项目编号: 001

**项 目 开 发 记 录**

**项目名称: 空气质量检测仪**

**立项部门:**

**项目领域:**

**申请日期:**

**结项日期: 2023.10.24**

# 目录

[目录 2](#_Toc29806)

[一、 项目介绍 3](#_Toc26468)

[1、 项目需求 3](#_Toc27780)

[2、设计方案 3](#_Toc9869)

[3、相关技术点 3](#_Toc12772)

[4、预计效果 3](#_Toc25207)

[二、设备开发 3](#_Toc7209)

[1.语音播报模块 3](#_Toc833)

[2. Wifi上传数据功能开发 3](#_Toc28188)

[3. DHT11温湿度检测 3](#_Toc30487)

[4. 光照及空气质量检测 3](#_Toc16820)

[5. RTC实时时钟模块 3](#_Toc10910)

[6. LCD屏显示模块 4](#_Toc26956)

[7. Flash存储 4](#_Toc1516)

[8. 字库下载 4](#_Toc5949)

[9. 整体项目开发 4](#_Toc6698)

[9.1 LVGL界面设计 4](#_Toc26485)

[9.2 语音播报 4](#_Toc19788)

[9.3 Wifi设置功能 4](#_Toc724)

[9.4 freeRTOS实时操作系统 4](#_Toc31875)

[三．项目结项 4](#_Toc4686)

[1.该项目能让自己有什么收获 4](#_Toc4955)

[2.总结项目中遇到的问题，及解决方法 5](#_Toc10264)

# 项目介绍

## 项目需求

1、需要采集室内温湿度、甲醛、TVOC、二氧化碳、烟雾浓度等数据,监测室内空气质量。

2、采用DHT11模块采集温湿度MQ2传感器采集烟雾浓度浓度光照以及CPU温度,KQM6600传感器采集TVOC、甲醛以及二氧化碳数据。

3、需要LCD屏实时显示采集的数据值,更新频率1秒。

4、需要设置WIFI模块,可以将数据通过网络定期上传到服务器，并从服务器获取实时时间以及实时天气。

5、需要SU\_03T语音合成模块,可以播报当前室内温度、湿度、甲醛浓度等信息。

6、要求系统稳定可靠,数据采集及显示延迟小于2秒。

7、要求整机外形小巧美观,便于放置在室内使用。

8、希望成本控制在500元人民币以下。

9、开发周期在2个月内完成。

## 设计方案

1、主控芯片选用STM32F103系列MCU,这是性价比高的通用系列,资源充足,支持多种接口,可以满足项目需求。

2、DHT11数字温湿度传感器通过单总线连接到STM32的I/O口,编写驱动程序获取温湿度数据，并通过串口发送给主机。

3、使用MQ2传感器采集烟雾浓度和光照值并输出模拟电压信号,采用STM32的ADC转换获取数字值。

4、KQM6600传感器采集TVOC、甲醛以及二氧化碳数据，并通过串口每秒一次发送给主机

5、ESP8266 WiFi模组配置为STA模式,连接路由器上传数据到云服务器，并从网上获取实时时钟和实时天气，配置为AP模式来与用户手机进行交互获取WiFi密码与账号等。

