

# 基于 STM32 的物联网环境监测系统

**摘 要：**随着人民对美好生活的向往，人们对于环境的重视程度越来越强烈，环境对生活的影响已经成为一个热点问题。本设计以 STM32 单片机作为控制和数据处理的单元，使用 AHT10、BH1750 和 BMP280 传感器去监测周围的环境参数，在 LCD 屏完成传感器数据和相关信息的显示。同时，采用了无线模组与单片机之间进行数据交互。当无线通讯模组成功连接网络，并接入物联网云平台时，用户可以通过登录物联网云平台后台查看传感器上报的数据和进行远程控制。在设计过程中，开发 Android 上位机软件，使用户在上位机实现对于温度、湿度、光照强度和气压等相关环境参数监测。

**关键词：**STM32；物联网；环境监测

---

# Internet of Things environmental monitoring system based on STM32

**Abstract:** With the people's yearning for a better life, people pay more and more attention to the environment, the impact of the environment on life has become a hot issue. This design uses STM32 as the control and data processing unit. Meanwhile, the AHT10, BH1750, BMP280 sensors to monitor the surrounding environmental parameters. Display of sensor data and related information on the LCD screen. At the same time, the wireless module and the single chip microcomputer are used for data interaction. When the wireless communication module is successfully connected to the network, and connected to the Internet of Things cloud platform. By login to the background of the Internet of Things cloud platform. Users can view the data reported by the sensor and conduct remote control. In the design process, Android software application is developed to enable users to monitor relevant environmental parameters. Display temperature, humidity, light intensity and pressure data on the APP.

**Keywords:** STM32; Internet of Things; Environmental Monitoring

---

# 基于 STM32 的物联网环境监测系统

## 1 引言

### 1.1 课题研究背景

在我国经济稳定发展的背景下，人们的生活质量得到了明显的提高。与此同时经济发展所带来的环境的问题，开始慢慢展现出来。环境问题被越来越多的人重视。为了拥有更健康的生存环境，治理环境污染，改善环境质量，促进绿色发展是目前社会发展的形势所趋<sup>[1]</sup>。

我国研究环境监测系统的研究发展过程比较坎坷。我国发展起步比较晚，与其他西方国家相比较，相关领域落后几十年。主要原因是由于当时的社会生产水平不高，国内相关的技术也没有很多，国外的相关核心技术进行封锁，限制性太大。只能去引进国外的环境监测系统设备以及获取服务。同时由于会操作国外先进设备的人员较少，很难实现大规模使用。

我国环境监测工作是随着国家对于环境保护的加大投入，向高质量发展的目标，从而发展起来的。国内的传统的环境监测装置系统，一般有易受到现场安装使用环境被限制的缺点。场景的应用适用性比较差。传统的测量和控制操作手段比较单一，不能够远距离的监控和报警等提示功能。但是，随着国家对物联网信息技术的发展加大投入。国内相关物联网平台和传统的环境监测的厂商之间共同合作。我国的环境监测系统的发展方向是明确的，是向着智能化方向前进。实现传感器设备端与云端之间的互联，进一步向着万物互联的方向发展。

国外的环境监测系统设备的研究发展情况如下。传感器技术以及相关电子技术最先出现在西方国家。相关的传感器芯片和数据处理软件有着快速的发展，早实现了智能化操作。许多西方发达国家的环境监测系统装置，已经能同时进行多路采集各种周围环境的相关质量参数<sup>[2]</sup>。通过物联网技术实现数据的云端监测和进行相关处理分析，从而进行预测并得到相关结论。历经多年的发展和相关电子信息技术的进步，研制出具有自动化、智能化的监测系统成为各个国家的目标。

### 1.2 课题研究目的及意义

进入 21 世纪，随着物联网信息技术与电子信息技术的快速进步。因此有必要去设

---

计出一种以物联网技术、传感器技术和电子信息技术构成的多功能环境监测装置。该系统可以同时监测周围环境的多路环境参数。通过环境监测系统装置<sup>[3]</sup>，能够帮助人们准确的去监测周围环境的参数。力求以最大的力度，寻求科学合理的方法，去解决日常生产生活出现的环境问题。

通过监测系统的长期的积累的数据，分析和追溯污染源头，分析其变化规律。从而建立相关的预警和预报系统。环境监测对于人们的日常生产生活非常重要。不但体现在环境保护和控制污染等方面，而且还是人类生存的重要的基石。坚持促进绿色发展，创建绿色中国的理念，才能拥有更美好的健康生活环境。

## 2 总体方案设计

### 2.1 整体方案设计

本环境监测系统设计选取了控制和数据处理能力强的 STM32 单片机作为核心部件。使用温湿度传感器对周围环境中的温度与湿度监测，选取高精度的光强度传感器完成对周围环境的光照强度的测量，对于气压数据的测量，将选取高精度的气压传感器完成工作。该系统使用液晶显示屏幕完成对温湿度数值，光照强度数值，大气压强数值以及设备的状态信息的显示。当无线通讯模组成功连接网络之后，接入云平台。用户可以通过登录物联网云平台后台<sup>[4]</sup>，查看传感器上报的数据信息和进行远程控制。开发 Android 上位机软件，使用户在上位机程序上实现对各种环境质量参数监测。整个系统的整体框架图如图 2-1 所示。

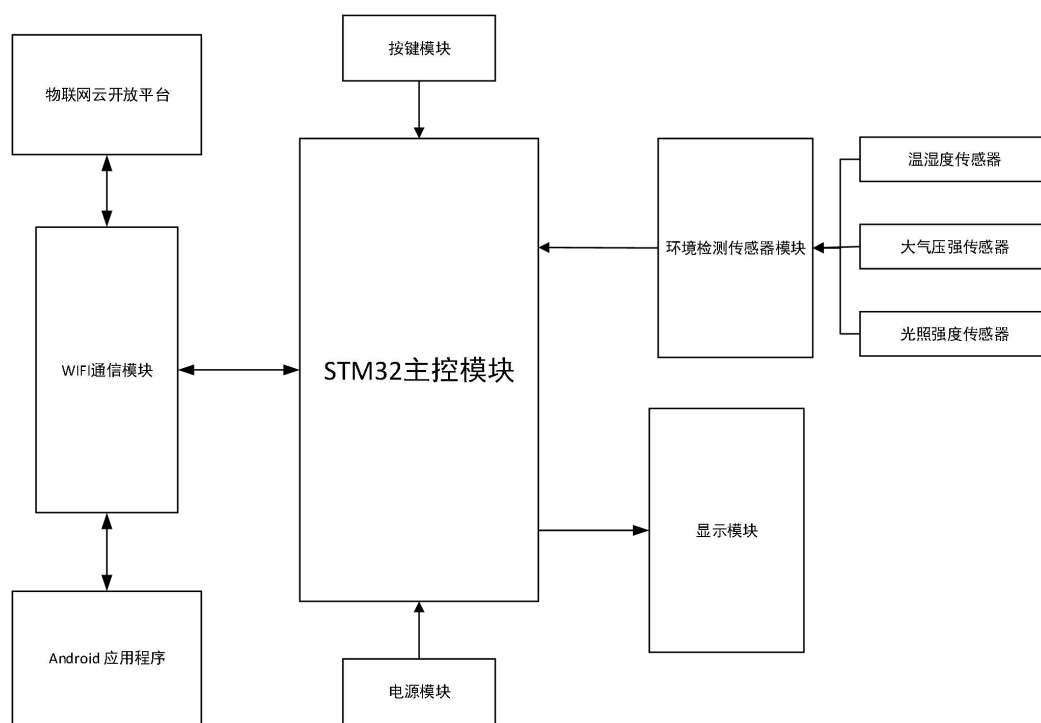


图 2-1 系统整体框架图

### 2.2 控制模块方案

本环境监测系统设计使用单片机作为整个系统的控制中心。需要完成传感器数据的采集，无线通讯模组的数据交互，显示传感器数据的功能<sup>[5]</sup>。在本次的环境监测系统中，控制模块采用的是 STM32 系列单片机。

例如在芯片选择上面,有低功耗类型 STM32L 系列、主流类型 F1 系列、高性能 F7/H7 系列、以及异构系统架构下的 STM32MP157 系列。在软件程序设计过程中,官方提供了图形化配置软件 STM32CubeMX。可以快速生成底层配置代码,减少重复性移植。同时该 32 位芯片的相关技术资料 and 参考设计资料较多,因此被运用到各种电子系统设计之中。本课题准备使用控制性能强劲和数据处理能力强的 STM32F103ZET6 芯片作为核心控制模块元件。

## 2.3 环境监测传感器方案

环境监测系统主要需要监测的数据有温湿度数据,光照强度数据和大气压强数据等基本环境质量参数。因此需要选择相应的温湿度传感器,光强传感器,精度较高的气压传感器<sup>[6]</sup>。

温湿度数据采集传感器选择了 AHT10。AHT10 模块上面有一个湿度传感器元件和一个片上温度传感元件,该产品具有快速响应、抗干扰能力强和高精度等优点<sup>[7]</sup>。

