

## 摘要

随着社会的迅猛发展带来了许多环境问题，其中就包括空气污染，实时检测空气质量对环境数据分析有极大帮助，方便人们能够及时发现和调节空气质量水平，呵护自身健康。故本文将介绍一款基于 STM32 的空气质量检测系统设计，该系统使用 STM32F103C8T6 作为核心主控，基于传感器采集的经过计算处理后的空气质量数据，采用 NB-IoT（窄带物联网）技术进行无线通信数据传输，将数据发送到云平台，实现远程实时监测，当有害气体浓度含量超过设定值时，系统会自动启动报警功能，提醒用户及时采取相应措施。实验结果表明该系统能够有效检测出空气质量异常，且具有监测范围广，功耗低的优点。可应用于智能城市建设、智能家居监测和环境污染监测等领域。

**关键词：**STM32；传感器；NB-IoT；空气质量检测

## ABSTRACT

With the rapid development of society, many environmental problems have been brought about, including air pollution. Real-time air quality detection is of great help to environmental data analysis, so that people can timely discover and adjust the level of air quality and protect their health. Therefore, this paper will introduce the design of an air quality detection system based on STM32. The system uses STM32F103C8T6 as the core control. Based on the air quality data collected by sensors after calculation and processing, it adopts NB-IoT (narrowband Internet of Things) technology to carry out wireless communication data transmission and send the data to the cloud platform. To realize remote real-time monitoring, when the concentration of harmful gas exceeds the set value, the system will automatically start the alarm function to remind users to take corresponding measures in time. Experimental results show that the system can effectively detect air quality anomalies, and has the advantages of wide monitoring range and low power consumption. It can be used in smart city construction, smart home monitoring and environmental pollution monitoring.

**Key Words:** STM32; Sensor; NB-IoT; Air quality detection

## 目 录

第 1 章	引言	1
1.1	选题背景	1
1.2	研究目标和意义	1
第 2 章	系统方案选择及关键技术	3
2.1	系统材料选择	3
2.1.1	主控单片机的选择	3
2.1.2	温湿度传感器的选择	3
2.1.3	气体浓度传感器的选择	4
2.1.4	甲醛传感器的选择	4
2.1.5	NB-IoT 通信模块的选择	5
2.2	关键技术	6
2.2.1	NB-IoT 技术	6
2.2.2	MQTT 协议	7
第 3 章	系统硬件设计	9
3.1	系统总体架构	9
3.2	STM32F103C8T6 最小系统板	9
3.2.1	微处理器模块介绍	9
3.2.2	晶振电路设计	10
3.2.3	复位电路设计	11
3.2.4	调试下载电路设计	11
3.2.5	启动电路设计	12
3.3	传感器模块	13
3.3.1	温湿度检测模块	13
3.3.2	气体浓度检测模块	14
3.3.3	甲醛浓度检测模块	15
3.4	声光报警模块	16
3.5	供电电路设计	17
第 4 章	系统软件程序设计	18

4.1 主程序设计 .....	18
4.2 温湿度采集程序设计 .....	19
4.3 气体浓度采集程序设计 .....	21
4.4 甲醛浓度采集程序设计 .....	23
4.5 OLED 显示模块程序设计.....	25
4.6 按键模块程序设计 .....	25
4.7 NB-IoT 通信模块程序设计 .....	27
第 5 章 系统调试及展示.....	30
5.1 硬件调试.....	30
5.1.1 PCB 电路板设计及焊接.....	30
5.1.2 电源电路调试 .....	31
5.1.3 传感器调试 .....	32
5.1.4 通信模块调试 .....	32
5.2 系统配置及流程测试 .....	33
5.2.1 云平台及 APP 配置 .....	33
5.2.2 系统测试 .....	34
第 6 章 结论.....	37
参考文献 .....	38
致谢 .....	39
附录 .....	40
附录一：开发板原理图.....	40
附录二：程序代码.....	40
外文资料原文.....	53
译文 .....	61

## 第1章 引言

### 1.1 选题背景

空气质量问题是全球范围内的严峻环境问题，室外空气污染是影响低收入、中等收入和高收入国家中每一个人的主要环境卫生问题。根据世界卫生组织的估计，2019 年城市和农村地区的环境（室外）空气污染导致全球 420 万人过早死亡，原因主要是暴露于细颗粒物，这些颗粒物能导致心血管和呼吸道疾病以及癌症<sup>[1]</sup>。同时，室内空气污染也是一种潜在的健康风险，因为它会增加许多健康问题，如哮喘或其他呼吸系统疾病。为了解决空气质量问题并保护人类健康，需要采取有效的措施，包括加强环境监测、减少污染排放、提高公众环保意识等。在解决空气质量问题方面，监测是关键的一步。117 个国家的 6000 多个城市正在实时监测空气质量<sup>[2]</sup>，这对于及时采取措施改善空气质量非常重要。空气质量的检测可以通过专业的监测站点和设备进行。中国环境监测总站国家环境空气监测网业务应用系统就是我国建立的环境空气监测网络，该系统实现了对全国范围内的空气质量进行实时监测和数据共享。

空气质量实时检测设备通常会配套空气监测系统平台，并配合手机、电脑等电子屏幕实时发布展示数据，这样的组合是为了实现更多的功能，而这些功能通常是空气检测所不具备的。例如基础的实时预警、大屏显示、远程查看、健康指数、治理建议、污染时段分析、污染区域定位等。结合该些功能就可以实现对于室内空气污染原因、污染源的排查及空气治理方式有效性的反复验证。另外，新的空气质量实时检测设备及系统又进一步突出了“智慧”的概念，结合新风和空气净化联动功能，可实现对于所处环境空气质量的自动调节和治理。当然，这一智慧功能实现的基础还是源于对于空气质量的实时检测数据。

### 1.2 研究目标和意义

通过监测和收集大量的环境数据，以了解空气质量问题的发生和变化趋势。这个系统可以帮助研究人员分析空气中的污染物类型、浓度和分布特点，为进一步制定和调整环保政策提供科学的依据和参考，帮助政府进行空气质量监管和治

理，促进环保政策的制定和调整。此外，空气质量检测还能够为企业经营决策提供依据，帮助企业制定和实施环保措施，提高生产过程中的资源利用效率和减少污染排放。同时，能够为公众提供实时、准确、科学的空气质量信息，保障公众的健康和安全，加强公众对大气污染的认识并促进公众参与环保工作。

## 第2章 系统方案选择及关键技术

### 2.1 系统材料选择

#### 2.1.1 主控单片机的选择

主控选择的是 STM32F103C8T6 最小系统板，STM32F103C8T6 是一款基于 ARM Cortex-M 内核的 STM32 系列的 32 位的微控制器，程序存储器容量是 64KB，通常需要 2V~3.6V 电压驱动工作，工作温度为 -40℃ ~ 85℃。STM32F103C8T6 最小系统板包括晶振电路、电源供电电路、复位电路以及调试电路，晶振电路主要为芯片提供时钟信号；供电方面最小系统板自带电压稳压芯片、方便进行电平转换可以由 MircoUSB 5V 接口接入、或者外部 5V IO 口接入、如果有条件提供 3.3V 电平(核心需要的工作电压、也可以直接接入标示为 3.3 的引脚、为核心板提供电源、注意严禁多电源输入、以防引起短路故障)；复位电路产生复位脉冲，从而使系统复位；调试采用的是 SWD 调试该接口至少需要 3 个接口就能完成调试下载的任务，相对于传统的 JTAG 调试有不少的好处。

#### 2.1.2 温湿度传感器的选择

DHT11 是一款含有已校准数字信号输出的温湿度复合传感器。它使用了专用的数字模块采集技术和温湿度传感技术，具有极高的可靠性与长期稳定性。传感器包含一个电阻式感湿元件和一个 NTC 测温元件，同时与一个高性能 8 位单片机相连接。每一个 DHT11 传感器都要经过在极为精确的湿度校验室中校准，校准系数会以程序的形式储存在 OTP 内存中，传感器内部在检测信号的处理过程中要调用这些校准系数。单线制串行接口，使系统集成变得简易快捷。超小的体积、极低的功耗，信号传输距离可达 20 米以上，使其成为各类应用甚至最为苛刻的应用场合的最佳选则。该产品为 4 针单排引脚封装，连接方便，特殊封装形式可根据用户需求而提供<sup>[3]</sup>。综合来看，DHT11 具有品质卓越、超快响应、抗干扰能力强、性价比高等优点，是一款高精度且易于操作的温湿度传感器。



图2-1 DHT11 传感器

### 2.1.3 气体浓度传感器的选择

气体浓度传感器选用的是 MQ135 传感器，是一款有害气体浓度传感器，可以检测多种有害气体的浓度，气体传感器使用二氧化锡( $\text{SnO}_2$ )作为气敏材料，当将其置于存在污染气体环境中时，传感器的电导率会随着气体浓度的增加而增大，通过简单的电路可以将电导率的变化转换为与该气体浓度相对应的输出信号。MQ135 气体传感器对硫化物、苯系蒸汽、氨气的灵敏度高，在监测烟雾和其它有害气体方面也很理想。综合来看，气体传感器是一种通过检测污染气体的电导率变化来实现气体浓度检测的设备，MQ135 气体传感器因其低成本、高灵敏度等特点广泛应用于各种有害气体的检测和监测。



图2-2 MQ135 传感器

### 2.1.4 甲醛传感器的选择

采用 ZE08-CH<sub>2</sub>O 传感器检测甲醛浓度。ZE08-CH<sub>2</sub>O 型电化学甲醛模组是一种通用小型化甲醛传感器模组。它依据电化学原理对空气中存在的甲醛( $\text{CH}_2\text{O}$ )进行检测，具有高选择性和稳定性的优点。同时，模组还内置温度传感器，可以进行温度补偿，来提高检测精度。另外，ZE08-CH<sub>2</sub>O 还有数字输出和模拟电压输出两种方式，方便用户进行数据的采集与处理。



该传感器模组广泛应用于室内空气质量检测、环境监测等领域。在家庭装修中，使用该传感器可以及时感知室内甲醛浓度，避免装修材料的甲醛含量过高而对健康造成影响。此外，它还可以用于生产过程中对甲醛、化工产品、涂料等有害物质的检测。



图2-3 ZE08-CH2O 传感器

### 2.1.5 NB-IoT 通信模块的选择

NB-IoT 又叫做窄带物联网，其构建于蜂窝网络，是万物互联网络的一个重要分支。相比于物联网的传统的技术，NB-IoT 具有接入数量多、功耗低、覆盖范围广、成本低的优点。所以本设计选用了移远通信公司的 BC26 作为 NB-IoT 模组，BC26 具有超小体积，尺寸仅为 17.7x15.8x2.0mm，能最大限度地满足可穿戴设备、智能家居、安防、资产追踪、智能表计、便携式健康监控仪器等紧凑型终端设备的需求。BC26 还支持低供电电压范围（2.1V-3.63V），更适合 NB-IoT 技术的应用场景。BC26 提供丰富的外部接口和协议栈，同时可支持中国移动 OneNET/Andlink、中国电信 IoT/AEP 以及阿里云 IoT 等物联网云平台，为客户的应用提供极大的便利。



图2-4 BC26 模块

## 2.2 关键技术

### 2.2.1 NB-IoT 技术

随着智能城市、大数据时代的到来，万物互联已成为可能。许多企业预测未来全球的物联网连接数量将达到千亿级别。目前，已经有许多物品通过蓝牙、Wi-Fi 等短距离通信技术连接在一起，但这些连接大多数并不是由运营商提供的移动网络。为了满足各种不同物联网业务的需求，3GPP 进行了基于窄带业务应用场景的技术研究，以增强移动通信网络的功能。NB-IoT 是 IoT 领域一种新兴的技术，是一种专为万物互联打造的蜂窝网络连接技术。顾名思义，NB-IoT 所占用的带宽很窄，仅需要 180KHz，而且其使用 License 频段，可采取带内、保护带或独立载波三种部署方式<sup>[4]</sup>，与现有网络共存，并且能够直接部署在 GSM、UMTS 或 LTE 网络<sup>[5]</sup>，即 2/3/4G 的网络上，实现现有网络的复用，降低部署成本，实现平滑升级。移动网络作为全球覆盖范围最大的网络，其接入能力可谓得天独厚，所以相比于 WiFi、蓝牙、ZigBee 等连接方式，基于蜂窝网络的 NB-IoT 连接技术的前景更加被看好<sup>[6]</sup>，已经逐渐成为开启万物互联时代的钥匙，且被商用到物联网行业中。

NB-IoT 具有以下四大特点：

一、广覆盖。NB-IoT 的信号传输覆盖范围更广，比现有的 GSM、宽带 LTE 等网络覆盖增强了 20dB，可以覆盖到深层地下 GSM 无法覆盖到的地方。实现原理主要是缩小带宽，提升功率谱密度，重复发送获得时间分集增益<sup>[7]</sup>。

二、大连接。NB-IoT 能够支持每个小区达到 5 万个连接，从同一个基站下接入设备增多 50-100 倍，为实现万物互联提供海量连接。实现原理在于采用话务模型和适应时延不敏感的特点，保存更多接入设备的上下文，在休眠态和激活态之间切换，减少空口信令交互，提升频谱密度<sup>[8]</sup>。

三、低功耗。NB-IoT 终端在 99% 的时间内处于休眠状态，并且集成了多种节电技术，待机时间可达 10 年。实现原理包括 PSM 低功耗模式和 eDRX 扩展的非连续接收省电模式<sup>[9]</sup>。

四、低成本。硬件可剪裁，软件按需简化，这确保了 NB-IoT 的成本低廉。NB-IoT 通信单模块成本不足 5 美元<sup>[10]</sup>。

此外，NB-IoT 还适用于低速率和低移动性场景。其理论上行速率仅有 56kbps，下行速率为 21.25kbps，同时仅支持终端设备在 30km/h 的移动速率下实现小区切换，远低于 4G 支持 250km/h 的速率。

## 2.2.2 MQTT 协议

### 2.2.2.1 MQTT 概述

MQTT(Message Queuing Telemetry Transport, 消息队列遥测传输协议), 是一种基于发布/订阅(publish/subscribe)模式的“轻量级”通讯协议, 该协议由 IBM 在 1999 年发布<sup>[11]</sup>, 构建于 TCP/IP 协议上。MQTT 最大优势是可以通过少量的代码和有限的带宽, 为远程连接设备提供实时可靠的消息服务<sup>[12]</sup>, 作为一种低开销、低带宽占用的即时通讯协议, 使其在小型设备、移动应用、物联网等方面有较广泛的应用。同时, MQTT 还具有以下优点:

- 1、消息头可以缩短为 2 个字节, 减少连接开销;
- 2、使用持久连接可以避免每次请求都需要重新建立连接的高开销问题;
- 3、可以从诸如断开连接之类的故障中恢复, 无需进一步的代码要求;
- 4、专为低功耗目标而设计, 适用于移动设备等需要低功耗的场景;
- 5、可以通过 WebSockets 连接防火墙, 并支持端口限制;
- 6、简单易用, 易于在不同平台上实现。

### 2.2.2.2 MQTT 通信模型

MQTT 协议提供了一对多的消息发布功能, 可以实现在服务器和多个客户端之间进行消息传递, 并且能够降低应用程序之间的耦合度, 用户只需要编写极少量的应用代码就能实现一对多的消息发布与订阅。MQTT 协议采用<客户端-服务器>模型, 其中主要有三种身份: 发布者、服务器和订阅者<sup>[13]</sup>。MQTT 协议中, 消息的发布者和订阅者都是客户端, 服务器作为一个中转的存在, 将发布者发布的消息进行转发给所有订阅该主题的订阅者。发布者可以发布其权限内的所有主题, 而且消息发布者同时可以作为订阅者, 实现了生产者与消费者的脱耦<sup>[14]</sup>, 发布的消息可以同时被多个订阅者订阅。MQTT 通信模型示意图如下:

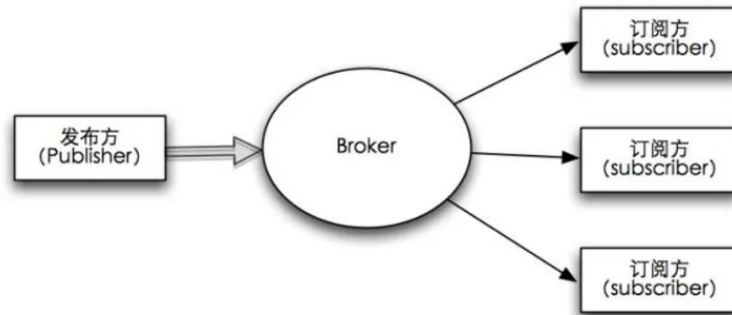


图2-5 MQTT 通信模型

### 2.2.2.3 MQTT 协议通信过程

MQTT 协议采用发布-订阅模式，其中消息的发布者和订阅者之间通过一个称为消息代理(broker)的中间件进行通信。以下是 MQTT 通信的基本过程：

- 1.客户端连接到 MQTT 代理。
- 2.客户端通过 MQTT 协议向代理发布一条消息，包括消息主题和消息内容。
- 3.客户端连接到 MQTT 代理。
- 4.客户端向代理订阅某个或某些主题。
- 5.当有消息发布到某个主题时，代理将该消息发送给所有已订阅该主题的客户端。
- 6.客户端接收到消息后可以进行处理，例如更新设备状态或执行某个动作。
- 7.客户端可以断开与代理的连接，也可以重新连接。需要注意的是，MQTT 协议支持多种 QoS 级别，分别为 0、1、2 级别。不同的 QoS 级别对消息传递的可靠性和延迟有不同的影响。例如，QoS0 级别为最低级别，消息可能会重复发送或者丢失；而 QoS2 级别为最高级别，消息会确保被发送且仅被发送一次。

## 第3章 系统硬件设计

### 3.1 系统总体架构

该系统主要由 STM32F103C8T6 单片机作为主控制器。将温湿度传感器、有害气体浓度传感器、甲醛传感器采集的数据传递给单片机，经过计算将数据显示在 OLED 显示屏上，通过 NB-IoT 模块将数据发布到云平台，通过按键设置各参数的阈值，当超过阈值时通过蜂鸣器和 LED 灯达到声光报警的效果。

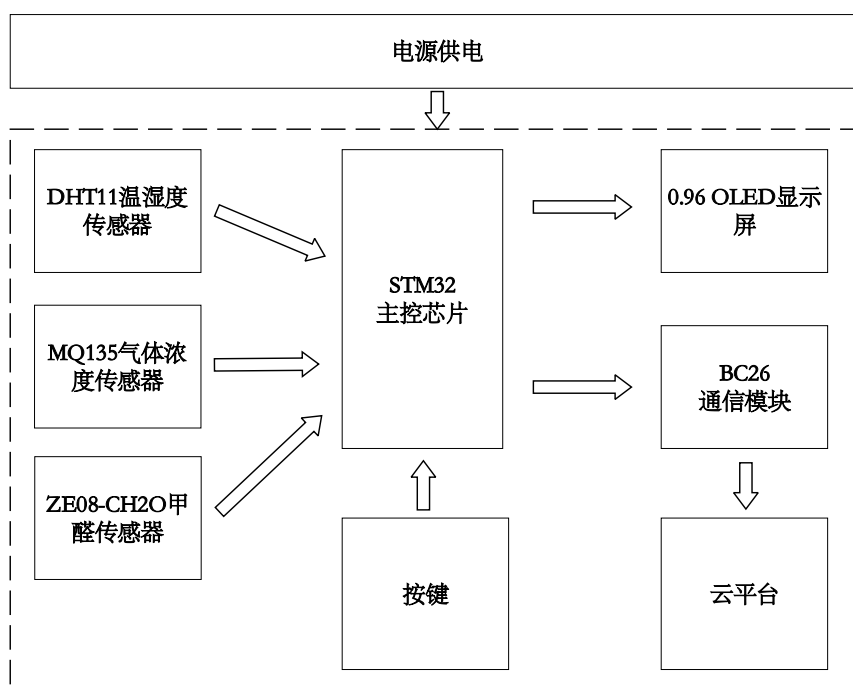


图3-1 系统框架图

### 3.2 STM32F103C8T6 最小系统板

#### 3.2.1 微处理器模块介绍

STM32F103C8T6 属于 STM32 系列的低成本产品，其拥有的资源包括 64KB 的 Flash 存储器、20KB 的 SRAM 存储器，37 个 GPIO 分别为 PA0-PA15、PB0-PB15、PC13-PC15、PD0-PD1；7 个定时器包括 1 个用于产生 PWM 控制电机的高

级定时器，3 个通用定时器，2 个看门狗定时器，1 个 24bit 的自减型滴答定时器 systick；通信接口包括 2 个 I<sup>2</sup>C 接口、3 个 USART 接口、2 个 SPI 接口、1 个 CAN 接口、1 个 USB2.0 全速接口；2 个 DMA 控制器共 12 个通道，其中 DMA1 拥有 7 个通道，DMA2 拥有 5 个通道，每个通道都用来管理外部设备对片内存储器的访问请求，此外还有一个仲裁器来协调各个 DMA 请求的优先级。除了管理外部设备（Timer、ADC、SPI、I<sup>2</sup>C、USART）到存储器之间数据的双向传输，DMA 还能够管理存储器之间的数据传输<sup>[15]</sup>。

### 3.2.2 晶振电路设计

图 3-2 是 STM32F103C8T6 最小系统板晶振电路，该电路包括了晶振 Y2、Y3，四个电容 C9、C12、C13、C14 以及电阻 R6。晶振 Y2 一般采用 8MHz 的晶振，单片机可以在 HSI 或者 HSE 模式下工作，HSI 是内部高速 RC 振荡器，而 HSE 是外部晶振。在本电路中，我们使用的是 HSE 模式。电容 C9、C12、C13、C14 的主要作用是使晶振的输出信号保持稳定。一般来说，这四个电容的值都是 20pF。电阻 R6 的作用是限制晶振电流，一般建议使用 10k $\Omega$  的电阻。值得注意的是，在 STM32F103C8T6 最小系统板中，晶振电路还需要连接到芯片的 OSCIN 和 OSCOUT 引脚。如果在电路设计中没有将这两个引脚连接到晶振电路，芯片将无法正常工作。

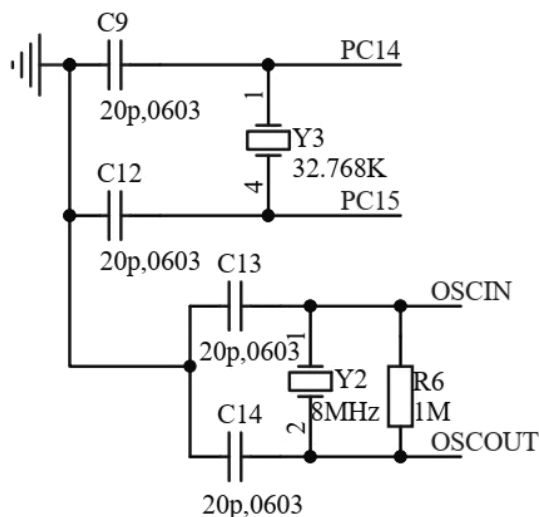


图3-2 晶振电路

### 3.2.3 复位电路设计

如图 3-3，为了设计 STM32F103C8T6 的最小系统板复位电路，需要使用电容、电阻、开关。下面是 STM32F103C8T6 最小系统板复位电路的设计步骤：1. 将电容连接到 STM32F103C8T6 芯片的 RESET 引脚和 GND 引脚之间。2. 将 10K 的电阻连接到 STM32F103C8T6 芯片的 RESET 引脚和 GND 引脚之间。3. 将按键开关的一个端口连接到 RESET 引脚，另一个端口连接到 GND 引脚。4. 完成以上连接后，可以通过按下按键开关来复位 STM32F103C8T6 芯片。这个复位电路的工作原理是：当按键开关被按下时，电路会将 RESET 引脚连接到 GND 引脚，从而将芯片复位。同时，电容会起到滤波的作用，避免复位引脚出现噪声信号干扰芯片正常运行。

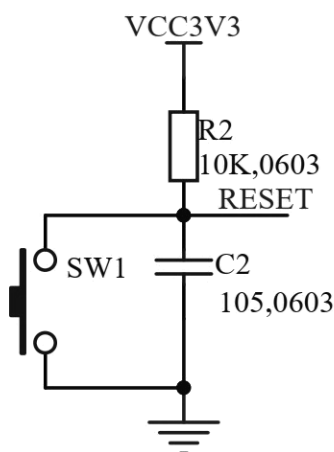


图3-3 复位电路

### 3.2.4 调试下载电路设计

STM32F103C8T6 的最小系统板的程序烧录方法包括 JTAG、SWD 和串口下载三种。

JTAG 是一种国际标准测试协议，主要用于芯片内部测试。现在多数的高级器件都支持 JTAG 协议，如 ARM、DSP、FPGA 器件等。标准的 JTAG 接口是 4 线：TMS、TCK、TDI、TDO，分别为模式选择、时钟、数据输入和数据输出线。相关 JTAG 引脚的定义为：

**TMS：**测试模式选择，TMS 用来设置 JTAG 接口处于某种特定的测试模式；

TCK: 测试时钟输入;

TDI: 测试数据输入, 数据通过 TDI 引脚输入 JTAG 接口;

TDO: 测试数据输出, 数据通过 TDO 引脚从 JTAG 接口输出;

在 STM32F103C8T6 上其对应的 IO 口分别为 PA13、PA14、PA15、PB3, 同时需要接上 VCC 和 GND 并把 BOOT0、BOOT1 引脚接地。

SWD 是一种用于嵌入式系统调试的接口标准, 又叫做串行调试接口可以用于下载程序、调试和性能分析。SWD 接口与 JTAG 接口相比, 具有更少的引脚、更简单的协议和更高的速度。将 SWD 调试器的引脚与目标板上的 SWD 接口引脚连接。引脚包括 SWCLK、SWDIO 和地。STM32F103C8T6 对应的 IO 口为 PA13、PA14 同时需要接上 VCC 和 GND 并把 BOOT0, BOOT1 引脚接地。

串口下载通过 USART 进行程序下载, 需要 USB 转 TTL 模块使单片机与电脑相连, 模块的 TX 和 RX 引脚分别接单片机的 RX 和 TX 对应 IO 口即 PA10 和 PA9, 同时单片机的 BOOT0 引脚要接高电平, BOOT1 引脚要接低电平。

综上, 本系统采用的烧录调试方式为 SWD 下载, 下载电路设计时将 PA13(SWDIO), PA14(SWCLK), VCC 和 GND 这 4 个引脚引出, 同时单片机的 BOOT0, BOOT1 引脚要接地。下载电路如图 3-4。

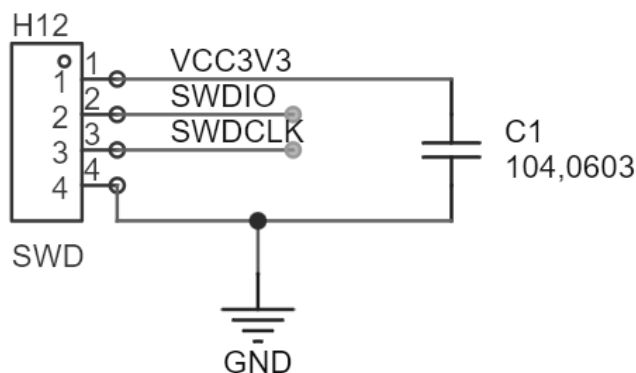


图3-4 调试下载电路

### 3.2.5 启动电路设计

如表 3-1, STM32C8T6 有 3 种启动模式可供选择, 用户可以通过设置 BOOT1 和 BOOT0 引脚的状态, 来选择在复位后的启动模式。



主闪存存储器启动模式：在此模式下，STM32 会从 Flash 存储器的地址 0x08000000 处启动。这是默认的启动模式。

系统存储器启动模式：在此模式下，STM32 会从内置的系统存储器中的地址 0x1FFF0000 处启动。这种启动模式通常用于在 Flash 存储器中出现问题时进行系统恢复或更新。

内存启动模式：在此模式下，STM32 会从内部的 SRAM 中的地址 0x20000000 处启动。这种启动模式通常用于进行芯片的调试和测试。

表3-1 启动模式选择

启动模式选择引脚		启动模式	说明
BOOT1	BOOT0		
X	0	主闪存存储器	主闪存存储器被选为启动区域
0	1	系统存储器	系统存储器被选为启动区域
1	1	内置 SRAM	内置 SRAM 被选为启动区域

启动电路如图 3-5 所示：

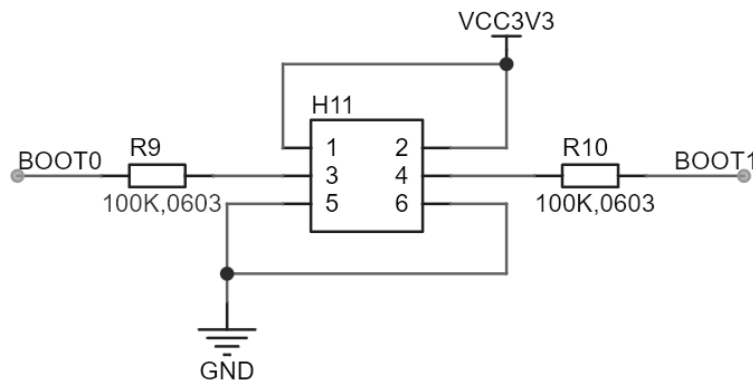


图3-5 启动电路

### 3.3 传感器模块

#### 3.3.1 温湿度检测模块

DHT11 包含四个引脚，电源引脚（VDD，GND）通常外接 3-5.5V 供电，DATA 引脚用于传感器与微处理器之间的数据传输，NC 脚一般置空。

表3-2 DHT11 引脚说明

Pin	名称	注释
1	VDD	供电 3-5.5VDC
2	DATA	串行数据，单总线
3	NC	空脚，请悬空
4	GND	接地，电源负极

DHT11 电路如图 3-6，因为传感器模块是单总线通信，所以仅需要占用单片机的 1 个 IO 口，节省了 IO 口资源。R1 为数据线上接入的上拉电阻，因为在单总线协议的特点是需要上拉电阻的支持来提供总线信号的驱动和恢复。因此，在使用 DHT11 传感器时，为了保证数据通信的稳定性，通常需要在数据总线上连接一个适当值的上拉电阻。另外，上拉电阻也可以避免总线中的信号干扰，能够减少噪声对数据传输的影响。在实际应用中，一般建议上拉电阻的取值在  $4.7\text{K}\Omega$  左右，以保证传输速率和抗干扰能力均能够得到满足。

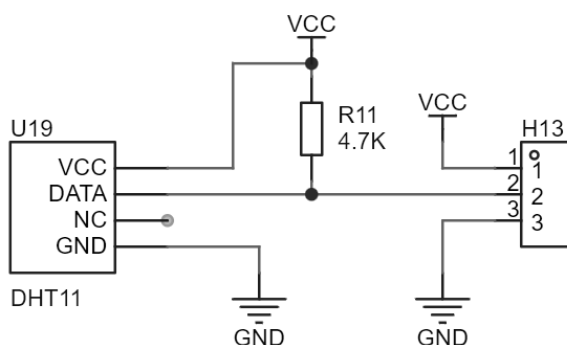


图3-6 DHT11 电路

### 3.3.2 气体浓度检测模块

MQ135 空气质量传感器模块有 4 个引脚，分别是 VCC、GND、和两个输出口 DO 和 AO，DO 引脚输出数字信号，AO 引脚输出模拟信号。该传感器需要两种电压：加热器电压( $V_H$ )和测试电压( $V_C$ )。图 3-7 是传感器的基本测试回路，其中，加热器电压( $V_H$ )用于提供传感器的工作温度，而测试电压( $V_C$ )用于测量负载电阻( $R_L$ )上的电压( $V_{RL}$ )，以便得到传感器测量出来的数据。由于该传感器具有轻微的极性，所以测试电压( $V_C$ )需要使用直流电源。在保证传感器电性能要求的前