6、2.4英寸LCD显示屏采用SPI接口,分辨率320\*240,显示采集数据。

7、语音芯片使用串口连接STM32,利用g固件库合成播报语音。

8、电源部分设计稳压供电,确保各模组正常工作电压。

9、PCB布局紧凑,控制在15x10cm规格,外壳设计美观大方。

10、使用FreeRTOS操作系统进行任务管理,提高系统实时性。

11、使用LVGL图形库设计界面框架

## 相关技术点

1、DHT11数字温湿度传感器技术 - 工作原理、接口技术、数据采集方法

2、MQ2传感器技术 - 工作模式、ADC数据转换方法、空气质量评价标准

3、ADC技术 - 模数转换原理、STM32 ADC接口、配置方法、数据处理

4、WiFi网络技术 - ESP8266 WiFi模组使用、AT指令、TCP/IP网络协议

5、语音合成技术 - 语音芯片接口、播放原理、固件库设计、语音控制方法

6、LCD显示技术 - LCD接口类型、驱动芯片、图形库使用

7、FreeRTOS - 多任务系统构建、任务管理、任务通信机制

8、Flash存储器 - 工作原理、串口编程、存储管理、加密算法

9、PCB设计 - 小型化设计、模组布局、散热方案、成本控制

10、数据处理算法 - 平均过滤、中值滤波、数据拟合等

11、LVGL – 界面设计

## 4、预计效果

1. 采集数据准确可靠,温湿度误差小于±1°,PM2.5浓度误差小于10%。

2. 数据显示及时,从采集到界面显示延迟小于2秒。

3. 语音播报清晰流畅,可明确听取数字语音内容。

4. LCD显示界面友好易用,支持触摸控制,可手动刷新。

5. WIFI传输稳定,每10分钟定期上传数据到服务器。

6. 连续工作时间超过24小时,Standby时间超过1周。

7. 外形小巧精致,尺寸小于15x10x5厘米,可放置在桌面使用。

8. 成品售价不超过人民币200元,具有市场竞争力。

# 二、设备开发

## 1.语音播报模块模块介绍

SU-03T模块是一种语音合成模块，具有高度集成、高声音质、高可扩展性的特点。它可以将文字转换成语音并进行播放，让设备具备与人进行语音交互的能力。

**硬件电路**

硬件连接:su-03t模块的TX、RX、GND引脚分别连接到STM32的串口TX、RX、GND。供电电源为5V。

**SU-03T使用规则**

1. 硬件连接：将SU-03T模块通过串口或其他接口连接到主控板上。

2. 配置：根据需要在智能公元网站上设置固件库，并通过固件库设置SU-03T模块的参数，例如语速、音量、语音指令等。

3. 发送指令：主控板通过串口向SU-03T模块发送指令，指示它进行语音播放操作。指令可以是简单的文本指令，也可以是预先定义好的声音片段指令。

## Wifi上传数据功能开发ESP8266模块介绍

ESP8266模块是一种低功耗、低成本的Wi-Fi模块，它可以实现设备与互联网的通信。这个模块非常适合用于物联网项目中，它小巧灵活，性能出色，具有强大的网络连接能力。ESP8266可以通过AT指令进行配置,设置为STA模式连接路由器,或者AP模式创建热点。AP模式是指将ESP8266模块配置为一个Wi-Fi热点，其他设备可以连接到它并与之通信。STA模式是指将ESP8266模块连接到一个现有的Wi-Fi网络中，以便与互联网通信。

**AT指令函数封装**

关于esp8266的AT指令集封装主要可以分为STA配置和AP配置两个，主要流程包括编写一个串口发送函数，通过串口向esp8266发送AT指令，并将该模块的返回值通过串口1打印在串口助手上，并在串口发送函数的基础上再次封装，即在串口发送后来判断其返回的字符串中的固定字符，来判断AT指令是否成功，并在发送AT指令后适时延时来等待模块执行指令，并通过该函数的层层嵌套来实现对esp8266的配置

## DHT11温湿度检测

**DHT11模块介绍**

DHT11温湿度检测模块是一种常用的数字式温湿度传感器，它能够实时监测当前环境的温度和湿度，并将数据以数字信号的形式输出。DHT11模块使用的是单一线串行通信协议，只需要连接到主控板的一个数字引脚，就可以读取到当前环境的温湿度数值。DHT11温湿度检测模块的测量范围是在温度0℃至50℃，湿度20%至90%RH之间。也就是说，它适用于这个温度范围内的温湿度测量。如果超出了这个范围，可能会影响到它的测量精度和可靠性。DHT11模块的采样周期约为1秒。这意味着它每隔1秒钟会进行一次温湿度的测量。因此，您可以通过编程设置一个适当的时间间隔来获取最新的温湿度数据。在单总线时序方面，DHT11模块采用了单一线串行通信协议。这意味着只需要通过一个数字引脚与主控板连接即可。主控板发送请求信号给DHT11模块，模块会将温湿度数据以数字信号的形式回传给主控板。这种单一线的串行通信方式非常方便，减少了接线复杂性。

