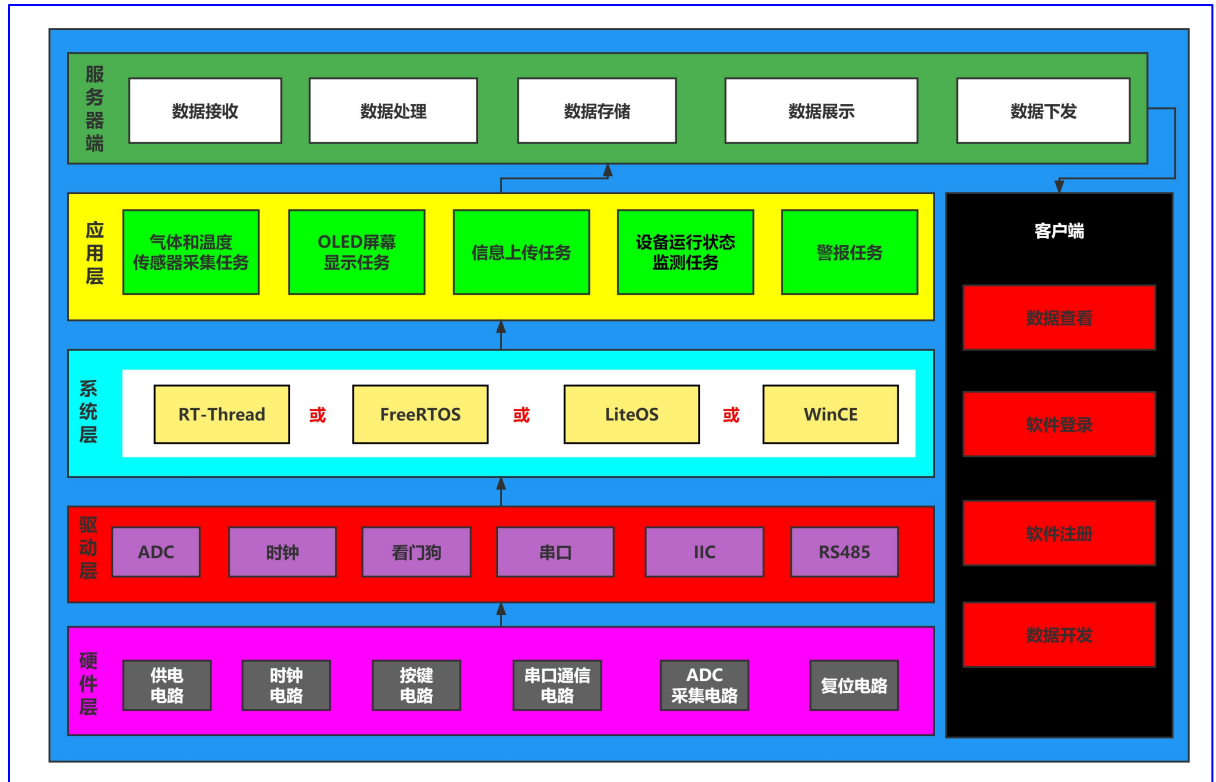
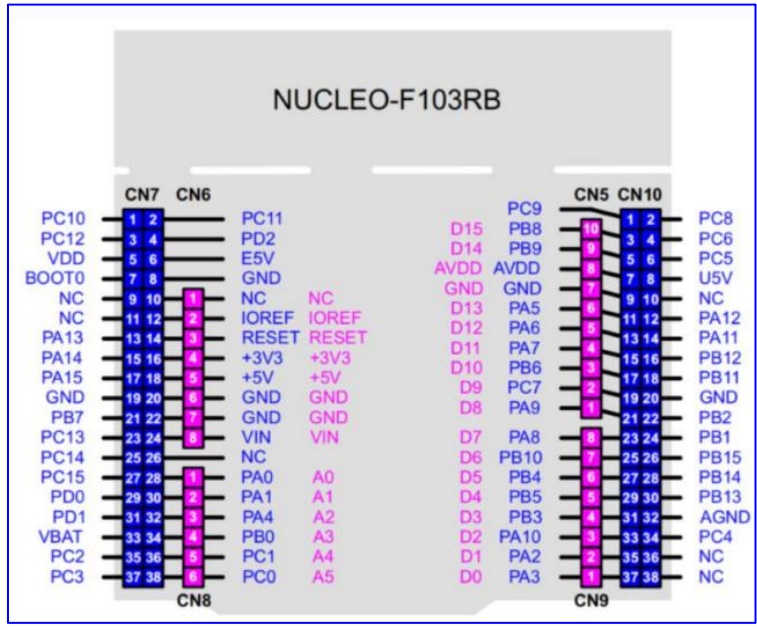