光照强度数据采集的传感器选择了 BH1750。BH1750 是标准 (I<sup>2</sup>C) 接口的 16 位数字输出类型的环境光强度传感器,可以利用 BH1750 模块的高分辨率探测较大范围内的光照强度变化情况<sup>[8]</sup>。

大气压强数据采集传感器选择了 BMP280。数字式气压传感器 BMP280,具有高精度的特点,而且也具有测量环境温度和高度的功能。

综上所述的传感器模块,在选择时考虑到了其成本,使用的方便性,相关设计资料的丰富性,传感器模块实物图如图2-2所示。

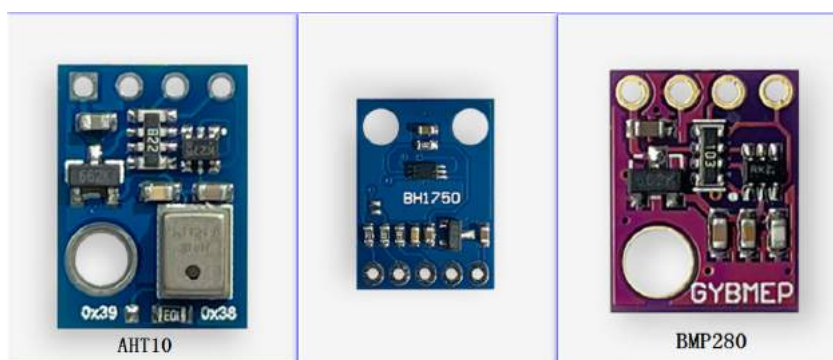


图 2-2 传感器模块图

## 2.4 无线模块方案

随着物联网技术等相关通信技术的快速发展和更新迭代,目前的物联网技术呈现出

智能化的特点<sup>[9]</sup>。因此在本系统设计中，决定使用远程通信模块进行数据通信和进行智能化操作。对接国内开放的物联网云平台，从而更加智能的，简单的完成整个设计的功能。

方案一：对于长距离无线通讯模块的选择。查询资料后，了解了国产 LTE 无线通讯模组上海合宙公司。该公司研发的 LTE Cat 1 无线通信模组提供了丰富的通用外设接口，并且支持 Lua 二次开发，合宙官方提供了嵌入式脚本运行框架 LuatOS。但是模组价格比较贵，而且开发过程比较复杂，相关资料和设计参考较少，开发难度较大。

方案二：ESP8266 芯片是国产芯片公司乐鑫科技所研发的一款 WIFI 无线模组芯片。内置 TCP/IP 协议栈，支持 AT 本地升级和 OTA 远程升级。该无线模组能够很好的接入国内的开放的物联网云平台，具有易操作化的特点，主要是相关资料和设计参考较多，开发难度较小。

经过对上述方案对比，方案二中的 ESP8266 模块，拥有丰富的相关设计资料。最终选择了以 ESP8266 芯片作为核心的无线模组。作为单片机与物联网云端进行数据传输和交换的单元，无线模块实物图如图 2-3 所示。



图 2-3 无线模块

## 2.5 液晶显示模块方案

本次环境监测系统的设计过程中，需要使用显示模块进行显示。需要的显示的数据，主要包括传感器设备测量采集到的数据和系统的基本状态信息。在电子系统设计过程中，常使用的显示模块主要有以下两种，TFT 彩屏和 LCD1602 两种。方案对比如下。

方案一：TFT 屏幕是以背透和反射相结合的方式工作的，通过点脉冲直接控制<sup>[10]</sup>。从而实现显示效果。查询资料了解到了一种 1.8 寸的 TFT 彩屏模块。在色彩显示上，TFT 屏幕的效果较好，色彩丰富且具有对比度高特点。同时在使用该 LCD 显示模组时，需要用到的单片机的 IO 端口数量较少。

方案二：LCD1602 是一种字符型液晶显示模块，具有成本低等特点。但是缺点是在

---

使用 LCD1602 需要较多的单片机的 IO 端口数量。而且需要通过调节变阻器的大小，才能改变屏幕的显示效果，不能直接得到显示效果。

经过上述两个方案对比，由于需要对传感器数据的实际测量值进行显示，选择了显示内容更多和色彩效果更好的方案一，TFT 显示模块实物图如图 2-4 所示。



图 2-4 TFT 显示屏幕





毫安，因此芯片功耗非常低。芯片可对系统电路之中的电压进行检测，可与电源芯片相接一起完成开关电源调节。芯片正常运行时，需要一个外部时钟触发信号，为此将使用频率范围在4~16MHz之间的晶体振荡器。芯片内部存在一个自动校准的RTC振荡电路，其实际运行的频率能够达到32KHZ。在本次设计中，因为STM32系列拥有丰富的串口接口，因此可以实现与多个通讯模组通信。同时可以添加调试接口，利用串口软件实现调试功能，打印设备状态信息。

### 3.2 温湿度传感器模块

温湿度传感器模块核心是以AHT10芯片作为核心部分，还包括电平转换电路。AHT10芯片的产品特性如下，芯片的输入电压范围：2.3V至3.3V；具有低功耗的特点；芯片通讯方式采用标准的I<sup>2</sup>C接口通信。该温湿度传感器芯片的精度如下，湿度精度±3%RH，温度精度0.5℃。模块的参考设计电路如图3-2所示。

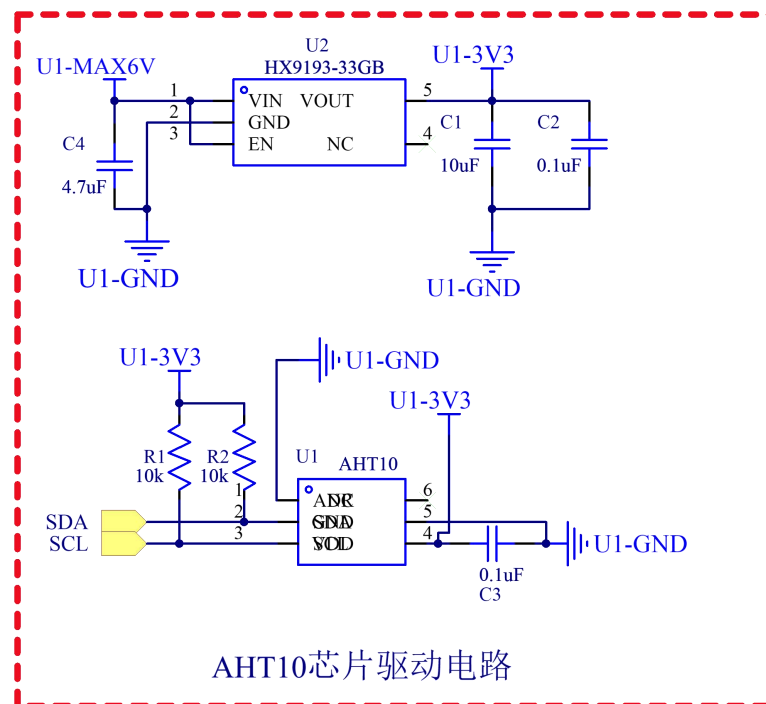


图 3-2 温湿度传感器模块

### 3.3 光照强度传感器模块

光强传感器模块核心是以BH1750芯片作为核心部分，还包括电平转换电路。BH1750芯片的产品特性如下，芯片的输入电压范围：3.0V至3.6V之间；其峰值灵敏度波长的典

型值为560nm；输入的光照强度范围为1~65535Lx，其最小的分辨率为0.5Lx<sup>[11]</sup>。BH1750芯片采用的通讯方式是标准的I<sup>2</sup>C接口通信。模块的参考设计电路如图3-3所示。

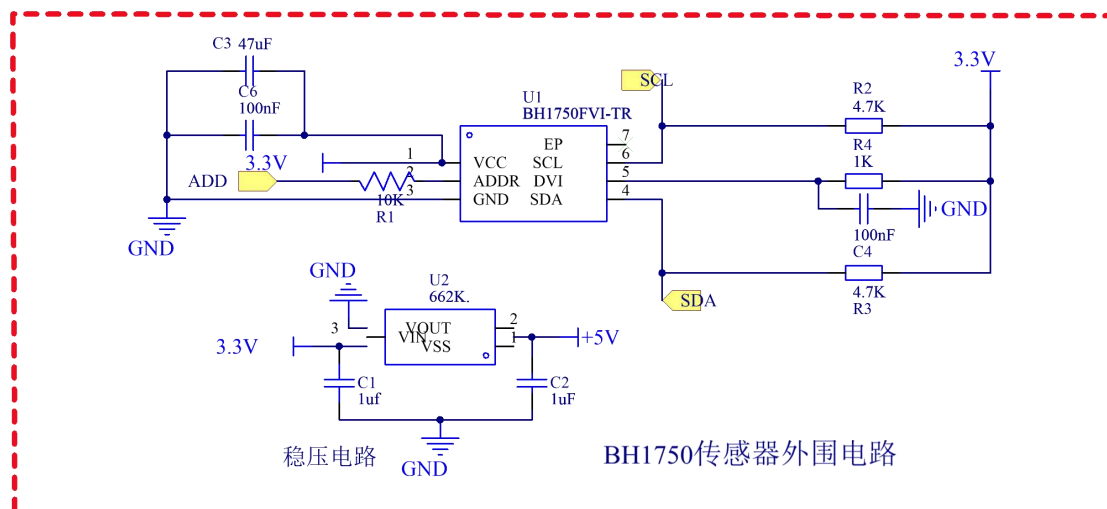


图 3-3 光照强度传感器模块

### 3.4 大气压强传感器模块

气压传感器模块核心是以 BMP280 芯片作为核心部分，还包括电平转换电路。气压传感器的产品特性如下，芯片的输入电压范围：1.7~3.6V；芯片的气压工作范围为 300hPa 至 1100hPa；芯片工作温度范围为-40℃至 85℃；BMP280 芯片支持标准的 I<sup>2</sup>C 接口通信和 SPI 通信；模块参考设计电路图如图 3-4 所示。