## 光照及空气质量检测

**MQ2传感器简介**

MQ2烟雾传感器是一种常见的气体传感器，专门用于检测空气中的烟雾浓度。它在各种应用中被广泛使用，比如家庭安防系统、火灾报警器以及空气质量监测装置等。MQ2烟雾传感器使用半导体气敏元件来探测空气中的烟雾。当烟雾颗粒进入传感器时，颗粒会与气敏元件上的化学物质发生反应，导致元件的电阻值相应变化。传感器将这种电阻值的变化转化为电信号输出，通过主控板进行处理和分析，最终得出烟雾的浓度。MQ2烟雾传感器不仅可以检测常见的烟雾，如烟草烟雾，还可以检测到气味较浓的气体，如液化气、丙烷、甲烷等易燃气体。这使得它成为一个多功能的气体传感器。

**ADC采样简介**

ADC是一种将模拟信号转换为数字信号的电子设备。它在各种电子系统中被广泛应用，用于测量和采集模拟量的数值。

ADC的采集原理是通过将模拟信号离散化为一系列数字值。这个过程分为两个主要阶段：采样和量化。首先是采样阶段，ADC会周期性地对模拟信号进行采样，获取连续时间内的一系列采样值。接下来是量化阶段，ADC会将采样到的模拟信号离散到特定的量化级别上。量化级别的数量取决于ADC的位宽，例如，8位ADC会将模拟信号量化为256个离散的数字级别，而12位ADC则会有4096个级别。转换后的数字精度取决于ADC的位宽。位宽越高，即采用更多的位数来表示模拟信号，精度就会越高。

测量模拟量的范围取决于ADC的参考电压。ADC通常有一个参考电压输入，测量的模拟量的范围通常是0到参考电压之间。如果输入电压超出了这个范围，ADC的结果将失真。

ADC多通道采样时，要使用ADC\_InitStruct.ADC\_NbrOfConversion = x;设置需要转换的通道个数，并打开扫描模式。

## RTC实时时钟模块

**RTC实时时钟简介**

RTC实时时钟模块是一种常见的电子设备，用于提供准确的日期和时间。它在很多应用中都有广泛的应用，比如计时器、闹钟、数据记录等

**RTC工作原理**

RTC实时时钟模块基于振荡器，该振荡器提供稳定的时钟信号。RTC是一个独立的定时器并使用这个时钟信号来更新RTC计数器，从而保持准确的日期和时间。RTC和时钟配置处于后备区域，系统复位时数据不清零

**RTC配置**

确定时钟源并启用RTC时钟，可选择3中RTC时钟源

HSE时钟除以128 通常为 8MHZ / 138

LSE振荡器时钟 通常为32.768KHZ

LSI振荡器时钟 40KHZ

配置RTC时钟预分频器和计数器精度。

配置日期和时间，并将其写入RTC寄存器

**后备寄存器**

BKP可用以存储用户应用程序数据，当VDD（2 -- 3.6）电源被切断，他们仍然由VBAT（1.8 -- 3.6）维持供电，当系统在待机模式下被唤醒，或系统复位或电源复位，他们不会被复位。向后备寄存器写入数据时需要开启电源和BKP时钟并且使能后备寄存器