提下，加热器电压( $V_H$ )和测试电压( $V_C$ )可以共用同一个电源电路。为了更好地利用传感器的性能，需要选择适当的电阻值( $R_L$ )。

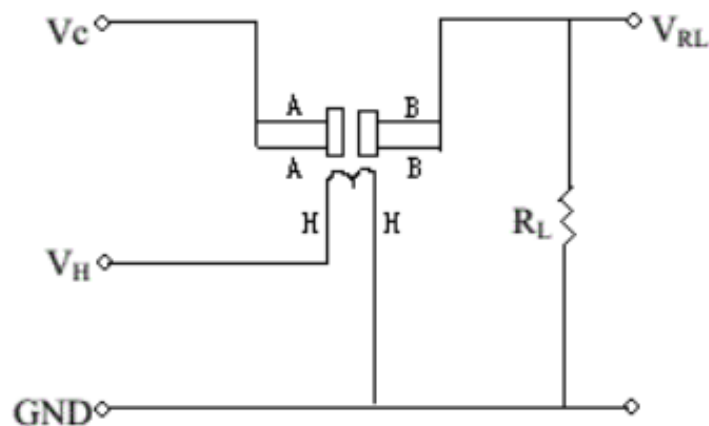


图3-7 基本测试回路

### 3.3.3 甲醛浓度检测模块

ZE08-CH<sub>2</sub>O 甲醛传感器模块支持模拟信号输出，即使用 Pin2 脚上的 DAC 直接输出模拟电压信号，该信号范围为 0~2V，可以连接到单片机的 ADC 转换引脚上，实现模拟信号的转换。此外，该模块内部还集成了数据处理电路，可以将采集到的甲醛浓度信号转换为数字信号，并通过串口（UART）发送出来。具体而言，模拟信号输出是一种直接输出模拟电压信号的方式，将传感器的模拟信号传输到其他电路中进行进一步处理。而数字信号输出则需要通过串口通信等方式将处理后的数据传输给其他设备进行处理。本系统采用的是串口输出数据的方式，方便了数据在程序上的处理。电路连接如图 3-8 所示，模块需要接入 5V 电源，并使模块的 UART 引脚与单片机的 UART 引脚相连。

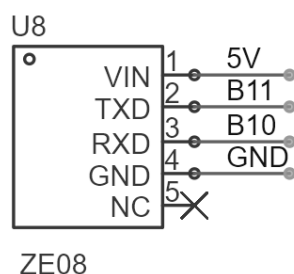


图3-8 ZE08-CH<sub>2</sub>O 电路

### 3.4 声光报警模块

声光报警模块主要由一个蜂鸣器和一个 LED 灯组成。当传感器采集数据超过设置的阈值时，声光报警模块开始工作，蜂鸣器和 LED 灯周期性的发声和闪烁提醒人们出现异常情况，及时发现报警原因。

蜂鸣器通常按驱动方式可分为有源蜂鸣器和无源蜂鸣器，无源蜂鸣器内部不带振荡源，只有提供一定频率的振荡方波脉冲，才能使蜂鸣器的振动发声，调整振荡脉冲的频率，就会发出不同的声音。有源蜂鸣器则不需要外部振荡源，只需要接入直流电源即可，直流电源输入经过振荡系统的放大取样电路在谐振装置作用下产生相对固定的声音信号<sup>[16]</sup>。在本设计中选用的是有源蜂鸣器，图 3-9 为通用有源蜂鸣器的驱动电路。使用 NPN 型三极管驱动蜂鸣器，电阻 R17 为限流电阻防止流过基极电流过大损坏三极管，R18 相当于基极的下拉电阻，C8 为电源滤波电容，滤除电源高频杂波，C9 可以在有强干扰环境下，有效的滤除干扰信号，避免蜂鸣器变音和意外发声。

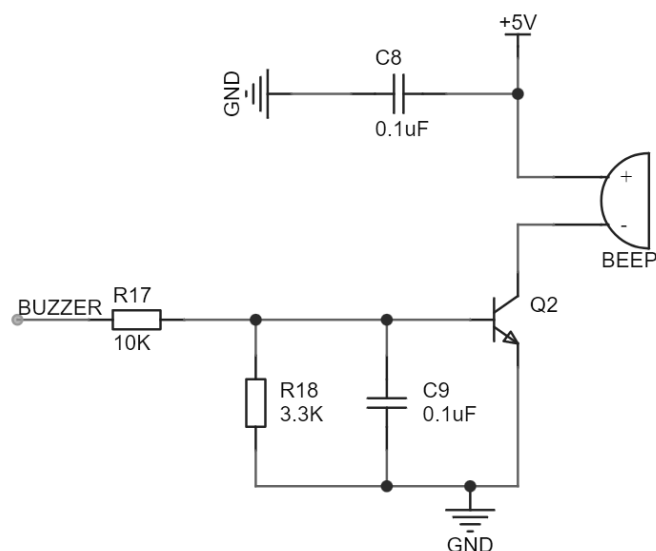


图3-9 蜂鸣器电路

LED 电路由一个发光二极管和一个限流电阻组成，正极接入到电源，负极接到单片机 IO 口上，通过程序控制 IO 口输出高低电平来控制 LED 灯的亮灭。



图3-10 LED 电路

### 3.5 供电电路设计

本系统设计了两种供电方案可供选择，第一种可以直接通过 TypeC 接口连接外部充电宝或者电脑 USB 口供电，输出 5V 电压；第二种可以通过两节 3.7V 的 18650 锂电池串联得到 7.4V 的电压，经过 LM2596 降压模块降压到 5V 使用。两种方法可以通过跳帽选择，得到的 5V 电压经过 AMS1117-3.3 稳压模块降压到 3.3V 以供单片机和部分传感器工作。供电电路如图 3-11 所示。

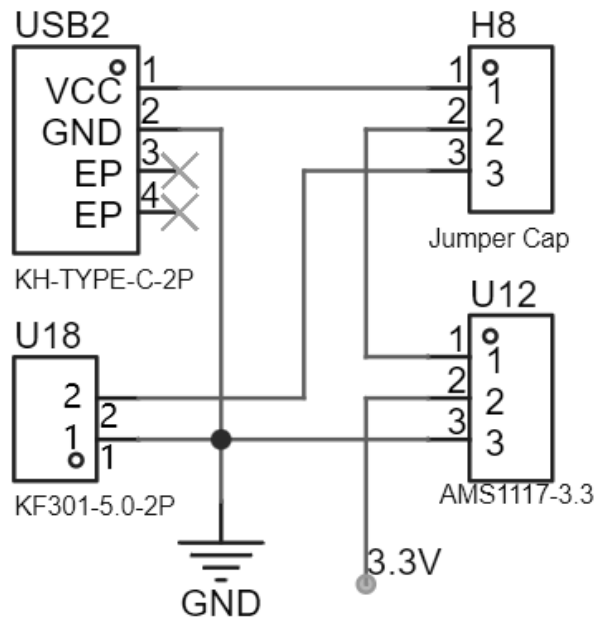


图3-11 供电电路

## 第4章 系统软件程序设计

### 4.1 主程序设计

如图，系统上电后 MCU、NB-IoT 通信模块和各传感器初始化配置相应的端口管脚，MQTT 初始化连接上云平台，各传感器采集空气环境数据，经过换算后得到具体数值，通过 NB-IoT 模块上报到云平台显示数据，云平台在把数据下发给 APP。在设备端通过 OLED 显示屏可以观察到系统进入菜单界面，菜单界面包括两个实时检测和阈值设置，通过按键选择，实时检测页面可以观察到对应传感器采集数据，阈值设置界面可以通过按键加减功能设置阈值。当数据超过阈值时，声光报警模块工作。

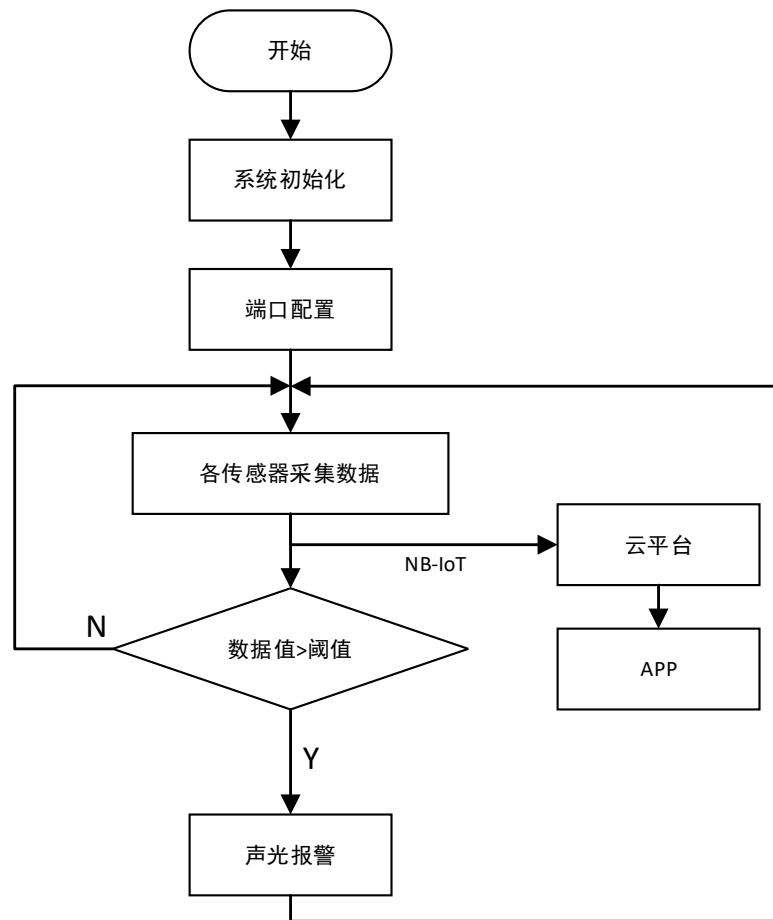


图4-1 主流程图

## 4.2 温湿度采集程序设计

DHT11 时序流程如下：

1.主机发送起始信号：主机将总线拉低至少 18 毫秒，然后将总线拉高 20~40 微秒。

2.传感器发回响应信号：DHT11 接收到起始信号后，将总线拉低 80 微秒，然后将总线拉高 80 微秒以表示响应。

3.发送数据：在响应信号后，DHT11 将发送 40 位数据。每个数据位由一个 50 微秒的低电平和一个 26-28 微秒的高电平表示。高电平的持续时间为 26-28 微秒表示逻辑“0”，为 70 微妙表示逻辑“1”。

4.数据传输结束：在发送完 40 个数据位后，DHT11 将释放总线，将总线拉高 50 微秒表示数据传输结束。

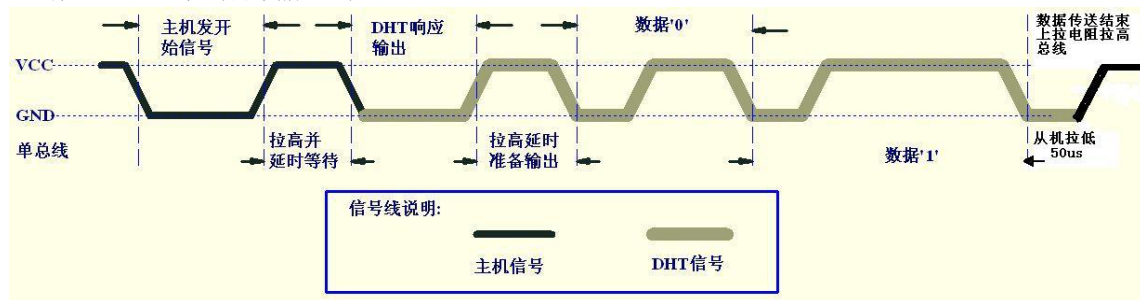


图4-2 DHT11 总时序图

初始化 GPIO 引脚为输出模式，并将引脚输出高电平，延时至少 18ms 来保证 DHT11 传感器已经启动。将引脚设置为输入模式，再次延时 20ms 以上，等待 DHT11 响应信号。读取 DHT11 响应信号，即低电平持续时间是否符合标准，如果不符合则表示响应失败，重新检测 DHT11 是否响应。如果响应成功，则读取 40 位数据，每个数据位分别由 50us 和 26-28us 的高电平时间组成。将这些数据存储到一个数组中。对 40 位数据进行校验，确保数据的正确性，如果校验失败则重新检测 DHT11 是否响应。校验成功提取湿度和温度数据。

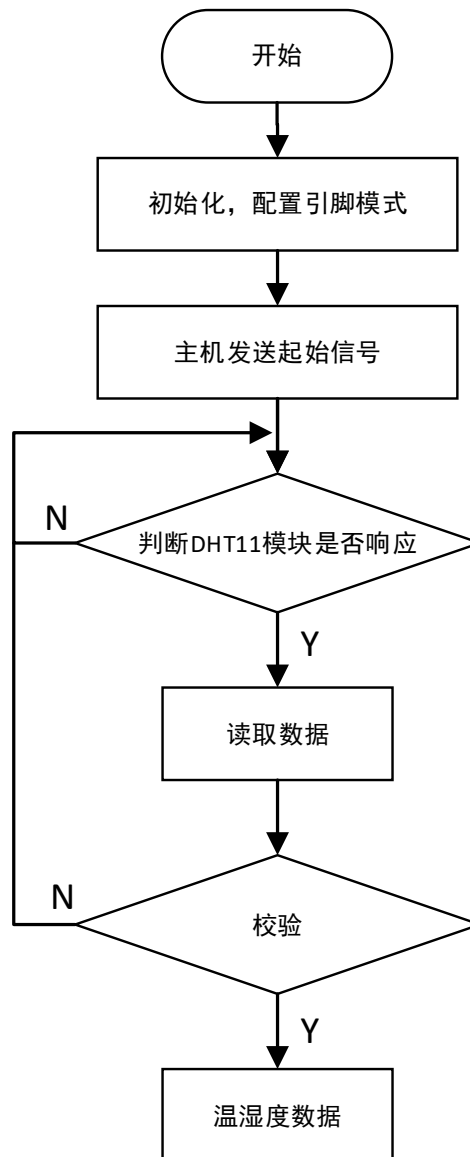


图4-3 温湿度采集程序流程图

温湿度读取程序函数如下，循环读取 40 位数据。

```
for(i=0;i<5;i++)//读取 40 位数据
```

```
{
```

```
    buf[i]=DHT11_Read_Byte();
```

```
}
```

判断数据校验和，校验和正确得到温湿度值。



```

if((buf[0]+buf[1]+buf[2]+buf[3])==buf[4])
{
    *humi=buf[0];
    *temp=buf[2];
}