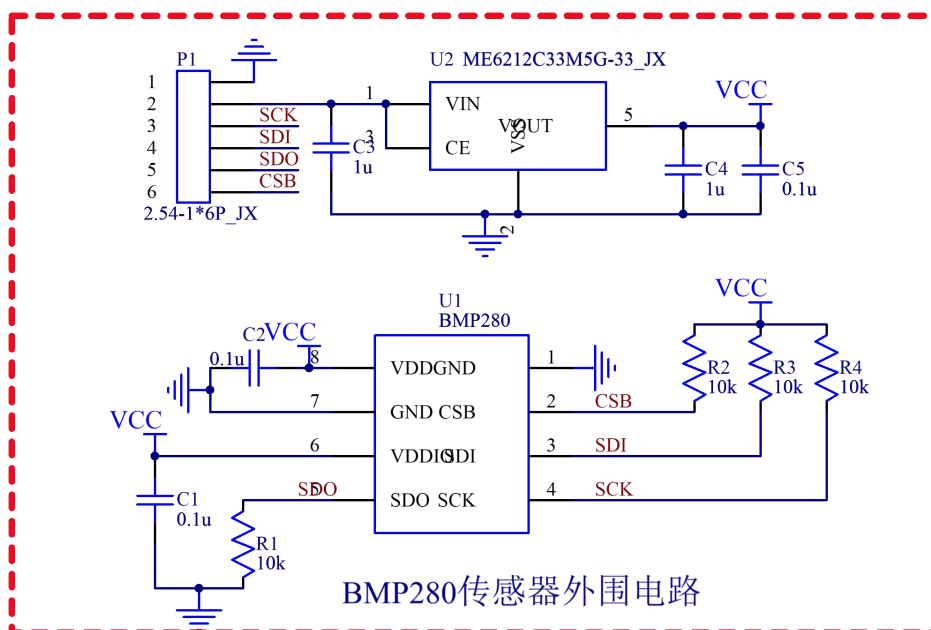


图 3-4 气压传感器模块

### 3.5 显示模块

在本次的环境监测系统设计中,选择了 1.8 寸 TFT 彩色屏幕作为该系统的显示模块, LCD 彩屏采用 SPI 接口通讯方式, 驱动 IC 是 ST7735S, 其支持的显示分辨率为 128\*160; 支持 16Bit RGB 范围 65K 色显示。该显示模块支字符, 字符串, 汉字, 图片等效果显示。 彩色 TFT 显示模块接口电路图如图 3-5 所示。

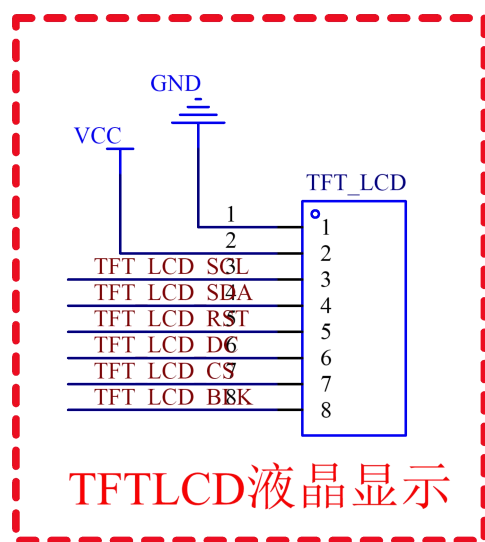


图 3-5 显示模块接口

### 3.6 无线通信模块

环境监测系统选择了 ESP8266 系列芯片作为无线通讯部分。ESP8266 系列芯片, 其 CPU 时钟速率最高可以达到 160MHz, 同时具有丰富的外设接口; 经典的 WIFI 模式支持三种类型, 包括 Station 客户端模式、SoftAP 服务端模式和 SoftAP+Station 混合模式<sup>[12]</sup>; 该无线模组支持 TCP、HTTP、MQTT 等基本网络协议。无线通信模块参考电路设计如图 3-6 所示。



### 3.8 电源接口电路

在电源供电方式选择过程中，可以采用的是连接 USB 接口进行供电。由于大部分的传感器设备模块的输入电压范围在 3.3V 左右，所以需要设计电平转换电路。该部分主要由电源线引入 5V/1A 的电源给系统供电，经 KIA1117 芯片降压为 3.3V，给 STM32 主控系统和传感器设备供电。电源接口电路图如图 3-8 所示。

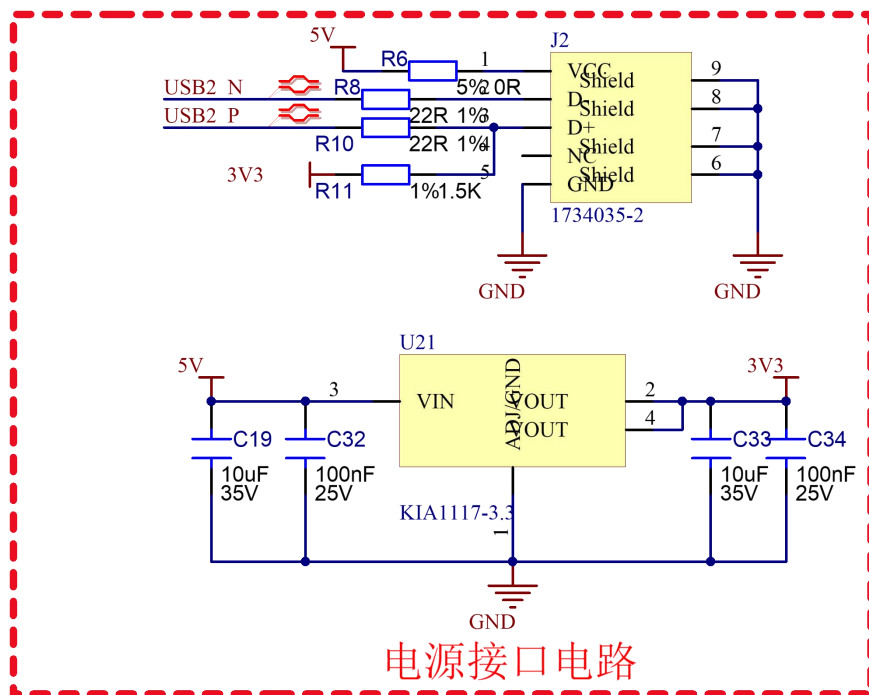


图 3-8 电源接口电路

## 4 系统软件设计

### 4.1 主程序设计

基于 STM32 的环境监测系统的主程序流程如下。整个系统，在提供电源之后，开始进行系统模块的初始化操作。如进行传感器设备的初始化，LCD 模块的初始化，无线通讯模组的初始化。当完成初始化操作之后，LCD 屏幕则会显示初始化信息。此时，利用路由器或手机热点提供 WIFI 信号，WIFI 模块会进行联网操作。当联网成功时，系统开始接收传感器设备采集的数据，APP 程序会显示相关传感器数据信息，LCD 屏幕也会显示环境监测系统的主要监测数据。如果联网失败，则会重新尝试连接网络。主程序设计流程图如图 4-1 所示。

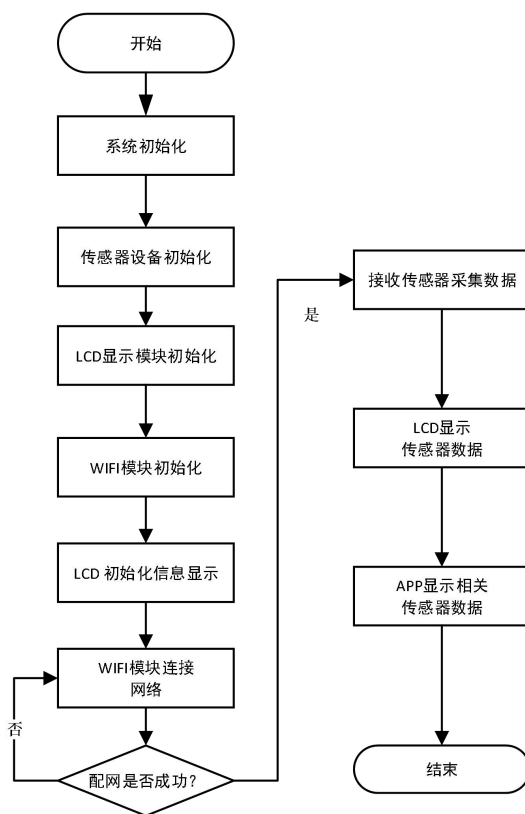


图 4-1 主程序设计流程图

### 4.2 传感器数据采集及处理

环境监测系统主要需要监测的数据，有温湿度数据，光照强度数据和大气压强数据。选择的环境监测系统的传感器，AHT10，BH1750，BMP280 都是通过标准的 I<sup>2</sup>C 接口进

