 global weather 的值，发送相应的数值

如果 global weather 等于 "晴" 发送 25

如果 global weather 等于 "多云" 发送 26

如果 global weather 等于 "少云" 发送 27

如果 global weather 等于 "晴间多云" 发送 28

如果 global weather 等于 "阴" 发送 29

如果 global weather 等于 "阵雨" 发送 30

如果 global weather 等于 "强阵雨" 发送 31

如果 global weather 等于 "雷阵雨" 发送 32
其他的天气情况可按照相似逻辑处理

8 总结与心得

起初：从四月底就开始进行原理图和 PCB 的设计，第一版在五一之前就完成了，当时也是有点急于求成，很多东西考虑的都不周全，但是在没有拿到具体实物之前就能完美的制作确实需要更多的经验。

中程：从五月开始，发现了第一次制板的所有问题，逐一解决，用了两个小时就重新设计了新的原理图和 PCB 板，并且这时候就有了想要制作 3D 外壳的想法，我就在想到底怎么设计才能让外壳上的按钮触发 PCB 上的按键，想了很久，突然想到在 pcb 背面焊接按钮，然后 3D 打印留出孔即可，果然效果很好，原始触发且灵敏，最后用热熔胶固定住以后也比较牢靠。之后就在面包板上进行全面的调试，写完了所有代码，再焊接了第二板 PCB，一通电就正常工作了，那一瞬间还是很开心的。

收尾：设计了手机 APP，调整了 UI 界面，使界面看起来更好看，然后学习设计了逻辑，多次调试，最后打包成了 apk 软件！之后设计将所有零件合理的摆放在盒子内，固定好，内部走线，做成了可实用的物品，效果良好！

过程中经历了很多问题，DEBUG 过程虽然已经很熟悉了，一层一层检查，但是还是总是会被一些硬件问题搞得摸不着头脑。这个过程积累了经验，从无到有，把一个项目亲手做出来，以产品的角度考虑问题，颇有收获！