FreeRTOS 单设备版本

系统层次图如下

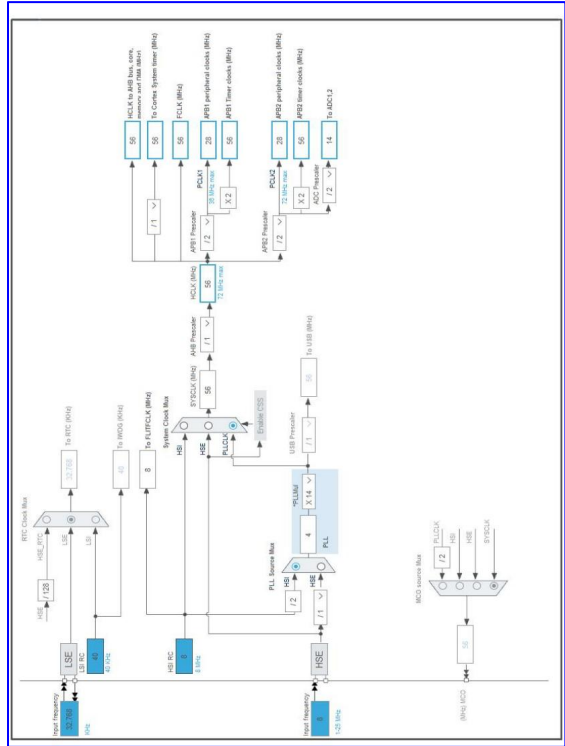


单设备版采用 FreeRTOS 实时操作系统，使用 NucleoF103RB 系列开发板作为主控，串口 1 连接 Esp8266 连接阿里云物联网平台，串口 3 与 PC 通信，串口 3 进行 ST-Link 进行调试。ADC1 采集气体传感器、温度传感器、内部参考电压等模拟值。



Pin Number LQFP64	Pin Name (function after reset)	Pin Type	Alternate Function(s)	Label
1	VBAT	Power		
2	PC13-TAMPER-RTC	I/O	GPIO_EXTI13	B1 [Blue PushButton]
3	PC14-OSC32_IN	I/O	RCC_OSC32_IN	
4	PC15-OSC32_OUT	I/O	RCC_OSC32_OUT	
5	PD0-OSC_IN	I/O	RCC_OSC_IN	
6	PD1-OSC_OUT	I/O	RCC_OSC_OUT	
7	NRST	Reset		
12	VSSA	Power		
13	VDDA	Power		
16	PA2	I/O	USART2_TX	USART_TX
17	PA3	I/O	USART2_RX	USART_RX
18	VSS	Power		
19	VDD	Power		
20	PA4	I/O	ADC1_IN4	
21	PA5 *	I/O	GPIO_Output	LD2 [Green Led]
22	PA6	I/O	ADC1_IN6	
23	PA7	I/O	ADC1_IN7	
26	PB0	I/O	ADC1_IN8	
29	PB10	I/O	USART3_TX	
30	PB11	I/O	USART3_RX	
31	VSS	Power		
32	VDD	Power		
42	PA9	I/O	USART1_TX	USART1_TX
43	PA10	I/O	USART1_RX	USART1_RX
46	PA13	I/O	SYS_JTMS-SWDIO	TMS
47	VSS	Power		
48	VDD	Power		
49	PA14	I/O	SYS_JTCK-SWCLK	TCK
55	PB3 **	I/O	SYS_JTDO-TRACESWO	SWO
60	BOOT0	Boot		
63	VSS	Power		
64	VDD	Power		