```

### 4.3 气体浓度采集程序设计

如图 4-4 是传感器的灵敏度特性曲线，图中的横纵坐标分别表示的是气体浓度和传感器的电阻比 ( $R_s/R_o$ )， $R_s$  表示传感器在不同浓度气体中的电阻值， $R_o$  表示传感器在 100ppm 氨气中的电阻值。当待测气体接触到传感器表面发生化学反应，使得传感器表面的电阻  $R_s$  变化，可以计算出气体浓度。其中  $R_s$  的计算公式为：

$$R_s = (V_C / V_{RL} - 1) \times R_L \quad (4-1)$$

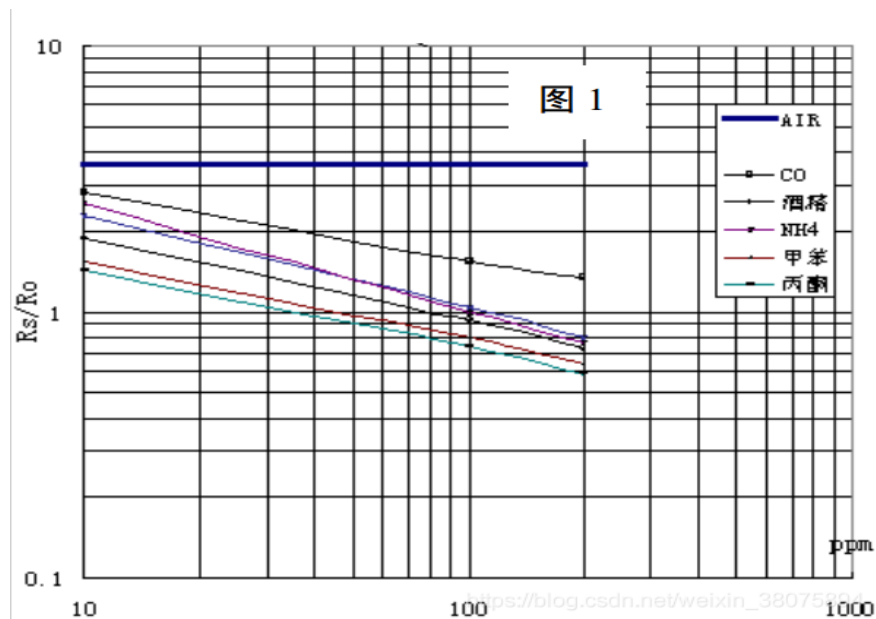


图4-4 灵敏度曲线

从甲苯曲线上采点进行拟合如图 4-5， $R_o$  取的是 10k， $R_L$  根据模块原理图是 1k， $V_{cc}$  是供电电压 5V， $V_{out}$  是输出引脚，ADC 采集得到的电压值，最终得到浓度公式：

$$\text{ppm}^{0.3203} = \frac{3.4880 \times R_o \times V_{OUT}}{(V_{CC} - V_{OUT}) \times R_L} \quad (4-2)$$

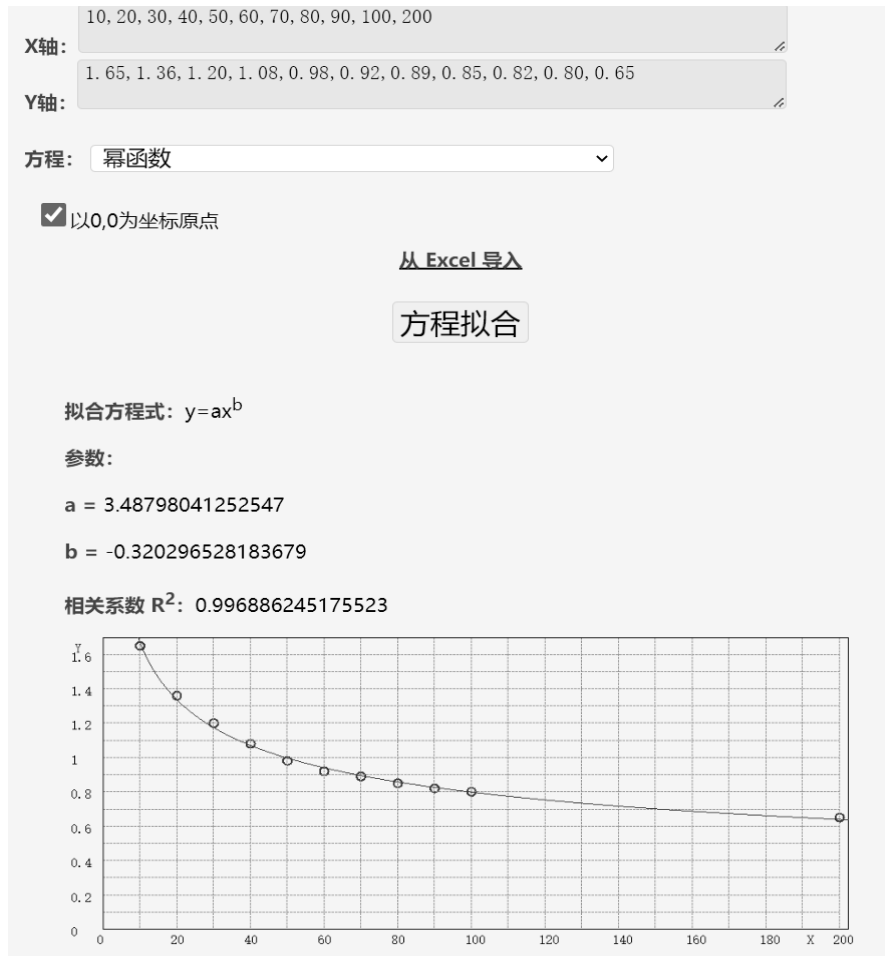


图4-5 拟合曲线图

由此可得到，气体浓度计算程序，首先将 AD 采样值换算为电压值，再计算出浓度值。

```
float vol;
```

```
float ppm;
```

```
vol = (float)ADC_Convertedvalue[0] / 4096 * 3.3;
```

```
ppm = pow((3.4880 * 10 * vol) / (5 - vol), (1.0 / 0.3203)) / 1000;
```

```
return ppm;
```

该系统采集的是气体浓度数值需要用到模拟输出口，所以首先需要配置 AO

输出引脚，将采集的数据通过模数转换计算电压数值，经过公式(4-1)和公式(4-2)计算出最终的浓度数值。同时为了减少 CPU 负担，在程序中使用 DMA（直接存储器访问）方式传输 ADC 采集的数据，它的优点在于使用 DMA 可以在不占用 CPU 时间的情况下，将 ADC 数据直接传输到内存中，这样可以减少 CPU 的负担，使其能够更多地处理其他任务，同时 DMA 可以在后台自动完成数据传输提高数据采集效率，且支持大容量数据传输从而满足较高的数据处理要求。

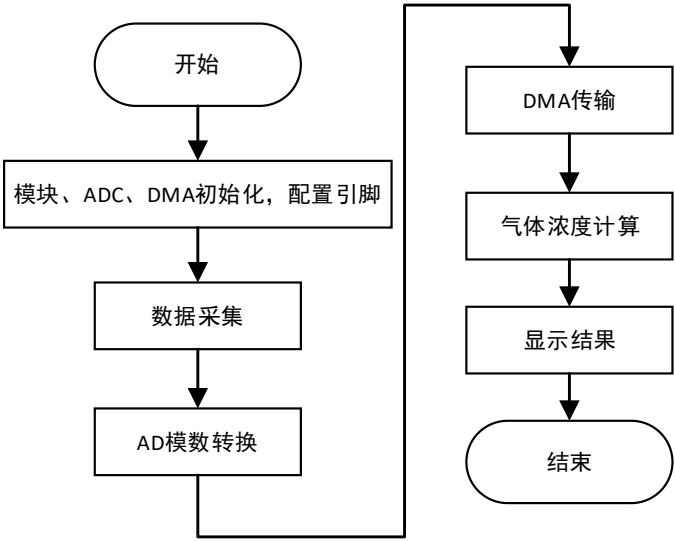


图4-6 气体浓度采集程序设计

4.4 甲醛浓度采集程序设计

ZE08-CH2O 甲醛传感器采用 UART 串口通讯，出厂波特率默认为 9600 且不可更改，所以要配置单片机的串口波特率同为 9600 才可以实现通信。该模块通信分为上传式和问答式，出厂默认为主动上传，发送浓度值的时间间隔为 1S。数据发送格式如下表 4-1。

表4-1 数据格式

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
起始位	气体名称(CH2O)	单位(ppb)	小数位数无	气体浓度低位	气体浓度高位	满量程高位	满量程低位	校验值
0xFF	0x17	0x04	0x00	0x00	0x25	0x13	0x88	0x25

采集到的气体浓度（PPB）=（气体浓度高位\*256+气体浓度低位），转换为 PPM 单位时可以用 PPB/1000 得到，1PPM 表示百万分之一，也就相当于  $1\text{mg}/\text{m}^3$ 。甲醛浓度计算程序如下，串口标志位等于 1 表示接收到数据可以进行数据换算。

```
if(flag_usart3 == 1)
{
    p_ze08 = (buff_usart3[4]*256+buff_usart3[5])/1000.0;
    memset(buff_usart3,'\0',255);
    flag_usart3 = 0;
}
```

校验和=(取反(Byte1+Byte2+……+Byte7))+1，由此可以得到校验函数如下：

```
unsigned char FucChecksum(unsigned char *i,unsigned char ln)
{
    unsigned char
    j,tempq=0 ; i+=1 ;
    for(j=0;j<(ln-2) ;j++)
    {
        tempq+=*
        i ; i++ ;
    }
    tempq= (~tempq)
    +1 ;
    return(tempq) ;
}
```

## 4.5 OLED 显示模块程序设计

该 OLED 显示模块基于 IIC 协议，IIC 协议的传输流程是主设备向总线发送一个低电平脉冲，表示启动传输，主设备通过总线向从设备发送设备地址，包括读/写位。从设备将检查地址以确定是否需要响应主设备的请求。如果从设备存在于总线上，则它会发送一个应答信号，表示已经准备好接收主设备的数据。主设备通过总线向从设备发送数据。如果从设备接收到数据，则会发送一个应答信号。如果从设备不能接收数据，则不会发送应答信号。主设备向总线发送一个高电平脉冲，表示传输结束。通过字库生成软件生成字库，调用相应函数调用字库数据写入 OLED 显示屏刷新后屏幕显示内容。

## 4.6 按键模块程序设计

首先配置连接按键的 IO 口为输入模式，配置 NVIC（中断控制器）、EXTI（外部中断线路），当按键按下的时候，引脚电平发生转变，同时触发沿触发 EXTI 中断，进而打断，使程序跳转到 EXTI 中断中执行。

按键 1：确定按键，确定选中；

按键 2：返回功能，返回菜单；

按键 3：上/加功能，向上选择，数值增加；

按键 4：下/减功能，向下选择，数值减少；

按键 1 程序如下，按键按下当系统处于菜单界面时，判断界面标志位进入对应界面；

```
if (menu_flag == 0) mode = 1;
```

```
if (menu_flag == 1) mode = 2;
```

处于阈值设置界面时改变的是确定按键标志位，用于选择取消所选项。

```
if (flag == 0) flag = 1;
```

```
else if (flag == 1) flag = 0;
```

按键 2 代码如下，如果系统此时在实时检测界面或阈值设置界面，按键按下

返回主菜单。

```
mode = 0;
```

按键 3 代码如下，在菜单界面和阈值设置界面按下都表示向上选择，如果在系统处于阈值设置界面同时确定按键标志位又为 1 表示选中所选项，此时按键功能表示增大数值。

按键为向上选择功能时：

```
menu_flag = 0;
```

```
Set_flag--;
```

```
if (Set_flag < 0) Set_flag = 0;
```

```
if(Set_flag==2) cl3=1;
```

按键为增大功能时：

```
param_t++;
```

```
if (param_t > 99) param_t = 99;
```

```
param_h++;
```

```
if (param_h > 99) param_h = 99;
```

```
am_p++;
```

```
if (param_p > 999) param_p = 999;
```

```
param_z = param_z+0.01;
```

```
if (param_z > 3) param_z = 3;
```

按键 4 代码如下，在菜单界面和阈值设置界面按下都表示向下选择，如果在系统处于阈值设置界面同时确定按键标志位又为 1 表示选中所选项，此时按键功能表示减小数值。

按键为向下选择功能时：

```
menu_flag = 1;
```

```
Set_flag++;
```

```
if (Set_flag > 3) Set_flag = 3;
```

```
if(Set_flag==3) cl3=1;
```

按键为减小功能时:

```
param_t--;
```

```
if (param_t < 0) param_t = 0;
```

```
param_h--;
```

```
if (param_h < 0) param_h = 0;
```

```
param_p--;
```

```
if (param_p < 0) param_p = 0;
```

```
param_z = param_z-0.01;
```

```
if (param_z < 0) param_z = 0;
```

## 4.7 NB-IoT 通信模块程序设计

MCU 通过串口发送数据给 BC26 模块，所以需要先进进行串口配置，成功建立串口连接后通过 AT 命令的方式发送数据，AT 命令是一种控制模拟调制解调器 (MODEM) 的指令集，它的全称是 ATtention 命令集，用于控制 MODEM 的各项功能，包括拨号、断线重拨、传输数据等等。常用的 AT 命令如表 4-2 所示。

表4-2 常用 AT 命令集

指令	功能描述
AT	测试 AT 指令功能是否正常
AT+CSQ	获取信号强度和信道误码率
AT+CEREG?	查询 EPS 网络注册状态
AT+CGATT?	查询网络是否被激活
AT+CGSN	请求产品序列号 IMEI
AT+CIMI	查询 IMSI 国际移动用户识别码
AT+CGPADDR	查询模块 IP 地址
AT+QMTCFG	配置 MQTT 可选参数
AT+QMTOPEN	打开 MQTT 客户端网络
AT+QMTCLOSE	关闭 MQTT 客户端网络
AT+QMTDISC	从 MQTT 服务器断开客户端连接
AT+QMTSUB	订阅主题
AT+QMTUNS	退订主题
AT+QMTPUB	发布消息

发送 AT 命令检查模块以及 SIM 卡是否到位，其程序如下。

```
printf("AT\r\n");
```

```
printf("AT+CSQ\r\n");
```

```
printf("AT+CPIN?\r\n");
```

```
printf("AT+QMTCLOSE=0\r\n");
```

发送 AT 命令通过 MQTT 协议连接服务器，

建立 TCP 连接：使用 AT+QMTOPEN 命令建立 MQTT 连接。

连接 MQTT 服务器：使用 AT+QMTCONN 命令连接 MQTT 服务器。

发布 MQTT 消息：使用 AT+QMTPUB 命令发布消息。

订阅 MQTT 主题：使用 AT+QMTSUB 命令订阅主题。



接收 MQTT 消息：使用 AT+QMTSUB 命令订阅主题并接收消息。

其程序如下。

```
printf("AT+CGPADDR=1\r\n");

printf("AT+QMTCFG=ALIAUTH,0,\"%s\",\"%s\",\"%s\"\r\n",ProductKey,Device
Name,DeviceSecret);

printf("AT+QMTOPEN=0,\"139.196.135.135\",1883\r\n");

printf("AT+QMTCONN=0,\"air_01\"\r\n");

printf("AT+QMTSUB=0,1,\"%s\",0\r\n",SubTopic);

printf("AT+QMTPUB=0,1,1,0,\"%s\",\"{\"sys\":{\"ack\":0},\"params\":{\"mtemp:%
d,mhumi:%d,mppm:%0.2f,mze08:%0.4f,alarm:%d,param_t:%d,param_h:%d,param_p:
%d,ppm_z:%0.2f}}\"\r\n",PubTopic,tt,hh,pp,zz,alarm,param_t,param_h,param_p,param
_z);
```

## 第5章 系统调试及展示

### 5.1 硬件调试

#### 5.1.1 PCB 电路板设计及焊接

基于前文的硬件电路设计画出 PCB 电路图如图 5-1，根据电路设计，将元器件排布在 PCB 板上，确定各个元器件的位置和摆放方向。布局方面需要考虑 PCB 板的尺寸、电源布局、信号传输线路的布线等因素，同时需要遵循一定的布局规则，如保证元器件之间的间距、信号传输线路的长度等。布线方面需要根据电路设计和 PCB 板布局，将各个元器件之间的连接线路布置在 PCB 板上。这一步需要考虑信号传输线路的宽度、长度、阻抗匹配等因素，同时需要遵循一定的布线规则，如避免串扰、交叉等。