**低功耗模式**

低功耗模式包括睡眠模式（sleep），停机模式（stop）和待机模式（standby），可在系统空闲时，降低STM32的功耗，延长设备待机时间

## LCD屏显示模块

**驱动器模块介绍**

**我们的**液晶屏内部包含有一个液晶控制芯片 ILI9341，这是一种常用的液晶显示控制器，广泛应用于各种嵌入式系统和电子设备中。ILI9341芯片支持最高分辨率为320x240像素的液晶屏，采用了SPI（串行外围设备接口）作为控制和通信接口。这种接口速度较快，并且仅需要4个信号线（时钟、数据输入、数据输出和片选）即可实现与主控芯片之间的数据交互。ILI9341芯片支持全彩显示，可实现对图像和文本的高质量显示。芯片内部集成了一部分显示数据缓存存储器，可以缓存图像和文本数据，提高显示的效率和质量。

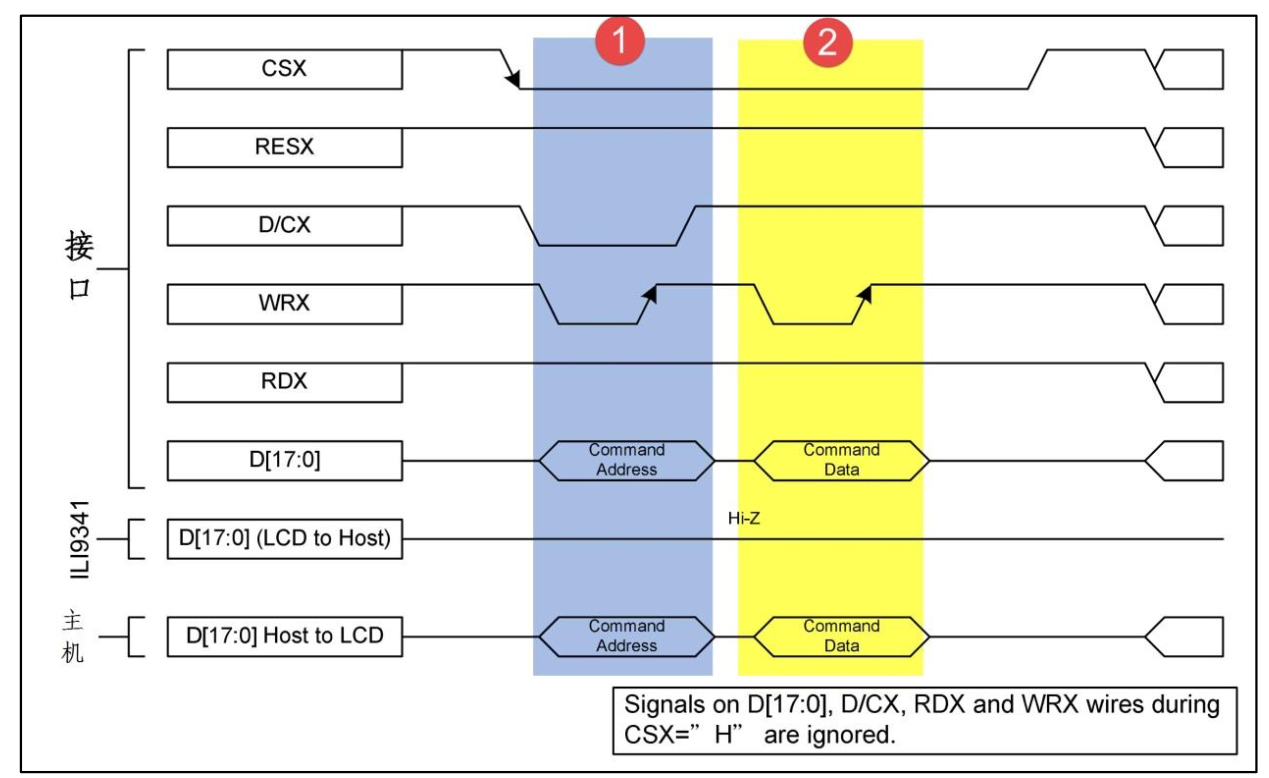
**LCD屏显示原理**

LCD屏中的液晶是一种介于固体和液体之间的特殊物质，它是一种有机化合物，常态下呈液态，但是它的分子排列却和固体晶体一样非常规则，因此取名液晶。 如果给液晶施加电场，会改变它的分子排列， 从而改变光线的传播方向， 配合偏振光片，它就具有控制光线透过率的作用，再配合彩色滤光片，改变加给液晶电压大小，就能改变某一颜色透光量的多少。 利用这种原理，做出可控红、绿、蓝光输出强度的显示结构，把三种显示结构组成一个显示单位，通过控制红绿蓝的强度，可以使该单位混合输出不同的色彩，这样的一个显示单位被称为像素。