行通信。

对于温湿度数据的采集，使用AHT10传感器。AHT10采集数据的流程。第一步，模块进行上电，等待40ms，目的是在读取温湿度数据过程之前，进行初始化操作。第二步，当等待时间结束后，进行触发测量操作。第三步，等待测量过程完成，从而获取传感器采集到的温湿度数据。流程如图4-2所示。

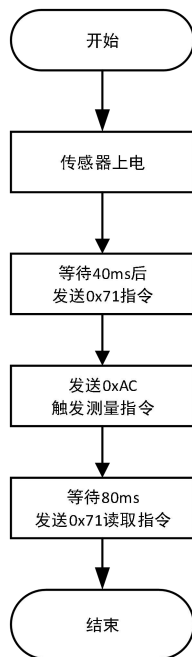


图 4-2 温湿度获取流程

对于光照强度数据的采集，使用BH1750 传感器。计算BH1750 采集结果，单片机通过I<sup>2</sup>C协议读取的第一个字节是 BH1750 寄存器的高 8 位，第二字节为寄存器的低 8 位数据，高 8 位数据左移 8 位再加上低 8 位数据<sup>[13]</sup>，利用公式去计算最终的光照强度数值。测量步骤如图 4-3 所示。

● 测量程序步骤

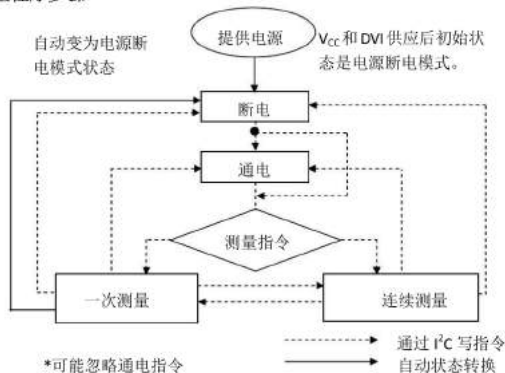


图 4-3 BH1750 测量程序步骤



对于气压数据的采集，使用 BMP280 传感器。BMP280 传感器的初始化流程如下，第一步，芯片内部数据全部清零；第二步，读芯片 ID；第三步，进行测量控制寄存器操作；第四步，进行测量，测量时间的间隔为 0.5ms；第五步，读取补偿值数据。官方数据手册提供的 BMP280 测量流程如图 4-4 所示。

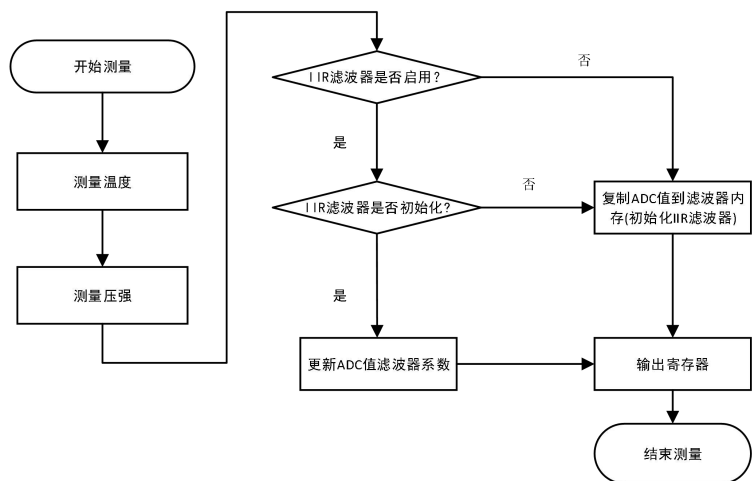


图 4-4 BMP280 测量流程

在读取数据时，需要进行滤波操作使结果的更加稳定。BMP280 传感器获取气压数值的底层驱动代码如图 4-5 所示。

```

//-----
//  函数功能: BMP280 外设驱动函数部分
//  函数标记: bmp280GetPressure
//  函数说明: 传感器获取大气压数值
//-----
//  | - | - | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
//-----
static void bmp280GetPressure(void)
{
    u8 data[BMP280_DATA_FRAME_SIZE];

    // read data from sensor
    BMP_iicDevRead(BMP280_ADDR, BMP280_PRESSURE_MSB_REG, BMP280_DATA_FRAME_SIZE, data);
    bmp280RawPressure = (s32) (((uint32_t) (data[0])) << 12) | (((uint32_t) (data[1])) << 4) | ((uint32_t) data[2]) >> 4);
    bmp280RawTemperature = (s32) (((uint32_t) (data[3])) << 12) | (((uint32_t) (data[4])) << 4) | ((uint32_t) data[5]) >> 4);
}
  
```

图 4-5 BMP280 驱动代码

### 4.3 显示模块程序设计

环境监测系统选择了 1.8 寸 TFT 彩屏作为显示模块部分。主要的显示内容是传感器采集的数据，网络状态信息，系统信息，NTP 网络时间以及系统开机启动的界面。LCD 显示驱动，提供了基本的显示基本的英文字符，数字，以及显示汉字和图片的函数接口，同时支持不同的大小的字体的函数接口。

第一部分，系统上电开机启动界面，显示开机界面图片及相关信息。在开机启动界

面的设计中,使用彩色图片取模软件,进行设置图片尺寸大小,色彩参数。从而获取对应的数组代码。调用 LCD 图片显示函数 TFT\_ImageShow 显示图片,支持不同尺寸的大小的图片显示,本次显示的图片大小为 120\*120。

第二部分,显示环境监测系统的系统信息功能选择界面。调用 system\_show()函数,实现的效果是三种基本信息功能选择的展示。第一, SensorMessage 传感器数据信息展示功能,第二, NetworkMessage 网络状态信息展示功能,第三, SystemMessage 系统基本信息展示功能。

第三部分,显示环境监测系统的传感器数据信息 SensorMessage。数据信息主要是显示温湿度 Temperature 和湿度 Humidity,光照强度数据 Light\_Intensity,气压传感器数据 Pressure。驱动代码如图 4-6 所示。

```
//=====
// 实现功能: 传感器数据显示
// 函数说明: TFT_Show_SensorDATA
//=====
void TFT_Show_SensorDATA(void)
{
    //显示温度
    TFT_ShowImage(0,0,32,32,Wendu);
    TFT_ShowString_16x16(30,0,"Temp",Blue1,White);
    TFT_ShowNumber_Float_16x16(30,16,aht_temp,2,1,Red,White);
    TFT_ShowChinese_16x16(64,0,"℃",Magenta,White);
    // 显示湿度
    TFT_ShowImage(78,0,32,32,shidu);
    TFT_ShowString_16x16(112,0,"Humi",Blue1,White);
    TFT_ShowNumber_Float_16x16(108,16,aht_hum,2,1,Red,White);
    TFT_ShowString_16x16(150,16,"%%",Magenta,White);
    // 显示光照强度
    TFT_ShowImage(0,35,32,32,Light_IMG);
    TFT_ShowString_16x16(40,30,"Light_Data",Blue1,White);
    TFT_ShowNumber_Float_16x16(30,48,Light,5,1,Red,White);
    TFT_ShowString_16x16(95,48,"Lux",Magenta,White);
    // 大气压 Hpa 正常的大气压是在1000百帕左右
    TFT_ShowImage(0,70,32,32,IMG_Hpa);
    TFT_ShowString_16x16(40,60,"Pressure",Blue1,White);
    TFT_ShowNumber_SignedInteger_16x16(40,76,Press,4,Red,White);
    TFT_ShowString_16x16(90,76,"Hpa",Blue1,White);

    TFT_ShowString_16x16(0,96,"-->SensorMessage<--",Blue1,White);
}
```

图 4-6 传感器数据信息显示

第四部分,显示环境监测系统的网络状态信息 NetworkMessage,主要是显示是否连接到无线网络的状态信息。第五部分,系统基本信息展示。

在本次显示模块的程序设计中,完成了基本数据信息的展示。由于需要进行图片显示,占用资源较多。后期可以使用 SD 卡存储媒体资源,外部调用,减少单片机内部芯片资源的使用。在功能选择界面的程序设计中,后期可以完善功能选择的部分代码,实现多级菜单的效果。