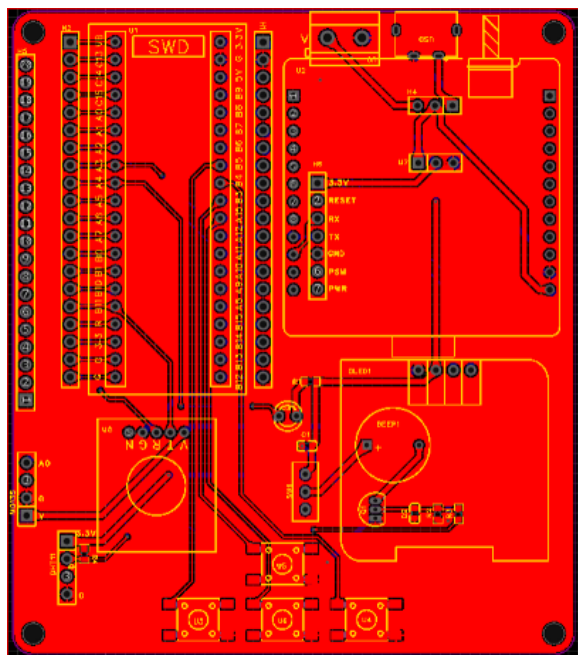


图5-1 PCB 电路图

完成 PCB 布局和布线后，进行电气规则检查和设计规则检查，确保 PCB 板的电性能和机械性能都符合要求。最后生成 PCB 制板文件，进行 PCB 板的制造，之后需要对 PCB 板进行测试，使用万用表对应电路原理图和 PCB 电路图检

查 PCB 板的电气连通性，确保电路中各个元器件之间的连通状态良好。做好准备工作后可以进行硬件焊接。电路板焊接是将电子元器件连接到电路板上的过程。通常使用的焊接方法有手工焊接和机器焊接两种。考虑到实际情况本设计使用手工焊接的方式，准备焊接工具：烙铁、焊锡丝、辅助工具（如镊子、剪刀等）以及电路板和元器件。将元器件放入对应的位置。使用烙铁加热焊接点，同时用镊子等辅助工具将焊锡丝放在焊接点上。当焊锡丝熔化时，将其与焊接点接触，使其润湿焊接点。然后停止加热，让焊锡丝冷却和凝固。下图 5-2 是焊接完成实物图展示。

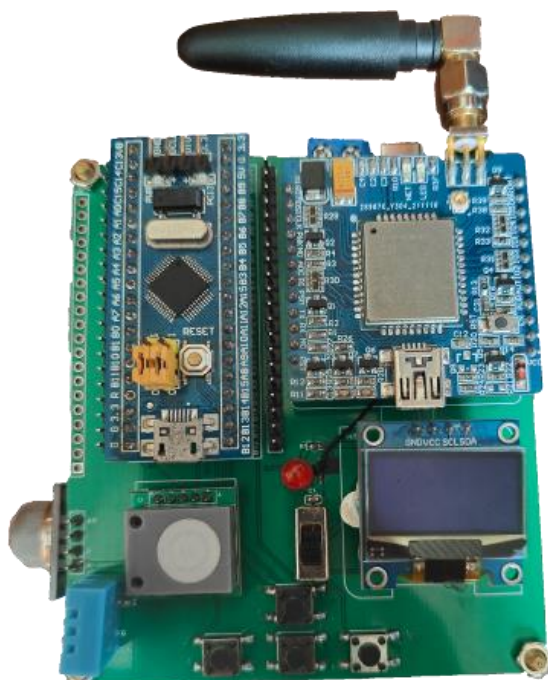


图5-2 实物图

### 5.1.2 电源电路调试

使用电压表测量供电电路，万用表调到直流电压 20V 挡位，测量 USB 供电，两表笔分别接到 Type-C 母座的 VCC 和 GND 引脚，万用表数据显示为 5V；跳帽切换到锂电池供电模式，测量锂电池供电万用表数据显示为 7.4V；测量 LMS2596 降压模块输出端口电压，调节电位器直到输出电压为 5V。测量 AMS1117-3.3V 降压模块输出端口电压万用表数据显示为 3.3V。

### 5.1.3 传感器调试

调试 DHT11 温湿度传感器，将数据展示在串口调试助手上，人呼出的气体具有一定的温度和水分，通过对传感器哈气可以直观的看到温湿度变化表示模块数据采集正常。

调试 MQ135 气体浓度传感器，将数据展示在串口调试助手上，使用打火机进行测试，打火机的燃料的主要成分是丁烷，轻微按下打火机释放出气体靠近传感器可以看到数据变化表示模块数据采集正常。

调试 ZE08-CH<sub>2</sub>O 甲醛传感器，将数据展示在串口调试助手上，因为本传感器能对除目标气体外的其他部分气体产生响应，为了方便测量调试，这里可以使用酒精靠近传感器，出现数据变化则表示传感器数据采集正常。

### 5.1.4 通信模块调试

使用 USB 转 TTL 模块将 BC26 模块的 VCC、GND、TX、RX 引脚与电脑连接，通过串口调试助手发送和接收数据。如图 5-3 发送指令后反馈“OK”，能接收到下发的命令表示通信正常。

```
OK
AT+QMTCFG=ALIAUTH,0,"a14Uk3lGTjE","air_01","c59bfec1a12104637a77cd96216f8807"
OK
AT+QMTOPEN=0,"iot-as-mqtt.cn-shanghai.aliyuncs.com",1883
OK
+QMTOPEN: 0,0
AT+QMTCONN=0,"air_01"
OK
AT+QMTSUB=0,1,"/sys/a14Uk3lGTjE/air_01/thing/service/property/set",0
ERROR
+QMTCONN: 0,0,0
AT+QMTSUB=0,1,"/sys/a14Uk3lGTjE/air_01/thing/service/property/set",0
OK
+QMTSUB: 0,1,0,1
AT+QMTSUB=0,1,1,0,"/sys/a14Uk3lGTjE/air_01/thing/event/property/post","{params:
{humidity:1}}"
OK
+QMTSUB: 0,1,0
+QMTRECV:
0,0,"/sys/a14Uk3lGTjE/air_01/thing/service/property/set","{method:thing.service.pro
perty.set,\"id\":\"907391069\", \"params\":{\"swch\":1,\"version\":\"1.0.0\"}"
```

图5-3 串口调试

## 5.2 系统配置及流程测试

### 5.2.1 云平台及 APP 配置

登录物联网云平台创建产品，配置产品连网方式、数据格式，添加设备，配置产品的功能参数包括温度、湿度、气体浓度、甲醛、报警开关。给设备通电检查是否能检测到设备。设备处于在线状态表示设备连接正常。

如图 5-4 查看日志服务，检查数据接收信息是否正常。

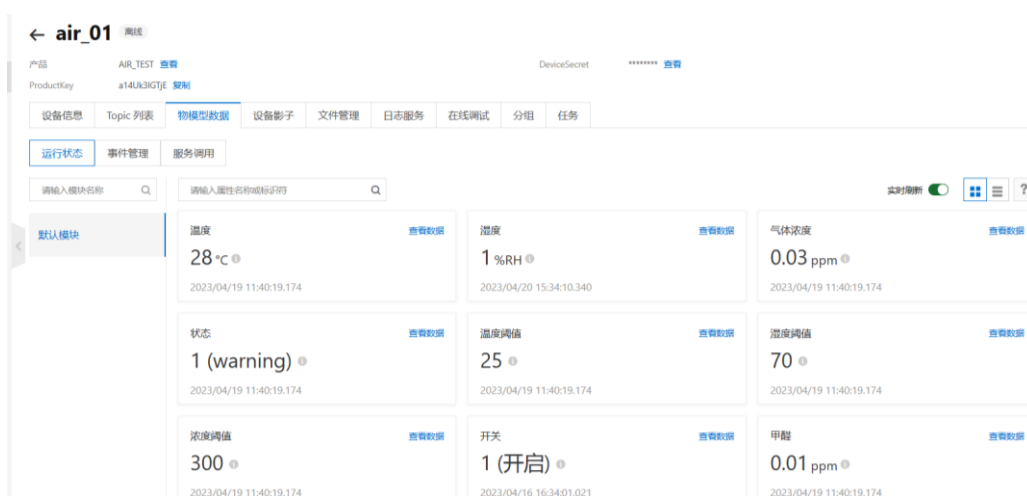


图5-4 云平台设备详情图

如图 5-5 基于云智能 APP 添加产品项目，配置面板界面。

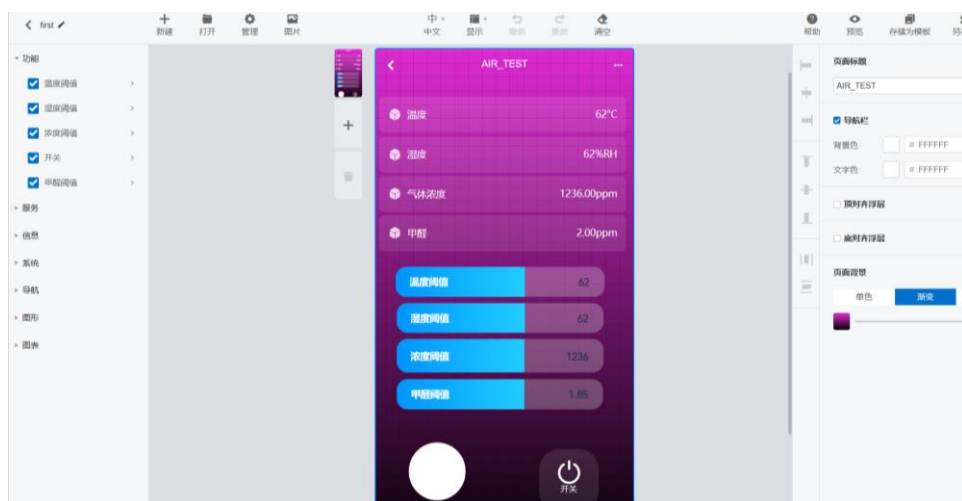


图5-5 APP 界面

### 5.2.2 系统测试

如图 5-6 系统上电后首先初始化使用到的外设以及各传感器模块，配置引脚端口，同时检测 NB 通信模块以及物联网卡是否到位并在屏幕上显示初始化状态。



图5-6 设备初始化

如图 5-7 初始化完成建立好通信连接后系统进入菜单界面，菜单界面包含 3 个选项分别为实时检测、阈值设置和模式，通过按键进行选择。



图5-7 菜单

如图 5-8 选择实时检测界面，按确定键进入可以观察到实时的温度、湿度、气体浓度、甲醛浓度的具体数值。



图5-8 实时检测

如图 5-9 选择阈值设置界面，界面显示为温度阈值、湿度阈值、气体浓度阈值、甲醛浓度阈值，选择对应的阈值参数按下确认键表示选中，上下键调整阈值，再次按下确认键表示设定该阈值退出设置。当有检测值超出其对应的阈值时，系统声光报警提醒。

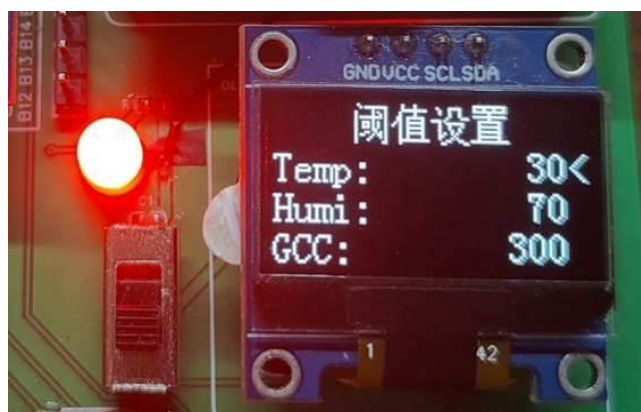


图5-9 阈值设置

如图 5-10 选择模式选项，通过按下确定键可以切换模式为自动模式或手动模式，在自动模式下，当有检测值超出其对应的阈值时会自动打开继电器，在手动模式下可以通过手机 APP 控制继电器开关与否，继电器可以外接加温、加湿、排气扇等设备，具体所接设备可以依据使用情景和需求而定。



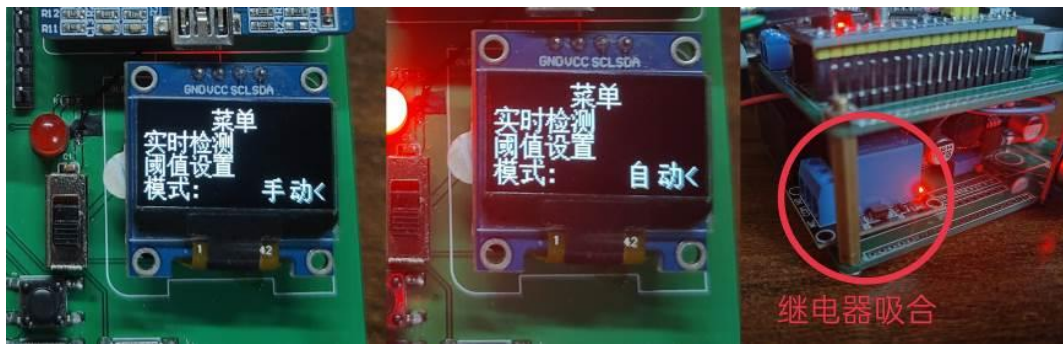


图5-10 模式切换

综上，经过实验后，结果表明该系统能够较为准确地测量温度、湿度、气体浓度、甲醛浓度的含量值，在超过阈值后能及时报警，能及时将数据及异常报警状态反馈给云端上位机。经实践测量后发现，该系统的达到了预期的效果。虽然还存在一些需要改进的地方，但总体来说，该系统已经初步具备了实际应用的能力。



## 第6章 结论

本文介绍了一种基于 NB-IoT 的空气质量检测系统设计。该系统采用 STM32F103C8T6 微控制器结合 NB-IoT 模块和各类传感器实现了数据传输和远程监控。系统采用了温湿度传感器、气体浓度传感器、甲醛传感器等，实现对空气质量的多方位检测。在硬件设计方面，本文详细介绍了硬件电路的设计和实现。在软件设计方面，本文介绍了系统的软件架构和程序设计，包括数据采集和处理、数据上传和远程监控等。实验结果表明，该系统能够准确地检测空气质量，并实现了数据的远程传输和监控。该系统具有实时性、可靠性和稳定性，可以广泛应用于室内和室外空气质量监测领域。最后经过整个系统设计的完整实现得出以下总结：

（1）通过空气质量检测，可以及时发现和监测空气质量，采取相应的措施来减少和防止空气污染。同时，空气质量检测也可以为环境保护和可持续发展提供数据支持和科学依据。

（2）在系统设计中同时存在许多待优化的地方，比如传感器数据采集不够准确不够灵敏后期可以通过更换更为专业的传感器来解决。

未来的工作将进一步优化系统的设计和性能，包括优化数据采集和处理算法、提高传感器的精度和准确性、优化数据传输和远程监控等。同时，将探索更广泛的应用场景，如城市环境监测、健康管理等，为人们提供更加健康的生活环境。

## 参考文献

- [1]世界卫生组织. 空气质量和健康 [EB/OL].(2022-12-19)[2023-04-12].  
[https://www.who.int/zh/news-room/fact-sheets/detail/ambient-\(outdoor\)-air-quality-and-health](https://www.who.int/zh/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health).
- [2]世界卫生组织. 世卫组织新数据:数十亿人仍呼吸着不健康的空气 [EB/OL].(2022-04-04)[2023-04-12].<https://www.who.int/zh/news/item/04-04-2022-billions-of-people-still-breathe-unhealthy-air-new-who-data>.
- [3]杨亚男,袁英.智能灌溉控制系统的设计[J].科技资讯,2023,21(04):93-96+157.
- [4]闫天瑞,余倩倩,陈新威,李抒智,张红,王江,邹军,石明明,王洪荣,苏晓峰,陈启,郑炜陵.基于窄带物联网 (NB-IoT) 技术在智慧机场水质监测系统中的应用分析 [J].应用技术学报,2022,22(04):395-398.
- [5]王宜朋.基于物联网的智能孵化设备设计[J].电子制作,2022,30(05):19-21+94.
- [6]刘武军,黄靖,欧立涛.基于 WM-BUS 的高精度智能止损超声波水表设计 [J].给水排水,2021,57(12):141-146.
- [7]吴铁军.基于 NB-IoT 的燃气管道气密性检测仪设计[J].仪器仪表用户,2023,30(02):1-4.
- [8]刘学武.NB-IoT 无线网络优化的特点及方法[J].物联网技术,2022,12(11):48-50+54.
- [9]杨镐辰.基于 FreeRTOS 的嵌入式设备管理关键技术研究及实现[D].北京邮电大学,2021.9-10.
- [10]曾涌泉,何浩,赵中军.基于 NB-IoT 的士兵训练无线终端系统的设计 [J].通信技术,2021,54(01):238-243.
- [11]何翔,卓建业.无线运动传感器系统设计[J].现代电子技术,2023,46(04):25-29.
- [12]虞礼辉,张世荣,黎贤钦,王汪洋,包日官,王捷.变压器油水冷却器 5G 物联管控系统设计[J].变压器,2023,60(04):19-24.
- [13]张小莉,何艳.基于物联网技术的能源数据服务平台[J].电气传动自动化,2023,45(02):59-63.
- [14]康晓辉.智能家居网关软件系统设计与实现[D].北京邮电大学,2020.
- [15]徐微,王天赐,郝琦琦,李守智.基于 NB-IoT 的直饮水检测系统 [J].自动化与仪器仪表,2023(03):209-212.
- [16]乐海文,刘文谦,刘对,高朋.基于 GSM 的家庭智能防护系统设计[J].电子质量,2022(07):96-100.