时钟配置



7.2. RCC

High Speed Clock (HSE): BYPASS Clock Source

Low Speed Clock (LSE) : Crystal/Ceramic Resonator

7.2.1. Parameter Settings:

System Parameters:

VDD voltage (V)	3.3
Prefetch Buffer	Enabled
Flash Latency(WS)	2 WS (3 CPU cycle)

RCC Parameters:

HSI Calibration Value	16
HSE Startup Timeout Value (ms)	100
LSE Startup Timeout Value (ms)	5000

工具链相关

Name	Value
Project Name	MX_Pig_V1.0
Project Folder	F:\code\MX_Pig_V1.0
Toolchain / IDE	MDK-ARM V5.27
Firmware Package Name and Version	STM32Cube FW_F1 V1.8.4
Application Structure	Advanced
Generate Under Root	No
Do not generate the main()	No
Minimum Heap Size	0x200
Minimum Stack Size	0x400

硬件相关

电源设计

考虑到电路的工作效率和输出功率等因素，本项目中选择开关电源作为供电方式，开关电源是一种比较新型的电源。它具有效率高，重量轻，可升、降压，输出功率大等优点。LM2596 是降压型电源管理单片集成电路的开关电压调节器，能够输出 3A 的驱动电流，同时具有很好的线性和负载调节特性。

LM2596 有两种封装形式，5 脚 TO-220(T); TO-263(S), TO-263(S)封装的 LM2596 是要焊接在 PCB 板上的表贴元件。其典型应用原理图如下。

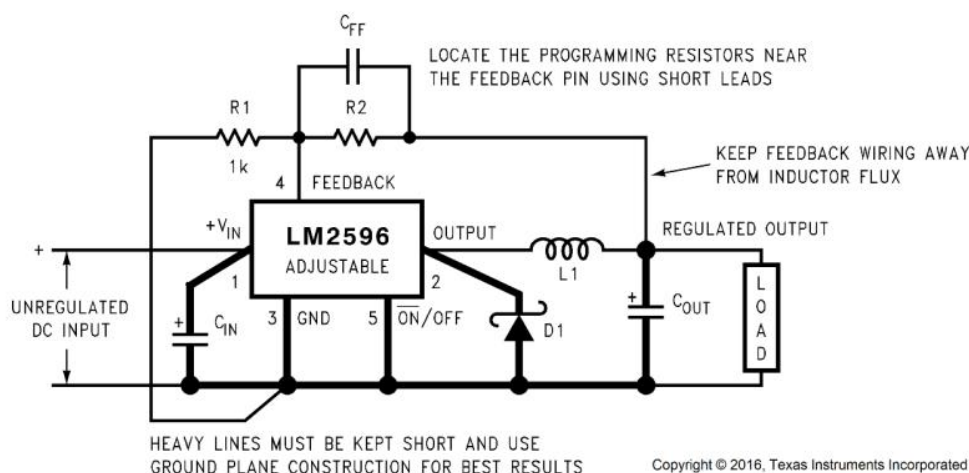


图 1 LM2596 可调输出电压版本

其计算如下：

$$V_{OUT} = V_{REF} \left(1 + \frac{R_2}{R_1} \right) \quad V_{REF} = 1.23V$$

$$R_2 = R_1 \left(\frac{V_{OUT}}{V_{REF}} - 1 \right)$$

为了获得最佳稳定性，R1 应使用 1%电阻，阻值约为 1KΩ，，输入电容 CIN 应为 470uF，50V 铝电解电容，输出电容 COUT 应为 220uF，35V 的铝电解电容。二极管应选用 40V 肖特基整流型，电感 L1 位 68uH。

如图 1.2，对于 3V3 固定输出电路：采用 LM2596S-3.3 基本电路将 6-12v 输入电压转化为固定 3.3v 输出，输出电压稳定，电流稳定，理论输出能达到 3A。对于 5v 固定输出电路：采用 LM2596S-5.0 作为电压降压稳压芯片，与 3.3v 电路原理一致。