## 4.4 物联网云平台设备接入

在本次的环境监测系统的设计中，利用了无线模块进行基本通信操作。传统的无线模组与单片机进行数据交换，一般的方案是利用无线模组建立局域网实现通信功能。随着云计算等相关信息技术的进步，现在的无线通讯模组可以实现与云端连接，实现数据远程传输的功能。在本次设计中，使用的无线模组需要连接网络，并接入云平台。用户可以通过登录云平台后台，查看无线通讯模块上报的传感器数据和进行远程控制等操作。在物联网云平台的选择中，选择了国内的机智云平台。物联网设备接入的方案，如图 4-7 所示。

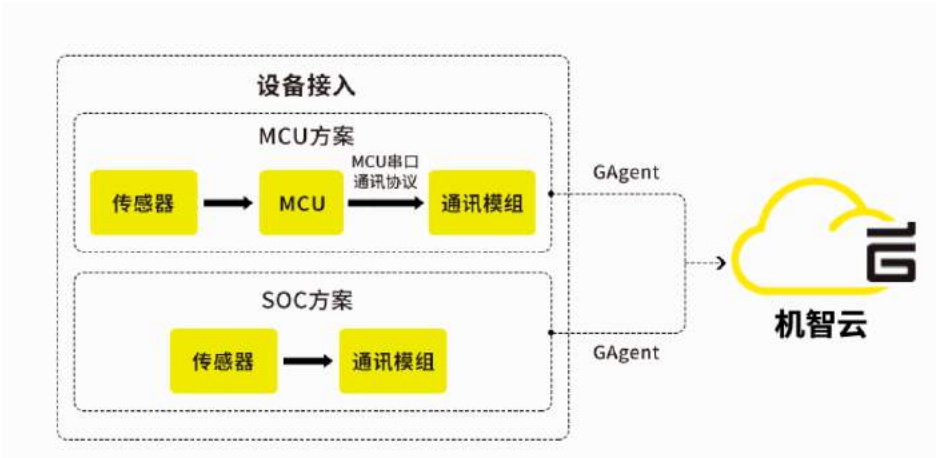


图 4-7 物联网设备接入

云平台提供了两种版本的 GAgent 通信协议的接入方案，主要是独立 MCU 方案和 SOC 方案两种。本次环境监测系统的设计中，选择了独立 MCU 方案接入平台。在本次设计过程中，需要把 GAgent 协议移植到 WIFI 模组上面，从而实现对接云平台进行数据交互的功能。云平台提供了 GAgent 固件，GAgent 固件的主要作用是完成对数据进行上报云平台 and 平台远程下发命令的功能<sup>[14]</sup>。固件下载效果如图 4-8 所示。

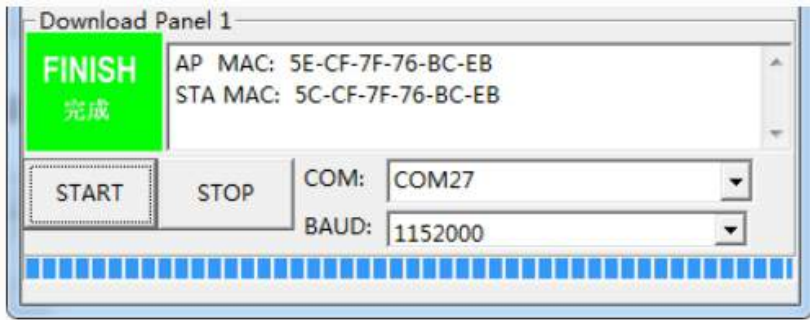


图 4-8 固件下载

在本次的环境监测系统是软件设计中，采用了 MCU +WIFI 通信模组方案，现在介

绍接入机智云平台的流程。第一步，注册 IoT 平台的开发者账号，并登陆选择产品类型并创建。第二步，填写产品相关的基本信息之后，获取到平台提供的 Product Key 和 Product Secret 等密钥信息<sup>[15]</sup>，如图 4-9 所示。

产品信息

基本信息

数据点

虚拟设备

设备日志

开发向导

服务

应用配置

应用开发

MCU开发

固件升级 (OTA)

产测工具

+ 添加服务

统计

概览

新增上线

活跃设备

基本信息

产品名称: GizwitsIoT

产品类型: 智能家居/其他

技术方案: Wi-Fi/移动网络方案

通讯方式: Wi-Fi

Product Key: f256e93d2fb1437dbcfbc473946f6c9a

Product Secret: db2326f1da2242a7ac2c57f631741dd1

数据传输方式: 定长

功耗类型: 正常

设备分享功能: 未开启

创建时间: 2021-12-31

更新时间: 2022-02-10

描述: 无

图 4-9 创建产品基本信息

第三步，创建相关数据点信息，完成相关数据类型的填写。在环境监测系统的设计中，需要创建温度数据 Temperature,湿度 Humidity,光照强度 Light\_Intensity,气压 Pressure 等主要的传感器数据点，如图 4-10 所示。

产品信息

基本信息

数据点

虚拟设备

设备日志

开发向导

服务

应用配置

应用开发

MCU开发

固件升级 (OTA)

产测工具

+ 添加服务

统计

概览

新增上线

活跃设备

数据点

定义数据点

+ 新建数据点

管理

显示名称: LED	标识名: LED	读写类型: 可写	数据类型: 布尔值	<input checked="" type="checkbox"/>	<input type="checkbox"/>
备注: 无					
显示名称: Temperature	标识名: Temp	读写类型: 只读	数据类型: 数值	<input checked="" type="checkbox"/>	<input type="checkbox"/>
数据范围: 0 - 100	分辨率: 0.1	增量: 0			
备注: 温度					
显示名称: Humidity	标识名: Hum	读写类型: 只读	数据类型: 数值	<input checked="" type="checkbox"/>	<input type="checkbox"/>
数据范围: 0 - 100	分辨率: 0.1	增量: 0			
备注: 湿度					
显示名称: Pressure	标识名: Press	读写类型: 只读	数据类型: 数值	<input checked="" type="checkbox"/>	<input type="checkbox"/>
数据范围: 0 - 2000	分辨率: 1	增量: 0			
备注: 无					
显示名称: Light_Intensity	标识名: Light_Intensity	读写类型: 只读	数据类型: 数值	<input checked="" type="checkbox"/>	<input type="checkbox"/>
数据范围: 0 - 65535	分辨率: 0.1	增量: 0			
备注: 无					

图 4-10 数据点信息

第四步，使用 IoT 平台提供 SDK 文件包，进行相关代码移植。在 SDK 包生成的过程中，平台提供了两种类型。本次系统设计过程中，选择独立 MCU 方案作为最终方案。在支持的硬件平台中，有常见 MCU 类型。例如有 STM32 平台，MSP430 平台等，都提供了相关的测试 SDK 包。选择使用通用的 MCU 平台生成的 SDK 的代码移植，有利于之后移植到其他的 MCU 芯片平台上面去，选择如图 4-11 所示。



图 4-11 MCU 开发平台选择

机智云 SDK 包的代码移植到 STM32 系列平台的流程如下。在整个代码移植的过程，主要是去移植串口初始化及配置，定时器的初始化及配置<sup>[16]</sup>，传感器数据的采集接收和上报函数的配置，以及相关模块的初始化配置。

第一部分，进行移植串口配置函数操作，在本次设计中主要是使用了串口 3 与无线通信模组进行数据交换。STM32 单片机的引脚 PB10 为 USART3\_TX, 引脚 PB11 为 USART3\_RX, 其中 USART3\_TX 与 WIFI 模组 UART\_RX 连接，其中 USART3\_RX 与 WIFI 模组 UART\_TX 连接。配置串口 3 中断函数 USART3\_IRQHandler(), 移植 gizPutData(), 把数据写入到缓冲区里面，代码如图 4-12 所示。



```

//串口3中断服务函数
void USART3_IRQHandler(void)
{
    u8 res;
    if(USART_GetITStatus(USART3, USART_IT_RXNE) != RESET) //接收到数据
    {
        res =USART_ReceiveData(USART3);
        gizPutData(&res, 1); //数据写入到缓冲区
    }
}

```

图 4-12 串口 3 中断服务函数

第二部分，定时器代码移植。在本次设计中使用了定时器 3 进行毫秒定时。在定时器 3 的中断服务处理函数中移植了 gizTimerMs(), 提供系统的毫秒定时操作，代码如图 4-13 所示。

```

// 定时器3 中断服务函数函数
void TIM3_IRQHandler(void)
{
    if (TIM_GetITStatus(TIM3, TIM_IT_Update) != RESET) //检查指定的TIM中断发生与否:TIM 中断源
    {
        TIM_ClearITPendingBit(TIM3, TIM_IT_Update ); //清除TIMx的中断待处理位:TIM 中断源
        gizTimerMs();
    }
}

```

图 4-13 定时器 3 中断服务函数

第三部分，网络协议的配置。在 Gizwits 初始化函数中，调用了定时器 TIM3 的初始化，配置了 1ms 定时，USART3\_Configuration 设置波特率为 9600。然后是进行设备状态结构体的初始化，数据缓冲区的初始化配置。网络协议的配置及初始化移植的代码如图 4-14 所示。

```

//=====
// 实现功能: 机智云协议配置
// 函数说明: 进行ESP8266配网操作
// 函数备注:
//=====
// | - | - | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
//=====
void Gizwits_Init(void)
{
    TIM3_Int_Init(9,7199); //1ms系统定时
    USART3_Configuration(9600); //WIFI初始化
    userInit(); //设备状态结构体初始化
    gizwitsInit(); //缓冲区初始化
}