**硬件接口时序**

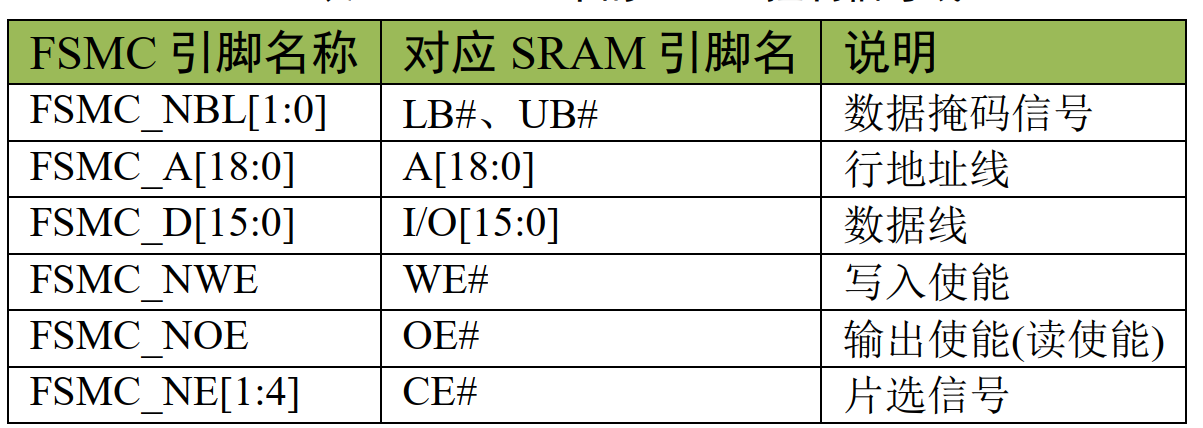
液晶屏的 ILI9341 控制器在出厂前就已经按固定配置好(内部已连接硬件电路)，它被配置为通过 8080 接口通讯，使用 16 根数据线的 RGB565 格式。内部硬件电路连接完， 剩下的其它信号线被引出到 FPC 排线，最后该排线由 PCB 底板引出到排针，排针再与实验板上的 STM32 芯片连接。

|  |  |  |
| --- | --- | --- |
| 信号线 | ILI9341的引脚 | 说明 |
| DB[15:0] | D[15:0] | 数据信号 |
| RD | RDX | 读数据信号，低电平有效 |
| RS | D/CX | 数据/命令信号，高电平时， D[15:0]表示的是数据(RGB像素数据或命令数据)，低电平时 D[15:0]表示控制命令 |
| RST | RESX | 复位信号，低电平有效 |
| WR | WRX | 写数据信号，低电平有效 |
| LCD\_CS | CSX | 片选信号，低电平有效 |
| LCD\_BK | - | 背光信号，低电平点亮 |



**FSMC工作原理**

FSMC 是 Flexible Static Memory Controller 的缩写，译为灵活的静态存储控制器。它可以用于驱动包括 SRAM、NOR FLASH 以及 NAND FLSAH 类型的存储器，不能驱动如 SDRAM 这种动态的存储器。



其中地址线 FSMC\_A 和数据线 FSMC\_D 是所有控制器都共用的。比较特殊的 FSMC\_NE 是用于控制 SRAM 芯片的片选控制信号线， STM32 具有FSMC\_NE1/2/3/4 号引脚，不同的引脚对应 STM32 内部不同的地址区域，从而达到控制多块 SRAM 芯片的目的。