## 致谢

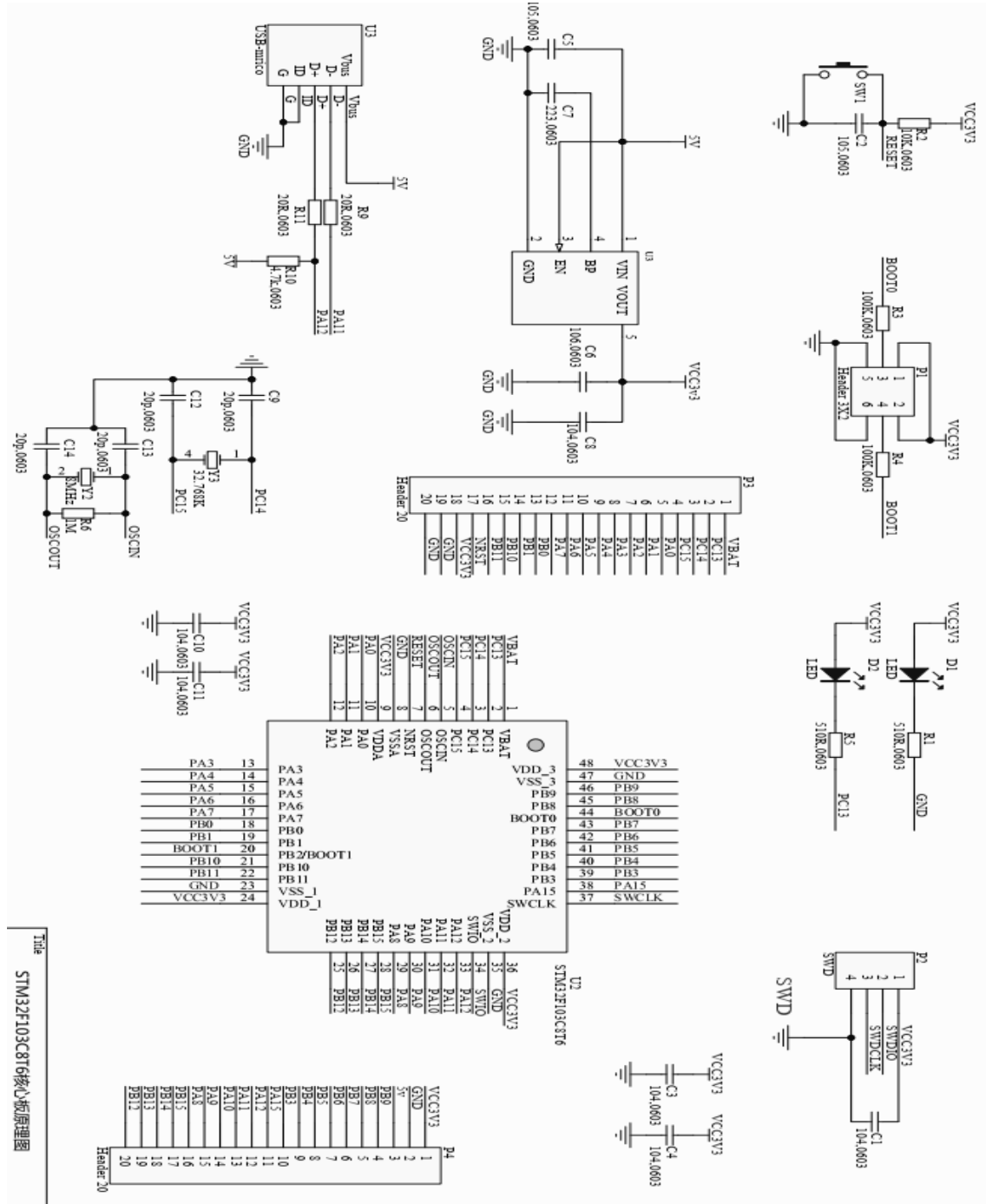
在本次毕业论文完成之际，我感到无比激动和欣喜，同时也对四年飞逝的大学时光而万分感慨，在大学四年的学习生涯中，我们度过了许多难忘的时刻。从刚入学的懵懂和新鲜，到大二、大三的努力拼搏，再到大四前途未卜的迷茫和压力，我们经历了成长的痛苦，也收获了人生中无价的财富。而这些离不开在背后默默支持着我的一群人，我想向他们表达我由衷的感激和深深的祝福。

首先，我要感谢我的指导老师李丹老师，感谢她给予我的指导和帮助。在选题和思路的确定、论文内容的拟定、实验方案的设计以及论文的修改和完善等方面，她都给予了我细致耐心的指导和建议。她的言传身教、治学态度、严谨的科学精神对我今后的生涯将产生不可磨灭的影响。我还要感谢一路以来支持和关心我的家人、朋友和其他支持我的人。他们的精神上的鼓励 and 物质上的帮助让我在学业征途中始终保持信心和勇气，克服困难和挑战。我也要感谢我的同窗好友们，在这四年的成长历程中和我一起陪伴和支持。感谢你们在论文写作过程中无私地给予帮助和支持，让我更深入地了解研究方向并帮助我不断完善我的论文，收获了诸多宝贵的经验和知识。

最后，我衷心的祝愿老师同学们在未来的道路上光明无限，前程似锦，事业蒸蒸日上，充满幸福和成功。

## 附录

### 附录一：开发板原理图



## 附录二：程序代码

```

int main(void)
{
    delay_init();    //延时函数初始化
    ADCInit();       //ADC 初始化
    Led_GPIO_Config(); //LED IO 口配置
    Buzzer_GPIO_Config(); //蜂鸣器 IO 口配置
    Relay_GPIO_Config(); //继电器 IO 口配置
    BASIC_TIM_Init(); //定时器初始化
    OLED_Init();     //OLED 初始化
    OLED_DisplayTurn(0); //屏幕旋转 180 度
    OLED_ColorTurn(0); //反显函数
    EXTI_Key_Config(); //按键初始化
    NVIC_Configuration(); //设置 NVIC 中断分组 2:2 位抢占优先级，2 位响应
    优先级
    uart1_init(9600);    //串口 1 初始化，可连接 PC 进行打印模块返回数据
    PA9、PA10
    uart2_init(9600);    //初始化和 BC26 连接串口 PA2、PA3
    uart3_init(9600);
    OPEN_Show();        //开机画面
    TIM_Cmd(BASIC_TIM, ENABLE); //使能计数器
    while (1)
    {
        //*****

```

```
DATA_GET();

Recv_Data();//使用 strstr 函数判断传回的数据

if (mode == 0)
{
    Menushow();
}

else if (mode == 1)
{
    TimeSow();
}

else if (mode == 2)
{
    ValueSet();
}

Compare_param();

Swch_ctrl();

Send_data();

//*****

}

}

void OPEN_Show(void)
{
    //-----空气质量检测系统-----

    OLED_ShowChinese(0, 0, 0, 16, 1);
```

---

```
OLED_ShowChinese(16, 0, 1, 16, 1);
OLED_ShowChinese(32, 0, 2, 16, 1);
OLED_ShowChinese(48, 0, 3, 16, 1);
OLED_ShowChinese(64, 0, 4, 16, 1);
OLED_ShowChinese(80, 0, 5, 16, 1);
OLED_ShowChinese(96, 0, 6, 16, 1);
OLED_ShowChinese(112, 0, 7, 16, 1);
//-----
OLED_Refresh();
OLED_ShowString(0, 20, "AT init ...", 16, 1);
OLED_Refresh();
AT_Init(); //模块初始化
OLED_ShowString(0, 20, "AT init OK ", 16, 1);
OLED_Refresh();
OLED_ShowString(0, 40, "MQTT init ...", 16, 1);
OLED_Refresh();
MQTT_Init(); //MQTT 初始化
OLED_ShowString(0, 40, "MQTT init OK ", 16, 1);
OLED_Refresh();
delay_ms(500);
OLED_Clear();
OLED_Refresh();
}
```

```
void Menushow(void)
{
    if (cl == 1)
    {
        OLED_Clear(); // 清屏
        cl = 0;
    }
    /*菜单界面*/
    OLED_ShowChinese(48, 0, 19, 16, 1); // 菜
    OLED_ShowChinese(64, 0, 20, 16, 1); // 单
    OLED_ShowChinese(0, 16, 21, 16, 1); // 实时检测
    OLED_ShowChinese(16, 16, 22, 16, 1);
    OLED_ShowChinese(32, 16, 23, 16, 1);
    OLED_ShowChinese(48, 16, 24, 16, 1);
    OLED_ShowChinese(0, 32, 15, 16, 1); // 阈值设置
    OLED_ShowChinese(16, 32, 16, 16, 1);
    OLED_ShowChinese(32, 32, 17, 16, 1);
    OLED_ShowChinese(48, 32, 18, 16, 1);
    OLED_ShowChinese(0, 48, 25, 16, 1); // 模式
    OLED_ShowChinese(16, 48, 26, 16, 1);
    OLED_ShowString(32, 48, ":", 16, 1);
    OLED_ShowChinese(103, 48, 29, 16, 1);
    OLED_Refresh();
```



```
if (menu_flag == 1)
{
    OLED_ShowString(119, 16, "<", 16, 1);
    OLED_ShowString(119, 32, " ", 16, 1);
    OLED_ShowString(119, 48, " ", 16, 1);
    OLED_Refresh();
}
else if (menu_flag == 2)
{
    OLED_ShowString(119, 16, " ", 16, 1);
    OLED_ShowString(119, 32, "<", 16, 1);
    OLED_ShowString(119, 48, " ", 16, 1);
    OLED_Refresh();
}
else if (menu_flag == 3)
{
    OLED_ShowString(119, 16, " ", 16, 1);
    OLED_ShowString(119, 32, " ", 16, 1);
    OLED_ShowString(119, 48, "<", 16, 1);
    OLED_Refresh();
}

    if(relay_flag == 0)
    {
        OLED_ShowChinese(83, 48, 27, 16, 1); //自
```

```
        OLED_Refresh();
    }
    if(relay_flag == 1)
    {
        OLED_ShowChinese(83, 48, 28, 16, 1); //手
        OLED_Refresh();
    }
}

void TimeSow(void)
{
    if (cl == 1)
    {
        OLED_Clear(); // 清屏
        cl = 0;
    }

    OLED_ShowChinese(32, 0, 21, 16, 1); // 实时检测
    OLED_ShowChinese(48, 0, 22, 16, 1);
    OLED_ShowChinese(64, 0, 23, 16, 1);
    OLED_ShowChinese(80, 0, 24, 16, 1);
    //-----温度:湿度:气体浓度:甲醛:-----
    OLED_ShowString(0, 17, "Temp:", 12, 1);
    OLED_ShowString(43, 17, "`C;", 12, 1);
    OLED_ShowString(72, 17, "Humi:", 12, 1);
    OLED_ShowString(121, 17, "%", 12, 1);
```

```
OLED_ShowString(0, 30, "GCC:", 12, 1);  
OLED_ShowString(0, 43, "CH2O:", 12, 1);  
sprintf((char *)string, "%d", temp); // 将浮点数转换为字符  
OLED_ShowString(30, 17, string, 12, 1);  
sprintf((char *)string, "%d", humi); // 将浮点数转换为字符  
OLED_ShowString(102, 17, string, 12, 1);  
sprintf((char *)string, "%0.2f PPM      ", ppm);  
OLED_ShowString(25, 30, string, 12, 1);  
sprintf((char *)string, "%0.4f PPM      ", p_ze08);  
OLED_ShowString(30, 43, string, 12, 1);  
OLED_Refresh();  
}
```

```
void ValueSet(void)  
{  
    if (cl == 1)  
    {  
        OLED_Clear(); // 清屏  
        cl = 0;  
    }  
    if(cl3==1)  
    {  
        OLED_Clear();  
        cl3=0;  
    }  
}
```

```
}

/*阈值设置*/

OLED_ShowChinese(32, 0, 15, 16, 1);

OLED_ShowChinese(48, 0, 16, 16, 1);

OLED_ShowChinese(64, 0, 17, 16, 1);

OLED_ShowChinese(80, 0, 18, 16, 1);


if(Set_flag!=3)
{
    OLED_ShowString(0, 18, "Temp:", 16, 1);
    OLED_ShowString(0, 34, "Humi:", 16, 1);
    OLED_ShowString(0, 50, "GCC:", 16, 1);
    OLED_Refresh();

    if (Set_flag == 0)
    {
        OLED_ShowString(119, 48, " ", 16, 1);
        OLED_ShowString(119, 32, " ", 16, 1);
        OLED_ShowString(119, 16, "<", 16, 1);
    }
    else if (Set_flag == 1)
    {
        OLED_ShowString(119, 16, " ", 16, 1);
        OLED_ShowString(119, 48, " ", 16, 1);
    }
}
```

---

```
OLED_ShowString(119, 32, "<", 16, 1);
}
else if (Set_flag == 2)
{
    OLED_ShowString(119, 32, " ", 16, 1);
    OLED_ShowString(119, 16, " ", 16, 1);
    OLED_ShowString(119, 48, "<", 16, 1);
}
sprintf((char *)string, "%2d", param_t); // 将浮点数转换为字符
OLED_ShowString(102, 16, string, 16, 1);

sprintf((char *)string, "%2d", param_h); // 将浮点数转换为字符
OLED_ShowString(102, 32, string, 16, 1);

sprintf((char *)string, "%3d", param_p); // 将浮点数转换为字符
OLED_ShowString(94, 48, string, 16, 1);
OLED_Refresh();
}

else if (Set_flag == 3) // 翻页
{
    OLED_ShowString(0, 18, "CH2O:", 16, 1);
    OLED_ShowString(119, 16, "<", 16, 1);
    sprintf((char *)string, "%0.2f", param_z);
```

```

        OLED_ShowString(87, 16, string, 16, 1);

        OLED_Refresh();

    }

}

void DATA_GET(void)

{

    ppm = Read_Mq135_Data();//获取气体浓度值
    // p_CO = Read_Mq7_Data();

    DHT11_Read_Data(&temp, &humi);//采集湿度
    // delay_ms(50);

    if(flag_usart3 == 1)//串口接收中断 3 标志位 接受 ZE08——CH2O 传来的
数据
    {

        p_ze08 = (buff_usart3[4]*256+buff_usart3[5])/1000.0;//单位面积
内的浓度值

        //          p_ze08 = (buff_usart3[4]*256 + buff_usart3[5])/66.64 * 0.08;

        //          USART1_printf("ppm:%f\r\n",p_ze08);//这里是测试语句 使用
串口 1 发送

        memset(buff_usart3, '\0', 255);

        flag_usart3 = 0;//清除串口 2 接受中断

    }

}

void Compare_param(void)

{