```

图 4-14 机智云初始化函数

传感器采集数据的成员，包括 valueTemp, valueHum, valuePress, valueLight\_Intensity 等，数据上报函数 userHandle()实现数据上报功能，代码如图 4-15 所示。

```

void userHandle(void) //数据上报给云平台的处理函数
{
    currentDataPoint.valueTemp = aht_temp ;           //上报温度数据
    currentDataPoint.valueHum = aht_hum;              //上报湿度数据
    currentDataPoint.valuePress = Press ;              //上报光照强度数据
    currentDataPoint.valueLight_Intensity = Light;    //上报气压数据
}

```

图 4-15 数据上报函数

在事件处理函数中，添加了相关的配网成功或失败以及网络时间获取的标志位，并进行相关的显示。在配置网络的操作，使用了开发板上按键模块进行配置。配网采用的 AirLink 连接模式，在主函数里面，编写了按键检测函数。当按键 KEY1\_PRES 的按下时，进行手机 APP 配网。在配网时，手机 APP 和无线模组需要连接到同一个热点信号下面进行配置。输入 WIFI 的账号 SSID 和 Password，选择 WIFI 模组的类型进行配置，等待配网完成，设备名称信息则会显示在 APP 上面。

## 4.5 上位机软件应用开发

上位机软件应用开发。在本次的环境监测系统的设计中，采用了 Android 应用程序作为上位机软件。一般在 APP 开发设计过程中，主要是进行 APP 的 UI 界面设计，和相关功能模块底层逻辑设计。在 APP 的 UI 界面设计的过程中，选择使用 XML 进行布局，在应用程序的底层逻辑设计使用的 Java 语言。应用开发选择的是安卓平台，如图 4-16 所示。



图 4-16 选择安卓平台

本次设计中，使用平台提供生成的 SDK 文件包进行修改。由于平台的提供的例程只是完成了基本的显示效果，界面效果一般。所以接下来的主要工作是对官方源码进行修改。打开 Android Studio 软件，导入从平台下载的 Demo 源码，通过修改 src 文件里面的 layout 的布局文件，进行界面 UI 设计和优化。通过对 src 文件里面的 java 代码，进行底层逻辑代码修改<sup>[17]</sup>。在界面的 UI 设计工过程，主要是传感器的数据和系统状态信息显示，基本显示效果如图 4-17 所示。APP 的界面设计需要优化，后期可以添加相关传感器历史数据的查看和实时动态曲线的功能<sup>[18]</sup>。



图 4-17 APP 界面效果



---

## 5 硬件与软件调试

### 5.1 硬件实物调试

环境监测系统的硬件实物的初步实验性调试工作。在实验性调试中，通过 USB 线接上电脑 USB 接口，实现供电。系统供上电之后，会有开机启动界面显示，如图 5-1 所示。



图 5-1 启动界面

开机结束之后，界面会停留在系统的功能选择界面。选择界面包括了三种功能展示。第一部分是传感器采集到的数据 `SensorMessage` 展示，第二部分是系统的网络状态信息 `NetworkMessage` 展示，第三部分是系统的基本信息 `SystemMessage` 展示。基本功能的选择界面，如图 5-2 所示。

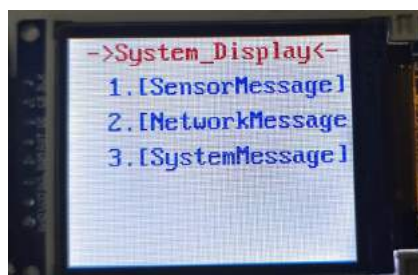


图 5-2 功能选择界面

在本次环境监测系统的设计中，当按键 KEY1 按下时，LCD 屏幕则会显示传感器数据监测信息。展示数据包括了温度 `Temp`，湿度 `Humi`，光照强度 `Light_Data`，气压 `Pressure`，显示效果如图 5-3 所示。



图 5-3 传感器数据显示界面

当按键 KEY2 按下时，LCD 屏幕则会显示该系统的网络信息。展示的内容主要部是使用的无线模组和和云平台的基本信息。在 NetworkState 设计中，如果无线模组还没有配网成功时，显示蓝色效果，如图 5-4 所示。当无线模组成功连接到 WIFI 信号时，则会显示红色效果，如图 5-5 所示。



图 5-4 未联网状态图



图 5-5 已联网状态

当按键 KEY3 按下时，LCD 屏幕则会显示系统基本信息。如图 5-6 所示。

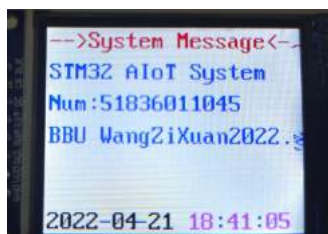


图 5-6 系统基本信息界面

在整个过程中，当无线模组成功的连接到 WIFI 网络信号时。在发送网络时间的获取的命令下，则会一直在 LCD 屏幕的最后一行显示时间。如图 5-7 所示。

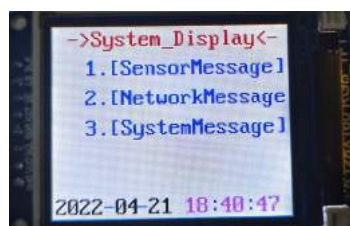


图 5-7 网络时间显示

## 5.2 软件调试

基于 STM32 的物联网环境监测系统的上位机软件调试。在使用该 Android 应用程序时，APP 和无线模组需要一起配合操作。在配网时，需要把手机和 WIFI 同时连接到同一个热点信号下面，进行配置。输入 WIFI 的账号 SSID 和 Password，选择 WIFI 模组的类型进行配置。配置过程中 APP 和硬件设备不要进行任何操作，否则会出现配置失败的情况。如果配置失败，请按下配网按键，重新配置。APP 配网流程如图 5-8 所示。

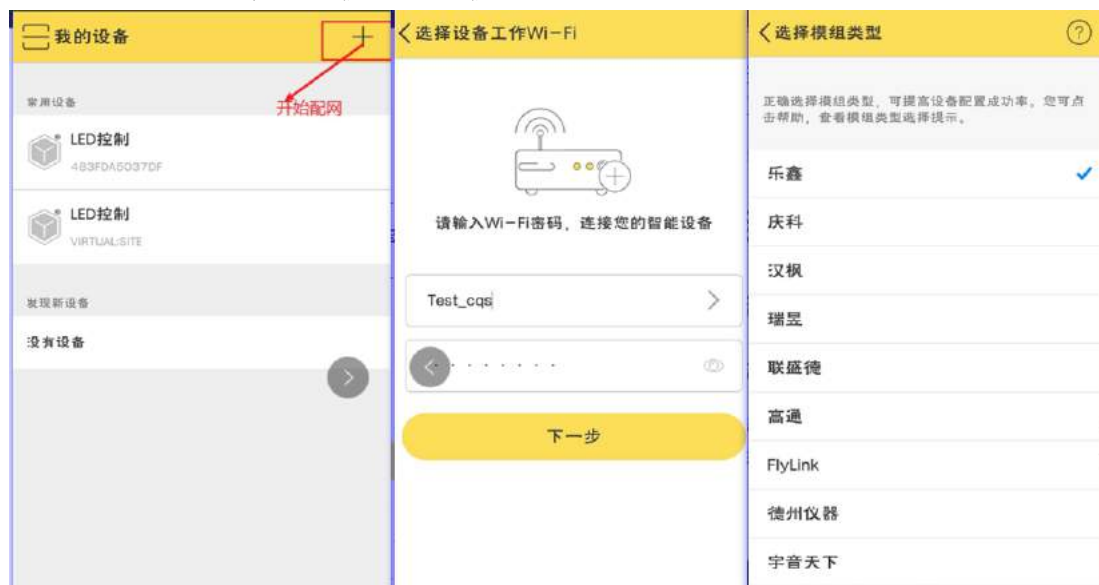


图 5-8 APP 配网流程

当配网成功时，会显示在物联网云平台上面注册的设备名称。点击进入，则会显示相关传感器数据信息，如图 5-9 所示。



图 5-9 APP 显示传感器数据信息

登录物联网平台的后台，选择查看设备日志功能，在设备的运行记录里面，可以看到上报的数据点信息，并支持动态曲线显示，如图 5-10 所示。



图 5-10 设备后台运行日志

---

## 6 总结与展望

基于 STM32 的物联网环境监测系统的整个设计过程中，使用了相关的环境传感器设备去监测周围环境的基本质量参数。在整个设计过程中，利用 LCD 屏幕显示数据信息，能够显示基本传感器测量的数据。由于显示效果一般，后期可以添加嵌入式 GUI 图形界面，例如 LVGL，TouchGFX，emWin。使用无线通信模组与单片机进行数据交互，完成了基本的通信功能，但在模组的选择上需要考虑到其稳定性和功耗问题。在上位机软件程序设计中，使用了 Android 应用程序去完成远程监测的功能，但是 APP 的界面设计需要优化，后期可以添加相关传感器实时动态曲线的功能。