**地址编程**

FSMC 连接好外部的存储器并初始化后，就可以直接通过访问地址来读写数据，而使用 FSMC 外接存储器时，其存储单元是映射到 STM32 的内部寻址空间的；在程序里，定义一个指向这些地址的指针然后就可以通过指针直接修改该存储单元的内容， FSMC 外设会自动完成数据访问过程，读写命令之类的操作不需要程序控制。

## Flash存储

**W25q64模块介绍**

W25Q64是一款64Mb（8MB）容量的串行闪存存储器，由Winbond生产。它采用SPI（Serial Peripheral Interface，串行外设接口）协议进行通信，可以用于存储各种数据，如程序代码、配置信息和用户数据等。W25Q64模块的引脚数多达8个，并且支持低功耗休眠模式，适用于各种嵌入式应用场景。

**空间划分**

W25Q64闪存可被划分为多个扇区（Sector）、块（Block）和页（Page）。一个扇区通常包含若干个块，一个块包含若干个页。每个扇区、块和页都有唯一的地址，可以通过地址来访问存储的数据。

**接口通信**

W25Q64模块使用SPI接口来与微控制器或其他设备进行通信。SPI接口通常由四个信号线组成：片选（CS）、时钟（SCK）、数据输入/输出（MISO/MOSI）和数据/命令选择（D/C）。通过SPI接口，微控制器可以发送命令和数据，读取和写入闪存存储器中的内容。

**读写时序**

W25Q64模块的读写时序遵循SPI协议。对于读操作，首先选择W25Q64模块，然后发送读取指令和地址，接着模块会返回指定地址处的数据。对于写操作，先选择模块，发送写指令和地址，然后发送要写入的数据。

## 字库下载

**汉字编码系统GB2312**

GB2312是一种汉字编码系统，它将每个汉字表示为一个双字节编码。每个汉字的数字表示由两个十六进制数字组成，表示该汉字在GB2312编码表中的位置。高字节范围为0xA1至0xF7，低字节范围为0xA1至0xFE。通过将汉字的高字节和低字节的编码值组合在一起，可以表示出GB2312编码表中的每个汉字。GB2312编码系统只包含了常用的6763个汉字，不包括繁体字等其他字符。对于非GB2312编码的汉字，可以使用更广泛的Unicode编码系统来表示。

**LVGL创建并使用字库**

首先选择TrueType 字体文件 (.ttf)并确保字体文件支持所需的字符集和大小。使用LVGL的在线字体转换工具将字体文件转换为所需的格式。将生成的字库文件（bin文件和C源文件）添加到LVGL项目中。包含相应的宏定义和头文件即可使用该字体

## 整体项目开发

### LVGL界面设计

Lvgl界面设计主要通过Gui\_Guider这个软甲完成，Gui\_Guider中设计界面并将生成的代码导入到工程中，主要界面包括室内空气数据和温度显示界面，实时天气显示界面，日历显示界面和WiFi切换界面

### 语音播报

语音播报主要通过su\_03t语音模块实现，该模通过UART主控板进行通信，通过将实现事先制作好的固件库烧录到su\_03t中来实现语音控制，su\_03t识别特定语音，并将其转换为相应的指令通过串口发送给主控板，主控板通过判断，将特定的信息回传给su\_03t，su\_03t根据回传信息播放相应的语音

### Wifi设置功能

WiFi设置主要通过esp8266的AP模式完成，将esp8266配置为AP模式，手机连接esp8266的热点，通过空气质量检测仪app向esp8266发送STA模式先所需的WiFi名称和密码并通过串口发送给主控芯片，主控芯片将其写入w25q64中，等待STA模式下连接网络时再读出