```

---

```
if((temp > param_t)||(humi > param_h)||(ppm > param_p)||(p_ze08 > param_z))
{
    if(count%5==0)//警报频率
    {
        Alarm_Moudle_OPEN();
    }
}
else
{
    Alarm_Moudle_CLOSE();
}
}

void Send_data(void)
{
    /*******定时发送数据*****

    if(count==8)
    {
        count=0;

        TIM_Cmd(BASIC_TIM,DISABLE);

        SendMessage(temp,humi,ppm,p_ze08,alarm,param_t,param_h,param_p,param_z);

        delay_ms(100);

        TIM_Cmd(BASIC_TIM,ENABLE);
    }
```

```
}
```

```
void Swch_ctrl(void)
```

```
{
```

```
    if(relay_flag==1)
```

```
    {
```

```
        LED1(1);
```

```
        if(swch_flag==1)
```

```
        {
```

```
            Relay(1);
```

```
        }
```

```
        else if(swch_flag==0)
```

```
        {
```

```
            Relay(0);
```

```
        }
```

```
    }
```

```
    else if(relay_flag==0)
```

```
    {
```

```
        LED1(0);
```

```
    }
```



## 外文资料原文

### **Design o fair quality detection system based on bluetooth communication**

Qidong Ling, Gang Wang and Ming Wang

#### **1.Introduction**

With the occurrence of severe atmospheric pollution phenomena such as smog, people are paying more and more attention to atmospheric environmental pollution, hoping to be able to grasp the specific data of atmospheric pollution in the monitored living environment in time, so as to rationally arrange and carry out related activities, and reduce the harm as much as possible caused by air pollution . The national environmental monitoring department publishes regional air quality in real- time, but due to the limited number of monitoring points, fixed locations, and limited monitoring data, it cannot accurately display the air quality of the user's current location in real-time, and cannot meet the needs of ordinary users for air quality monitoring. In order to solve the above problems, an air quality detection system based on the Arduino controller is proposed, which can upload data to the mobile phone APP in real-time through Bluetooth communication to realize the monitoring of common indoor and outdoor air pollutants. The system can accurately detect and display the air quality information of the user's current location and upload the data to the mobile terminal.

#### **2.System function and structure**

The system structure of the portable air quality detection system is shown in Figure 1, which includes the microcontroller, sensor module, LCD display module, alarm reminder module, communication module and other parts. The controller chooses the Arduino nano control platform. Arduino Nano is a development platform based on ATmega328P. It has 14 digital IO pins and 6 analog input pins. It can virtualize multiple soft serial ports and can realize multiple analog and digital sensor Access. The Arduino

micro-control platform processes and analyzes the air pollutant signals collected by the air sensor in real-time, and sends the processing results to the LCD display module and mobile phone terminal for display. At the same time, the processing results are compared with the preset threshold in real-time, and the system sends a warning signal to the alarm reminding module when the threshold is preset. The system can accurately detect and display the air quality of the user's current location, and upload the data to the mobile terminal. The sensor module mainly includes PM2.5 acquisition circuit, carbon monoxide acquisition circuit, formaldehyde acquisition circuit and temperature and humidity acquisition circuit.

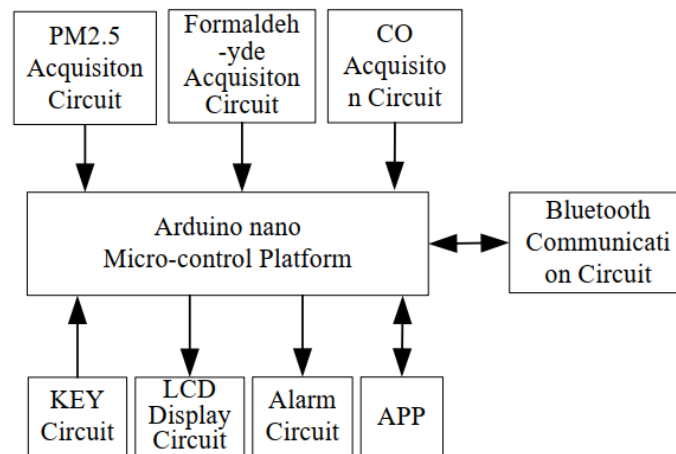


Figure1.System structure diagram

### 3.System development and design

#### 3.1 PM2.5 acquisition module design

The PM2.5 acquisition module uses the GP2Y1014AU sensor. GP2Y1014AU is an optical dust monitoring sensor module developed by Sharp. Air can flow freely through the hole in the middle of the module. Infrared LEDs and phototransistors are placed at the corners of the hole. Infrared light-emitting diode sends infrared light directionally. When particles in the air block infrared rays, the infrared sends diffuse reflection; the phototransistor receives infrared, and the voltage at the signal output pin changes. The PM2.5 acquisition module circuit is shown in Figure 2.

The LED pin of the GP2Y1014AU sensor module is connected in series with a 150Ω resistor to limit the current supply to protect the light-emitting diode. The positive and

negative poles of the light-emitting diode are connected in parallel with a 220uF capacitor to achieve a stable power supply. The LED pin is a pulse input pin and is connected to the Arduino Nano control module. The D2 pin is connected to provide an input signal for the sensor and controls the LED light inside the GP2Y1014AU sensor to turn on and off at a period of 1ms. Since the GP2Y1014AU sensor requires 5V power supply, and the maximum AD sampling voltage of the MCU is 3.3V, the signal output Vo pin needs to be divided through two resistors R23 and R24. The Vled, LED\_GND, and LED-pins of the sensor control the internal LED lights. The vent in the center allows air to flow freely, and the light refracted by the dust in the air is detected to determine the content of dust. The analog voltage output from the Vo pin of the sensor in proportion to the measured dust concentration, the analog voltage output by the sensor is converted into a digital signal by AD conversion processing.

### 3.2 CO acquisition module design

The carbon monoxide sensor uses the ZE16-CO general-purpose gas module. ZE16-CO uses electrochemical principles to detect carbon monoxide presenting in the air. The built-in temperature sensor can perform temperature compensation and has good selectivity and stability. Through pin5 and pin6, it is connected with the soft serial port pin D10 and D11 pin of Arduino nano control module. Based on the serial port protocol, the collected carbon monoxide data is sent to Arduino nano control module in 1s cycle.

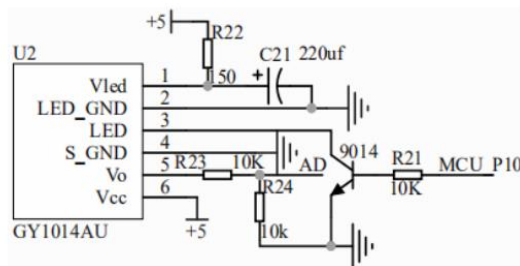


Figure 2. PM2.5 acquisition circuit design

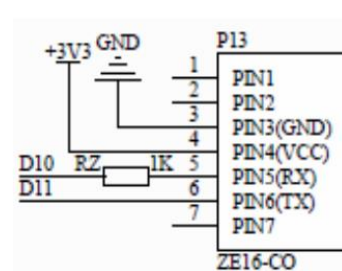


Figure 3. Co acquisition circuit design

### 3.3 Formaldehyde and CO2 acquisition module design

The formaldehyde acquisition module uses Adafruit's SGP30 sensor. SGP30 is a digital multi-pixel gas sensor. It uses temperature-controlled micro-heating plates and two pretreatment indoor air quality signals, which provide information on formaldehyde and carbon dioxide in the air. The formaldehyde acquisition circuit diagram is shown in

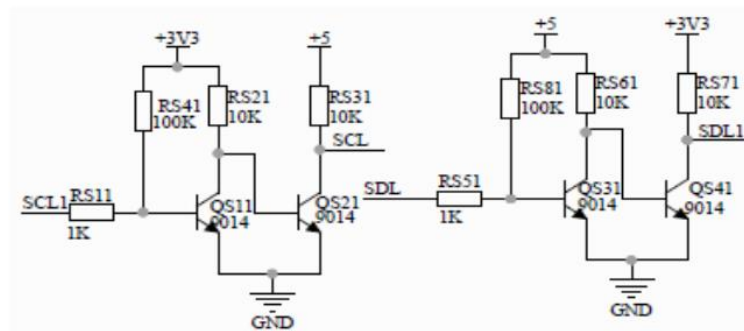
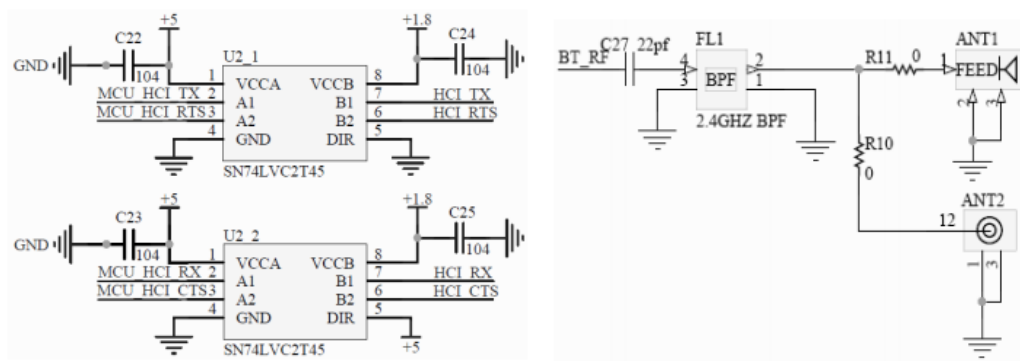


Figure 4. Formaldehyde acquisition circuit design

### 3.4 The bluetooth communication module design

The radio frequency circuit of CC2564 is shown in Figure 5(b). The BT\_RF pin emits a single-ended radio frequency signal. Its physical baseband clock is generated by a low-frequency crystal oscillator through PLL phase-locked loop frequency multiplication. The internal power amplifier provides amplification for the signal. FL1 is a Bluetooth filter, and C127 is used to match the impedance of the onboard antenna. R29 and R30 are zero ohm resistors, which are used to select the onboard antenna or external antenna. The transmitting power can reach 10dBm, the receiving sensitivity is -95dbm, and it has good RF performance.



(a) The interface circuit

(b) The radio frequency circuit

Figure 5. Bluetooth communication system design

#### 4.The design of the App

In order to facilitate the storage and processing of the data detected by the device, an APP based on the Android system is designed. The data detected by the detection device is transmitted to the mobile phone APP through Bluetooth communication in real-time. The APP stores the data in the SQLite database and synchronizes it in the active interface in real-time, which could be displayed and queried.

First, you need to declare permissions in the registration file AndroidManifest.xml: BLUETOOTH\_ADMIN permission, BLUETOOTH permission, ACCESS\_COARSE\_LOCATION permission, which is used to start the device, connect request, and dynamically obtain location permission. The design uses the Bluetooth adapter class to get the Bluetooth adapter object and enable the Bluetooth function, and scans the surrounding Bluetooth, registers for broadcast, then accepts the scan results through BroadcastReceiver, and pairs the pre-bound Bluetooth device. At the same time, the

binding address can be added to the new Bluetooth device after the pin code pairing verification. After the pairing is successful, the socket is connected to the device and the Bluetooth connection thread is opened. The connection is established based on the SPP serial communication protocol through the UUID of Bluetooth. After the connection is successful, the data transmission thread starts to receive the data sent by the hardware device and saves it to the SQLite database, design historical information query function using the ListView control, show the trend of air quality in the environment through the chart.

## **5.The system test and data analysis**

### **5.1 The system test**

The system adopts the embedded design method, completes the hardware test system, and develops the mobile APP based on the Android system. The air quality is collected by the microcontroller Arduino Nano and the sensor, and the Bluetooth communication transmits the collected data to the mobile phone APP to realize display of the air quality data storage, alarm and other functions, so as to complete the real-time monitoring of the air quality and the detection of the air quality. The liquid crystal display can achieve PM2.5, carbon monoxide, formaldehyde, temperature and humidity information. The display effect of the APP is shown in Figure 6. After opening the APP, Bluetooth module detection is carried out first, then Bluetooth is searched and connected. After successful connection, the page will display the connect page. After connecting Bluetooth module successfully, the current air temperature, humidity, PM2.5, CO (carbon monoxide) and TVOC (formaldehyde) values detected by the hardware system in the current air will be received in turn. When the value is greater than the predetermined value Value, there will be a buzzer alarm and an alarm prompt dialog box will pop up on the app.

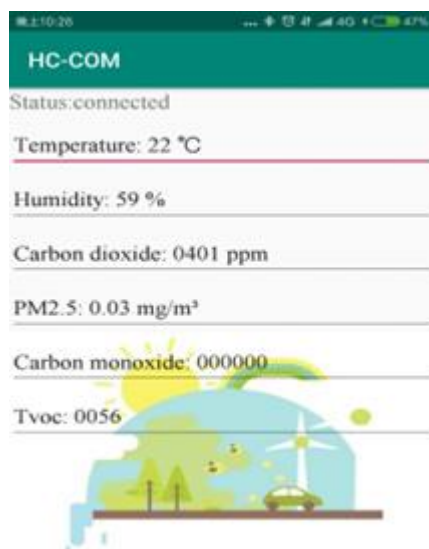


Figure 6. App test rendering

## 5.2 Test performance analysis

Select the outdoor PM2.5 detection data near the Huaita monitoring point within one day of September 23, 2019 for analysis, and calculate the average value of the monitoring data of the portable air quality monitoring system in two-hour cycles and the environmental monitoring station Huaita monitoring point. Comparison with published data, the analysis results are shown in Figure 7, The monitoring detection value of the monitoring system tends to be consistent with the published value of the monitoring point of the environmental monitoring station, and the deviation is basically kept within 5%, which can more accurately indicate the concentration of atmospheric pollutants in current location.

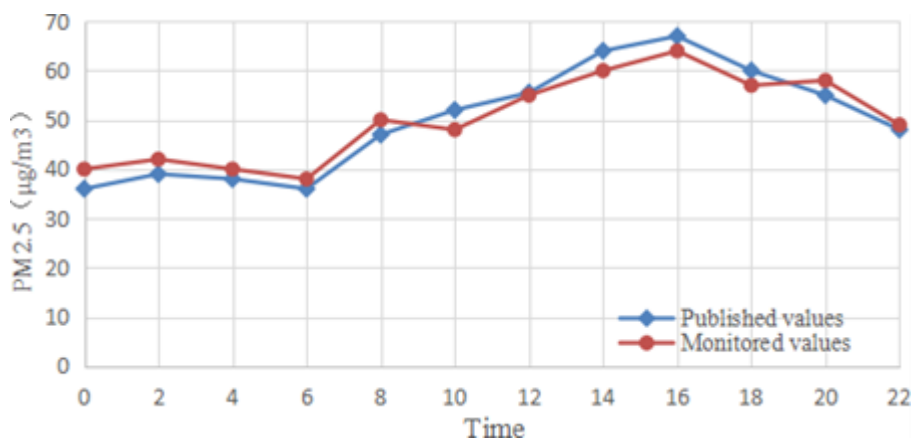


Figure 7. Analysis of portable inspection data

## **6.Conclusion**

Real-time detection of air quality is an important issue in current air pollution monitoring. This paper has designed an air quality detection system based on Bluetooth communication. It uses a variety of low-cost air sensors to detect indoor and outdoor air quality in real-time. The data is sent to the mobile APP in real-time, which facilitates the user's query. Experimental results show that the system can accurately detect and display the air quality of the user's current location, which has certain significance for low-cost air quality detection.