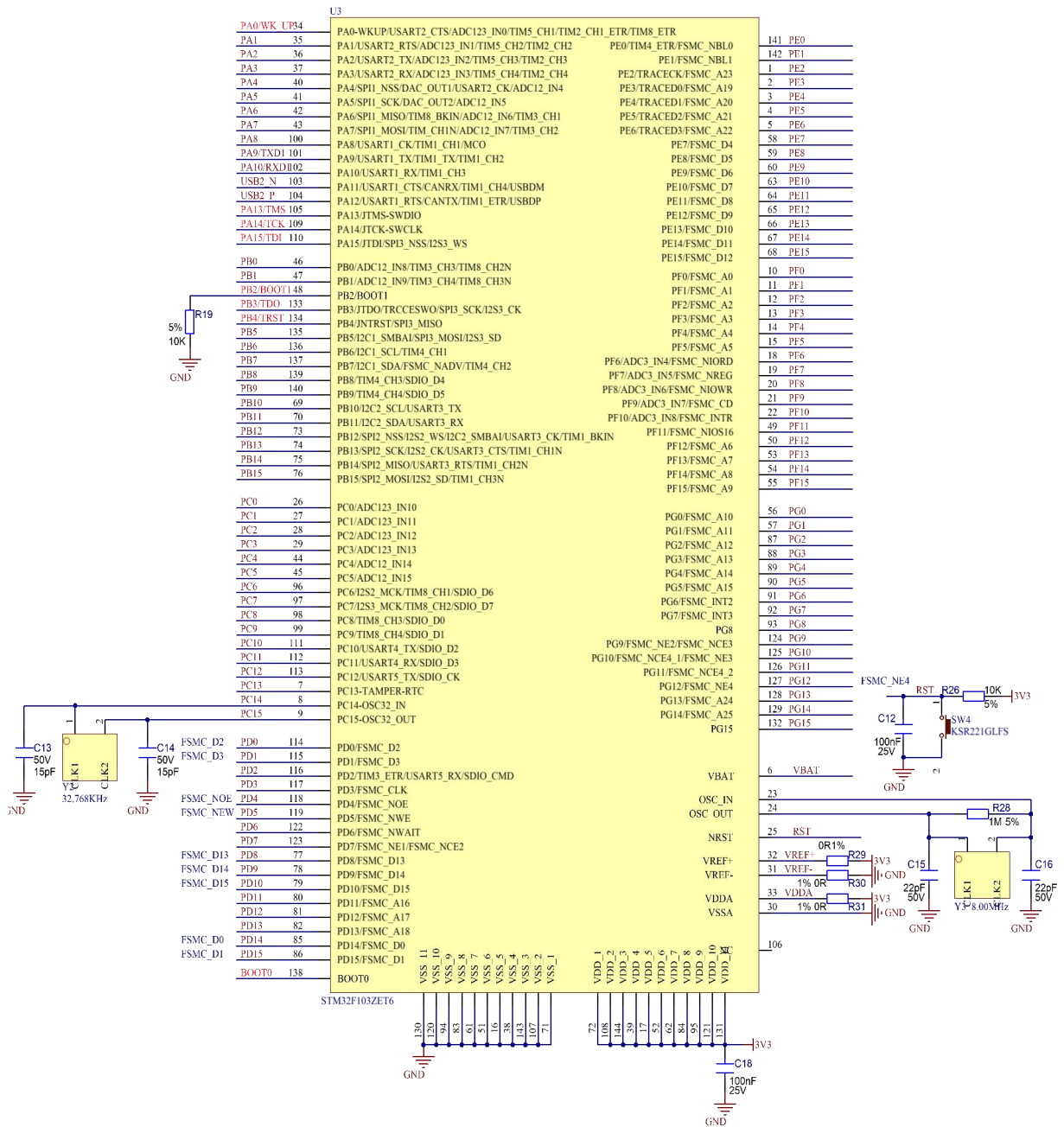
在本次的设计中，虽然初步完成了基本的设计要求，实现了基本物联网通信功能。但是，本设计的硬件方案和传感器选型不够成熟，系统的功耗问题和稳定性还需要去解决。在系统的稳定性上面，后期可以移植 RTOS，减少裸机开发过程中的问题。本设计的使用场景的多样性有待进一步加强。特别是在芯片的选择上面，由于在国内外疫情的反复多变的情况下，半导体行业发展受到了严重打击。芯片出货延期和成本增加，像 STM32 系列芯片的价格疯狂增涨。因此有必要去在国产的芯片平台实现移植。例如在国产的兆易创新 GD32、沁恒 CH32、灵动 MM32、乐鑫科技的 ESP32 等系列国产芯片实现功能移植。

---

## 参 考 文 献

- [1] 曲奉东. 基于纳米尺度金属氧化物异质结构的气体传感器的研究[D]. 吉林:吉林大学,2020.
- [2] Kumar, A. et al. "Environmental Monitoring Systems: A Review." IEEE Sensors Journal 13 (2013): 1329-1339.
- [3] 张恒. 基于 WiFi 的室内环境监测系统设计[D]. 安徽:安徽大学,2019.
- [4] 蒋涛. 基于物联网技术的无线光功率计设计[D]. 浙江:浙江工业大学,2020.
- [5] 孟萌. 基于 Arduino 的物联网数据采集器设计与实现[D]. 北京:北京工业大学,2016.
- [6] 陈宇, 戴亮, 戚蕊,等. 智能环境监测系统设计[J]. 数字技术与应用, 2021, 39(6):157-159.
- [7] 李昂昆, 万彦辉, 黎坤,等. 惯导平台稳定回路的高精度控制方法对比分析[J]. 导弹与航天运载技术, 2018(3):61-68.
- [8] 王伟. 分布式智能照明系统及其控制算法的研究[D]. 四川:电子科技大学,2017.
- [9] 王凯. 物联网在电梯管理中的应用研究[D]. 吉林大学.2016.
- [10] 沈建华, 郝立平. ARM Cortex-M0+微控制器原理与应用:基于 Atmel SAM D20 系列 [M]. 北京航空航天大学出版社, 2014.102-103.
- [11] 赵远超. 基于物联网的温室环境监控系统的设计与实现[D]. 山东:曲阜师范大学,2020.
- [12] 胡炜瑜. 手套机生产车间信息系统的设计与实现[D]. 杭州电子科技大学,2021.
- [13] 杨晨. 基于 ARM11 和 Linux 的室内环境监测系统的设计与实现[D]. 沈阳工业大学.2013.
- [14] 李开放, 金昱芃, 翟天嵩. 肉鸡鸡舍环境监控系统设计[J]. 数码世界, 2020(10):272-273.
- [15] 机智云接入流程[EB/OL].[http://docs.gizwits.com/zh-cn/quickstart/UseMCU\\_BK.html](http://docs.gizwits.com/zh-cn/quickstart/UseMCU_BK.html)
- [16] 王雪. 嵌入式系统中 MODBUS 通信的教学研究与实践[J]. 无线互联科技, 2019, 16(8):1-2.
- [17] 应用程序开发[EB/OL].<http://docs.gizwits.com/zh-cn/AppDev/AndroidSDKA2.html>
- [18] 李强. 基于 Android 和 Zigbee 网络的数据监测系统设计与实现[D].武汉理工大学,2016.