输入供电主要有两节 18650 电池提供，18650 型锂离子电池具有能量密度大、无记忆、使用寿命长等特点。可以长期不用更换电池，给设备带来了更好的稳定性。同时还可以使用 DC 电源插座，由外部供电，可以避免电池电量带来的问题。

电压输出端提供了两种不同的接口，两个 4×4 排针和两个螺钉式接线端子，接口的数量，可以给后续电路模块提供多路供电。接口的丰富性满足了不同情况下的使用，适用于大电流和大电压，满足不同温度情况下的使用。

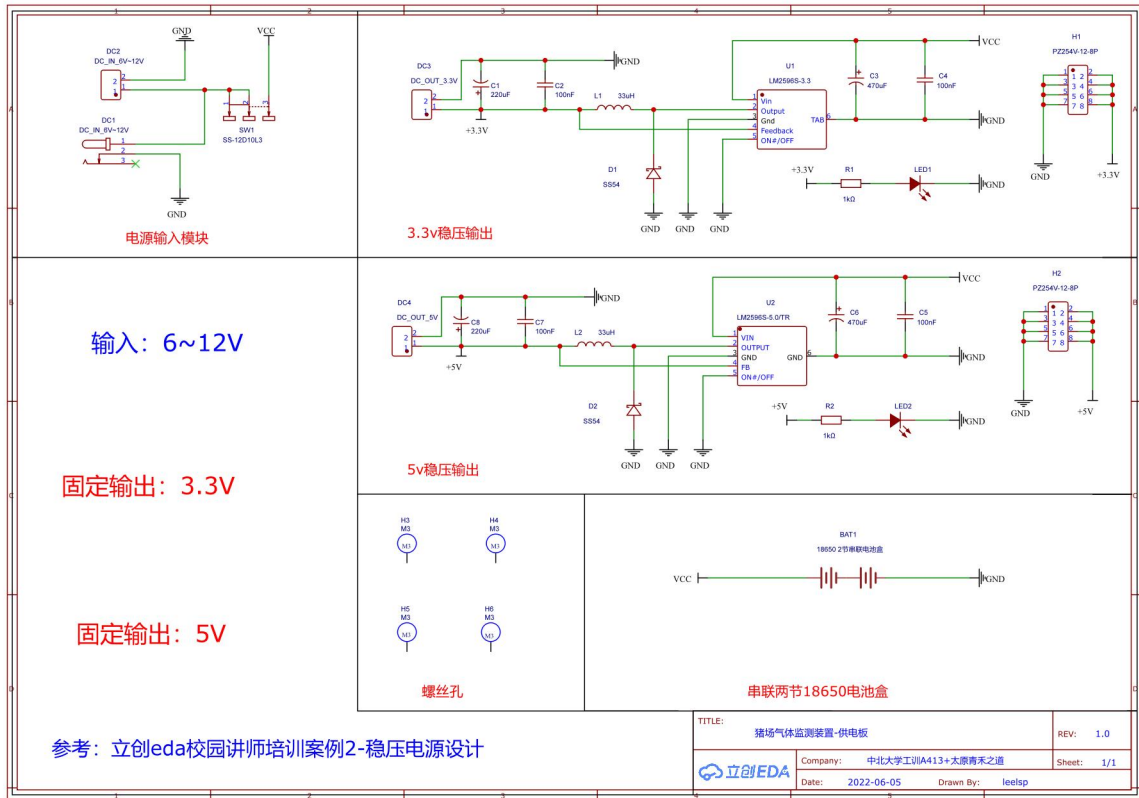


图2 电源设计原理图

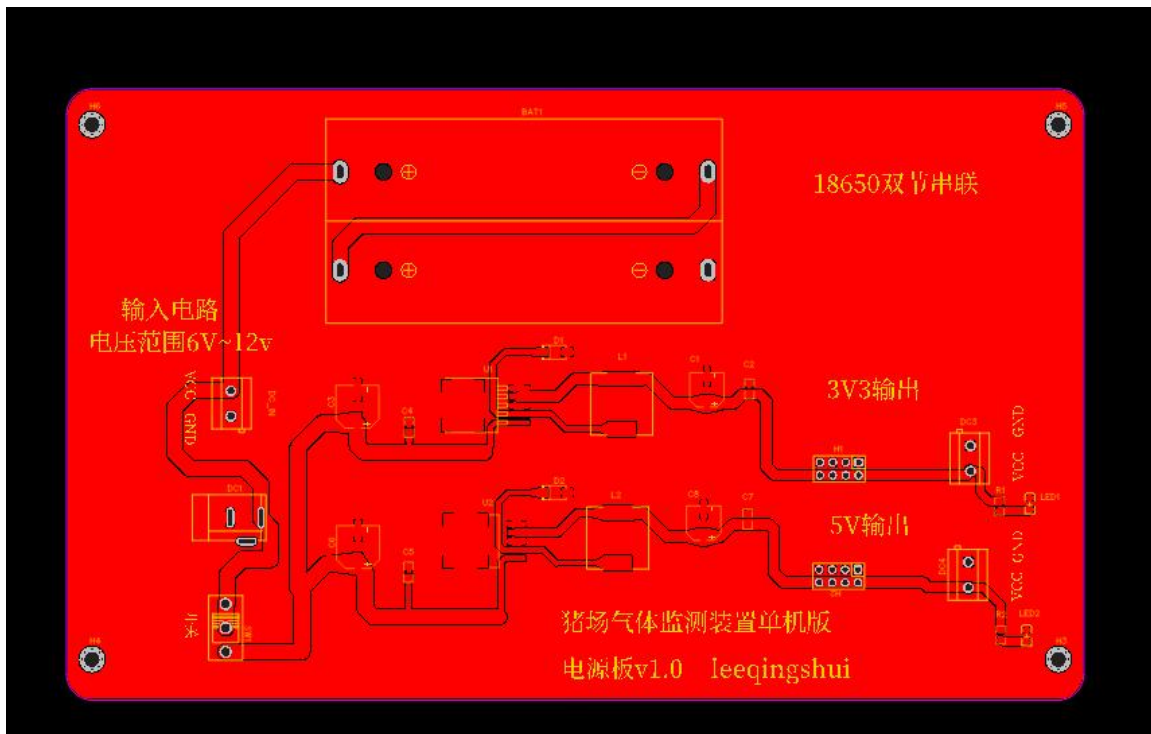


图3 电源设计 PCB 图

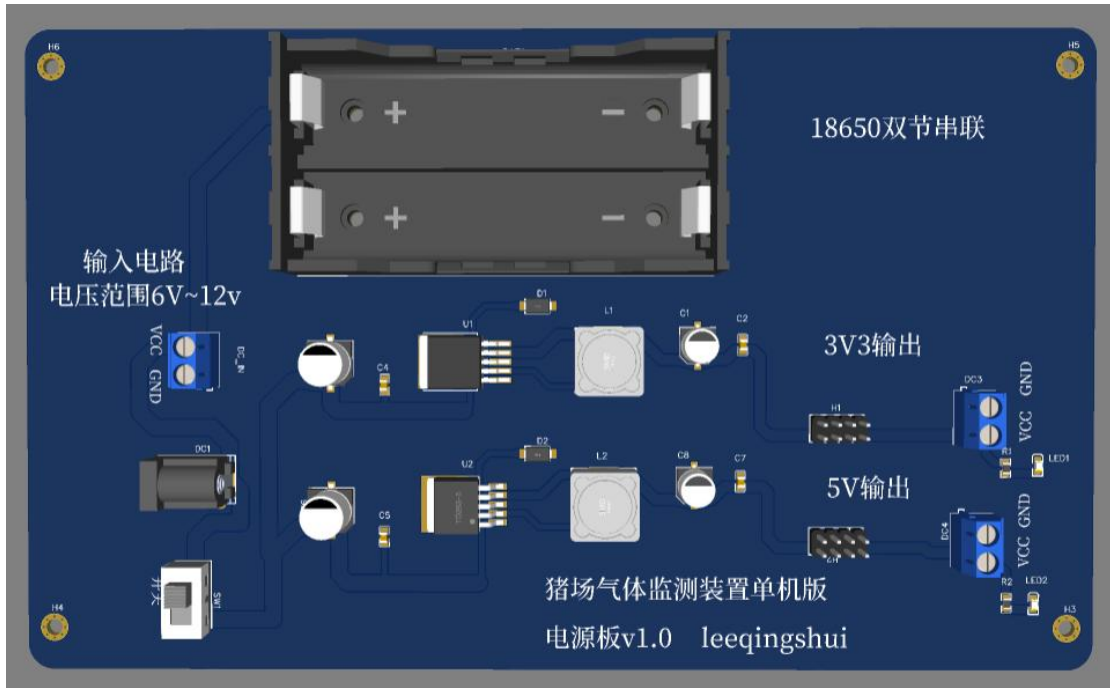


图 4 供电模块 3D 视图（正面）



图 5 供电模块 3D 视图（背面）

传感器采集板设计

采集板由供电板提供稳定 5V 供电。

气体传感器模块采用 MQ 气体传感器系列。检测有毒气体和污染气体始终与我们息息相关，气体传感器执行此操作，有些气体传感器非常昂贵，幸运的是，MQ 气体传感器系列，提供了一种经济高效的气体传感器，可帮助从特定区域检测特定的气体排泄物。