## 译文

# 基于蓝牙通信的空气质量检测系统设计

Qidong Ling, Gang Wang and Ming Wang

## 1. 介绍

随着雾霾等严重大气污染现象的发生，人们对大气环境污染越来越重视，希望能够及时掌握监测生活环境中大气污染的具体数据，从而合理安排和开展相关活动，尽可能减少空气污染造成的危害。国家环境监测部门实时发布区域空气质量，但由于监测点数量有限，位置固定，监测数据有限，无法实时准确显示用户当前位置的空气质量，无法满足普通用户对空气质量监测的需求。针对上述问题，提出了一种基于 Arduino 控制器的空气质量检测系统，通过蓝牙通信将数据实时上传到手机 APP，实现对常见室内外空气污染物的监测。系统能够准确检测并显示用户当前所在位置的空气质量信息，并将数据上传到移动终端。

## 2. 系统功能与结构

便携式空气质量检测系统的系统结构如图 1 所示，包括单片机、传感器模块、LCD 显示模块、报警提醒模块、通信模块等部分。控制器选用 Arduino 纳米控制平台。Arduino Nano 是一个基于 ATmega328P 的开发平台。它有 14 个数字 IO 引脚和 6 个模拟输入引脚。它可以虚拟化多个软串口，可以实现多个模拟和数字传感器接入。Arduino 微控平台对空气传感器采集到的空气污染物信号进行实时处理和分析，并将处理结果发送到 LCD 显示模块和手机终端进行显示。同时，实时将处理结果与预设阈值进行比较，系统在预设阈值时向告警提醒模块发送警告信号。该系统能够准确检测并显示用户当前所在位置的空气质量，并将数据上传到移动终端。传感器模块主要包括 PM2.5 采集电路、一氧化碳采集电路、甲醛采集电路和温湿度采集电路。

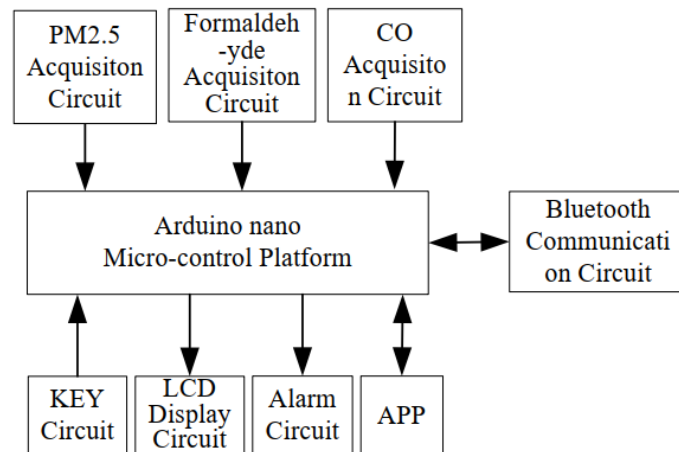


图 1.系统结构图

### 3. 系统开发与设计

#### 3.1 PM2.5 采集模块设计

PM2.5 采集模块采用 GP2Y1014AU 传感器。GP2Y1014AU 是夏普公司研发的一款光学粉尘监测传感器模块。空气可以通过模块中间的孔自由流动。红外 led 和光电晶体管被放置在孔的角落。红外发光二极管定向发射红外光。当空气中的微粒阻挡红外线时，红外线发出漫反射;光电晶体管接收红外线，信号输出引脚的电压变化。PM2.5 采集模块电路如图 2 所示。

GP2Y1014AU 传感器模块的 LED 引脚与  $150\Omega$  电阻串联，以限制电流供应以保护发光二极管。发光二极管的正极和负极与  $220\mu\text{f}$  电容器并联以实现稳定的电源供应。LED 引脚为脉冲输入引脚，连接 Arduino Nano 控制模块。连接 D2 引脚为传感器提供输入信号，并控制 GP2Y1014AU 传感器内部的 LED 灯以  $1\text{ms}$  的周期打开和关闭。由于 GP2Y1014AU 传感器需要  $5\text{V}$  电源，而单片机的 AD 采样电压最大为  $3.3\text{V}$ ，所以信号输出  $\text{Vo}$  引脚需要分成两个电阻 R23 和 R24。传感器的  $\text{Vled}$ 、LED\_GND 和 LED 引脚控制内部 LED 灯。中心的通风口可以让空气自由流动，通过检测空气中灰尘折射的光来确定灰尘的含量。传感器  $\text{Vo}$  引脚输出的模拟电压与测量到的粉尘浓度成比例，传感器输出的模拟电压经 AD 转换处理后转换为数字信号。

#### 3.2 CO 采集模块设计

一氧化碳传感器采用 ZE16-CO 通用气体模块。ZE16-CO 利用电化学原理检测空气中的一氧化碳。内置温度传感器可进行温度补偿，具有良好的选择性和稳

定性。通过 pin5 和 pin6 连接 Arduino 纳米控制模块软串口引脚 D10 和 D11 引脚。基于串口协议，采集到的一氧化碳数据以 1s 周期发送到 Arduino 纳米控制模块。

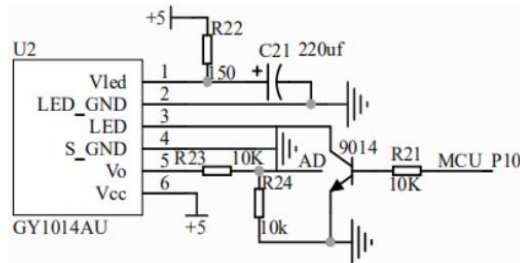


图 2. PM2.5 采集电路设计

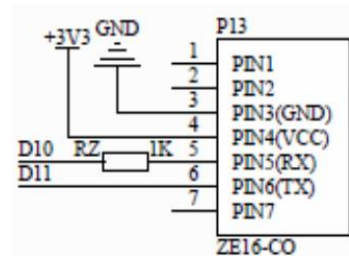


图 3. CO 采集电路设计

### 3.3 甲醛、CO<sub>2</sub> 采集模块设计

甲醛采集模块采用 Adafruit 的 SGP30 传感器。SGP30 是一种数字式多像素气体传感器。它采用温控微加热板和两个预处理室内空气质量信号，提供空气中甲醛和二氧化碳的信息。甲醛采集电路图如图 4 所示。Arduino Nano 控制模块的 I2C 总线接口通过驱动电路连接到 SGP30 的 SCL 引脚和 SDL 引脚。SGP30 传感器的 SCL 引脚和 SDL 引脚将数字信号传输到 Arduino Nano 控制模块的 I2C 总线接口。由于 SGP30 传感器和 Arduino Nano 的供电电压不同，所以在两者之间增加了一个驱动电路。数据传输基于 I2C 通信，SDA 为双向数据线，SCL 为时钟线。工作原理是主机先发送最高位发送启动信号。当时钟线状态高时，双向数据线由高变低，主机开始发送数据。数据传输完成后，主机发送停止信号，否则，当时钟线处于低状态时，双向数据线由低状态变为高状态。

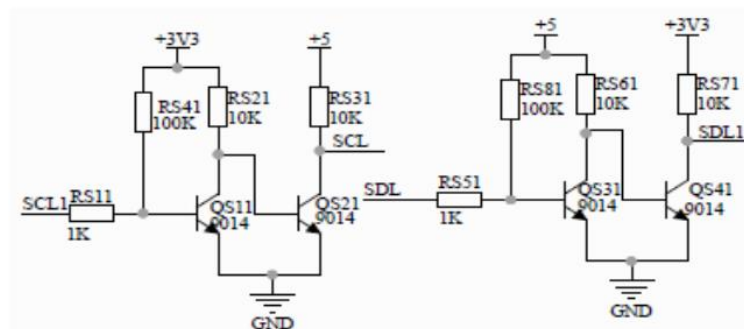


图 4. 甲醛采集电路设计

### 3.4 蓝牙通信模块设计

蓝牙通信模块采用 CC2564 芯片，基于 TI 公司第七代蓝牙核心，兼容蓝牙

4.2 协议，能耗低。芯片具有温度检测和补偿功能，确保在温度范围内 RF 性能可靠。CC2564 的多个 LDO 引脚接地并连接到内置的高级电源管理系统，为内部信号处理[5]提供可靠的参考电压调节器。蓝牙与单片机的电路设计如图 5(a)所示。CC2564 使用四线 UART 的标准 HCI 接口。由于蓝牙引脚的逻辑高电平为 1.8V，与 MCU 引脚的逻辑高电平不匹配。实现两级信号转换需要两台 sn74lvc2t45 双功率母线收发器。

CC2564 射频电路如图 5(b)所示。BT\_RF 引脚发射一个单端射频信号。它的物理基带时钟是由一个低频晶体振荡器通过锁相环乘频产生的。内部功率放大器对信号进行放大。FL1 是蓝牙滤波器，C127 用于匹配板载天线的阻抗。R29 和 R30 是零欧姆电阻，用于选择板载天线或外部天线。发射功率可达 10dBm，接收灵敏度为-95dbm，具有良好的射频性能。

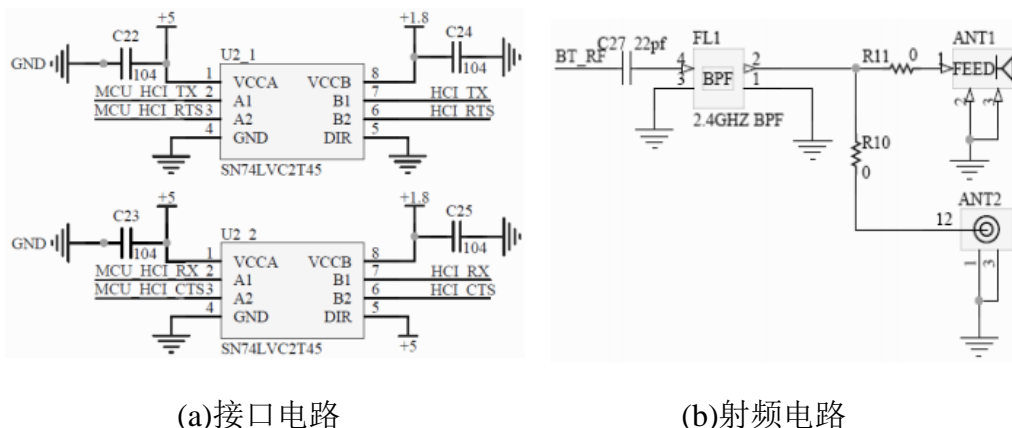


图 5.蓝牙通信系统设计

#### 4. App 的设计

为了方便对设备检测到的数据进行存储和处理，设计了一款基于 Android 系统的 APP。检测设备检测到的数据通过蓝牙通信实时传输到手机 APP 上。APP 将数据存储在 SQLite 数据库中，实时同步到活动界面中，可显示和查询。

首先，需要在注册文件 AndroidManifest.xml 中声明权限：BLUETOOTH\_ADMIN 权限、BLUETOOTH 权限、ACCESS\_COARSE\_LOCATION 权限，用于启动设备、连接请求、动态获取位置权限。本设计使用蓝牙适配器类获取蓝牙适配器对象并启用蓝牙功能，扫描周围蓝牙，注册广播，然后通过 BroadcastReceiver 接受扫描结果，并对预绑定的蓝牙设备进行配对。同时，在 pin 码配对验证后，可以将绑定地址添加到新的蓝牙设备

中。配对成功后，套接字连接到设备，蓝牙连接线程打开。通过蓝牙的 UUID，基于 SPP 串行通信协议建立连接。连接成功后，数据传输线程开始接收硬件设备发送的数据并保存到 SQLite 数据库中，利用 ListView 控件设计历史信息查询功能，通过图表显示环境空气质量的趋势。

## 5. 系统测试和数据分析

### 5.1 系统测试

本系统采用嵌入式设计方法，完成了硬件测试系统，并开发了基于 Android 系统的手机 APP。通过单片机 Arduino Nano 和传感器采集空气质量，通过蓝牙通信将采集到的数据传输到手机 APP，实现空气质量数据的显示存储、报警等功能，从而完成空气质量的实时监测和空气质量的检测。液晶显示可实现 PM2.5、一氧化碳、甲醛、温湿度等信息。APP 的显示效果如图 6 所示。打开 APP 后，先检测蓝牙模块，然后搜索蓝牙并连接。连接成功后，页面将显示连接页面。蓝牙模块连接成功后，依次接收硬件系统在当前空气中检测到的当前空气温度、湿度、PM2.5、CO(一氧化碳)、TVOC(甲醛)值。当数值大于预定值 value 时，app 上会出现蜂鸣器报警，并弹出报警提示对话框。

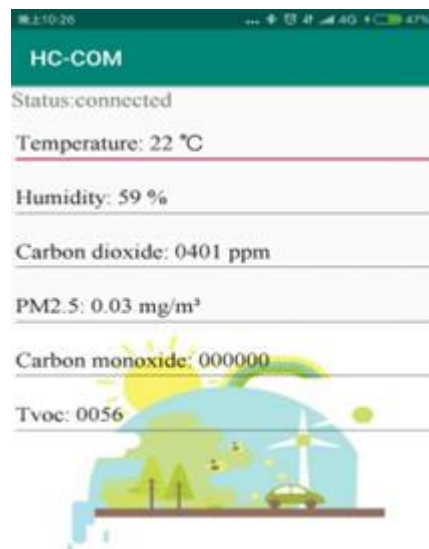


图 6.应用测试渲染

### 5.2 试验性能分析

选取 2019 年 9 月 23 日起一天内怀塔监测点附近的室外 PM2.5 检测数据进行分析，计算便携式空气质量监测系统 2 小时周期监测数据与环境监测站怀塔监测点监测数据的平均值。与公布数据对比，分析结果如图 7 所示，监测系统的监测

检测值与环境监测站监测点的公布值趋于一致，偏差基本保持在 5% 以内，可以更准确地指示当前位置的大气污染物浓度。

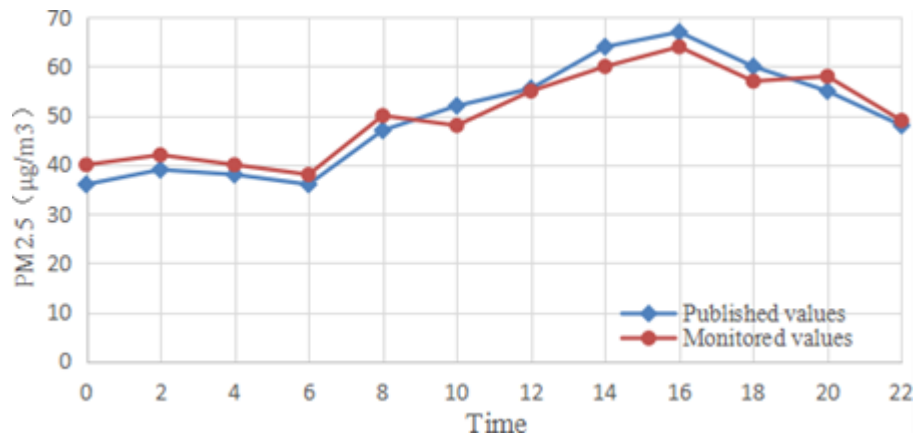


图 7.便携式检测数据分析

## 6. 结论

空气质量的实时检测是当前空气污染监测中的一个重要问题。本文设计了一种基于蓝牙通信的空气质量检测系统。它使用各种低成本的空气传感器，实时检测室内外空气质量。数据实时发送到手机 APP，方便用户查询。实验结果表明，该系统能够准确检测并显示用户当前所在位置的空气质量，对低成本的空气质量检测具有一定意义。