### freeRTOS实时操作系统

（freeRTOS如何进行任务管理，任务的几个模式，任务栈表示什么的空间大小，任务间通信有哪些，任务的阻塞机制有哪些）

**freeRTOS简介**

freeRTOS（Free Real-Time Operating System）是一个开源的实时操作系统，被广泛应用于嵌入式系统的开发中。它的目标是提供一个简洁、高效、可靠的操作系统，适用于各种资源受限的嵌入式设备。作为一个实时操作系统，freeRTOS被设计用于需要实时性能的应用。它采用优先级调度算法，确保高优先级任务能够及时响应，并以可预测的方式执行。freeRTOS支持多任务并发执行和管理，每个任务独立运行且具有自己的优先级和调度规则。freeRTOS还提供了多种任务间通信机制，使不同任务之间可以进行数据交换、同步和互斥访问共享资源。这些机制包括消息队列、信号量、事件标志和互斥量，可以根据应用需求选择适当的通信方式。除了实时性和任务管理，freeRTOS还注重低功耗支持。它可以与低功耗模式和硬件中断相结合，来有效管理系统的能耗。例如，它提供了睡眠模式和节能模式，以便灵活地配置和控制系统的功耗。freeRTOS具有高度的可移植性，支持多种处理器架构和编译器。无论是ARM、AVR、PIC还是MSP430等，都可以在freeRTOS上运行，并且可以方便地移植到不同的硬件平台上。

**freeRTOS任务管理**

freeRTOS使用任务（Tasks）的概念来管理代码的执行。每个任务是一个独立的函数，可以并发地运行在系统中。通过任务管理，您可以创建、启动、停止、删除和切换任务等。

**freeRTOS任务模式**

freeRTOS提供了多种任务模式，包括优先级任务模式、循环任务模式和一次性任务模式。优先级任务模式允许您为每个任务分配一个优先级，并根据优先级决定任务何时执行。循环任务模式允许您定义一个任务的函数，然后以循环方式执行该函数。一次性任务模式是在任务完成后自动删除任务。

**任务栈**

任务栈是每个任务的私有堆栈，用于保存任务执行过程中的临时变量、函数调用和返回地址等。任务栈的大小取决于任务函数的复杂性和任务所需的内存。通常，您可以根据任务的代码复杂性和内存需求来选择适当的任务栈大小。过小的任务栈可能导致堆栈溢出，而过大的任务栈则浪费内存。

**任务间通信**

freeRTOS提供了多种方式来实现任务间的通信，包括消息队列、信号量、事件组和共享资源等。消息队列允许任务通过发送和接收消息来进行通信。信号量允许任务互斥地访问共享资源，以避免竞态条件。事件组允许任务等待和触发特定的事件。共享资源允许多个任务同时访问相同的资源。

**任务的阻塞机制**

freeRTOS提供了多种任务阻塞机制，用于控制任务的执行和等待。例如，延时阻塞允许任务暂停一段时间后再继续执行。指定事件阻塞允许任务等待特定的事件发生再继续执行。互斥阻塞允许任务在访问共享资源时阻塞其他任务的访问。

# 三．项目结项

## 1.该项目能让自己有什么收获

首先，在这个项目中，涉及到了多种传感器，例如语音合成模块、WiFi模块、温湿度传感器、光照传感器等多种技术的应用和开发。通过与不同硬件模块的接入和配置，我能够更加熟练地理解和运用各种传感器以及互联网设备的开发技术。其次，通过这个项目，我将进一步加深对单片机操作系统FreeRTOS的理解和运用，在这个项目中，我们需要进行任务优先级调度、堆栈资源的合理分配、进度控制等各种任务管理工作。通过这个项目，我对任务管理有了新的理解。并且学会使用STM32的ADC功能采集模拟传感器的输出值。了解ADC的转换原理和编程方法。学会驱动LCD屏,绘制界面。熟练使用LVGL图形库进行界面设计。通过这个项目,将软硬件知识联系起来,把理论应用到实践中,对各部分知识点有更深入的理解。总体来说,这个项目对提高我的嵌入式系统设计与开发能力很有帮助。