气体传感器的尺寸（便携式和固定式），范围和感应能力差异很大。它们通常是较大的嵌入式系统（例如安全系统）的一部分，并且通常连接到声音警报或接口。由于气体传感器不断与空气和其他气体相互作用，因此与许多其他类型的传感器相比，它们必须更频繁地进行校准。

以 MQ137 氨气传感器说明采集原理，MQ137 气体传感器所使用的气敏材料是在清洁空气中电导率较低的二氧化锡(SnO_2)。当传感器所处环境中存在氨气时，传感器的电导率随空气中氨气浓度的增加而增大。使用简单的电路即可将电导率的变化转换为与该气体浓度相对应的输出信号。

气体传感器 ADC 分压采集，由于原输出 0~5V，使用电阻分压，将范围缩小到 0~2.5V，但有一些问题：如果外部信号超过了 AD 测量范围，可以采用电阻分压的方法，但是要注意阻抗匹配。举个简单的例子，比如说你的 ADC 参考电压是 2.5V，要测量一个 0~5V 的电压。因为输入的电压最大值大于了 ADC 参考电压，所以需要电阻把输入电压分压，最简单的就是用两个 10K 的电阻串联，即 +V->R1->R2->GND，这样 ADC 在 R1 和 R2 中间取信号，那就是输入电压的 1/2，但要考虑 