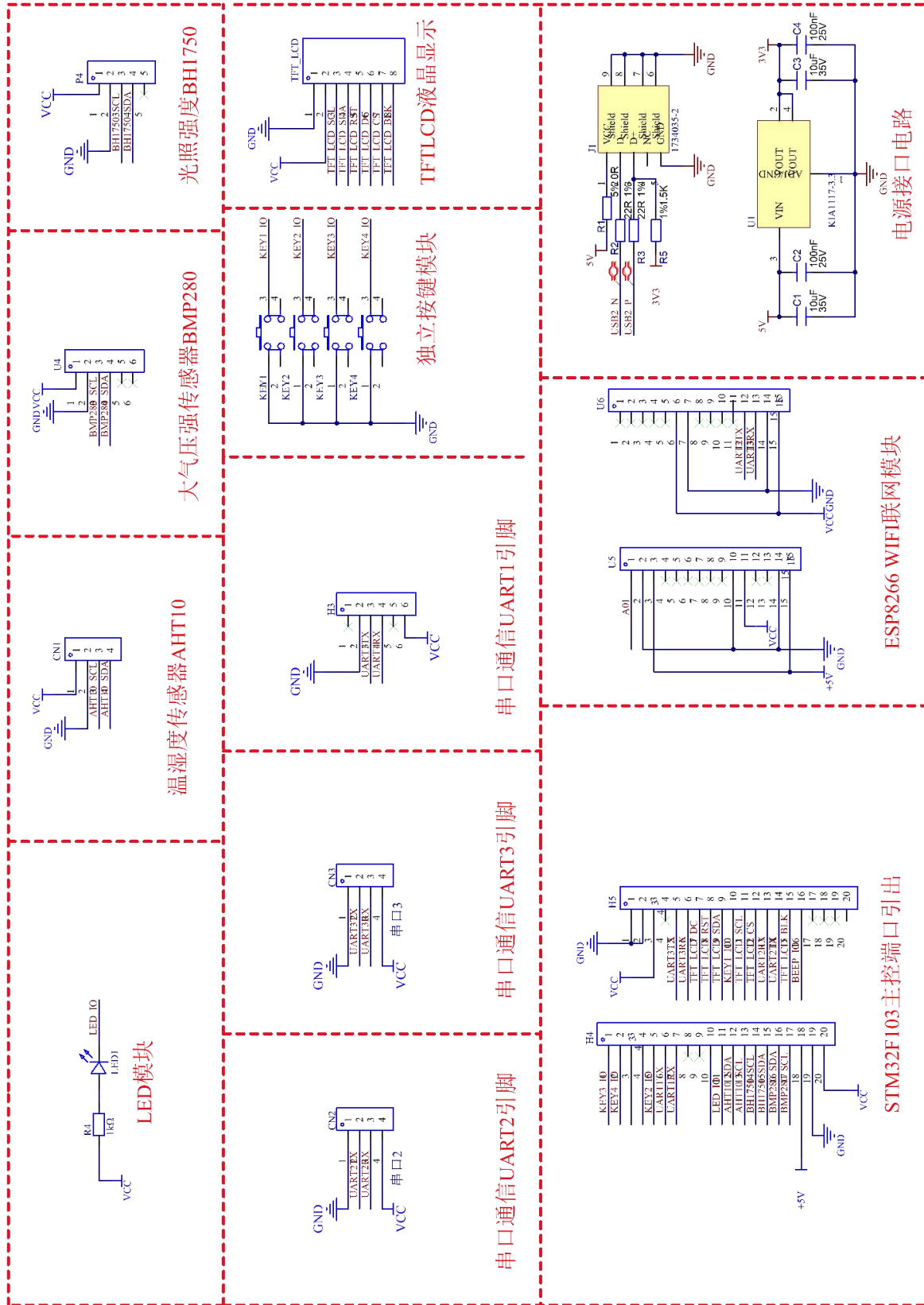
## 附录 A 原理图 A



## STM32F103主控模块

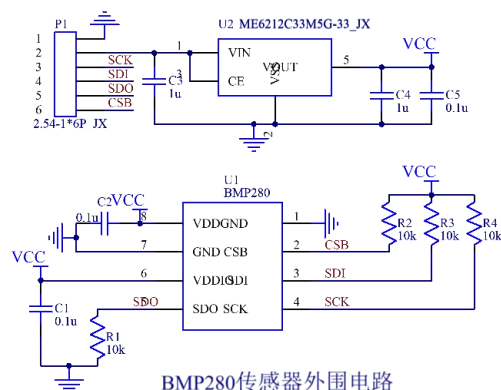


## 附录 B 原理图 B

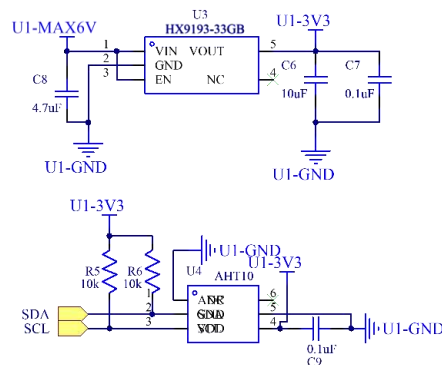




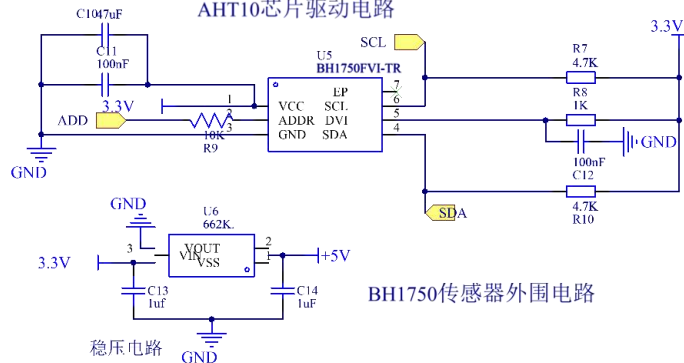
## 附录 C 原理图 C



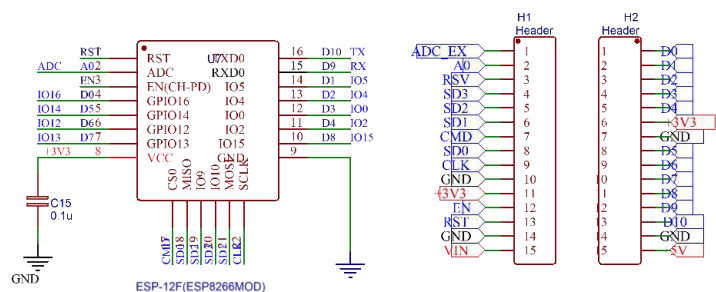
BMP280传感器外围电路



AHT10芯片驱动电路

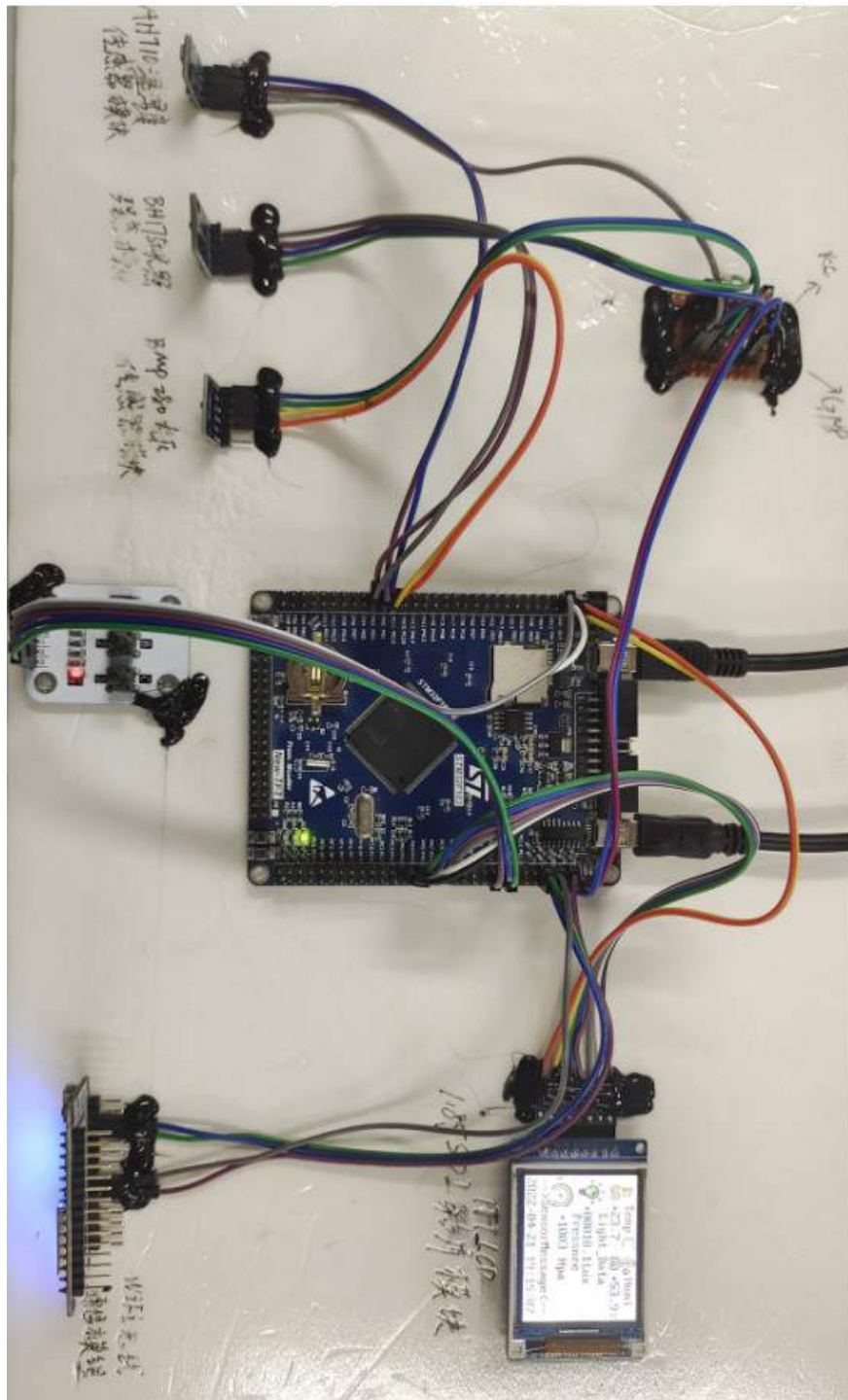


BH1750传感器外围电路



ESP8266 WiFi模块

## 附录 D 实物图



---