ADC 的输入阻抗 R_a ，假设是 20k，前面的电路情况变成 +V->R1->R2|| R_a ->GND， $R_2||R_a$ 表示两个电阻的并联，并联之后的电阻肯定小于 10K，那分压就不是 1/2 了，即产生了误差。因此用运放构成电压跟随器，其输入阻抗在 $M\Omega$ 以上，和 10K 并联时造成的电阻变化是很小的，所以保证分压比很接近 1/2。同时跟随器输出电阻很小，只有几十欧姆，和 ADC 输入电阻相比相差很大，所以 ADC 的到的信号电压近似等于跟随器的输出，即减少了误差。

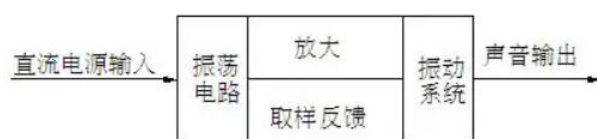
电阻分压后用 ADC 采样，因为电阻是有误差的，因此会带来测量误差。解决误差的方式有：高精度电阻+软件校准+高精度基准。有软件校准，采样电阻精度再差也没关系，温漂和老化才是重要指标。

温湿度传感器选用的一款湿温度一体化的数字传感器。该传感器包括一个电阻式测湿元件和一个 NTC 测温元件。通过单片机等微处理器简单的电路连接就能够实时的采集本地湿度和温度。DHT11 与单片机之间能采用简单的单总线进行通信，仅仅需要一个 I/O 口。传感器内部湿度和温度数据 40Bit 的数据一次

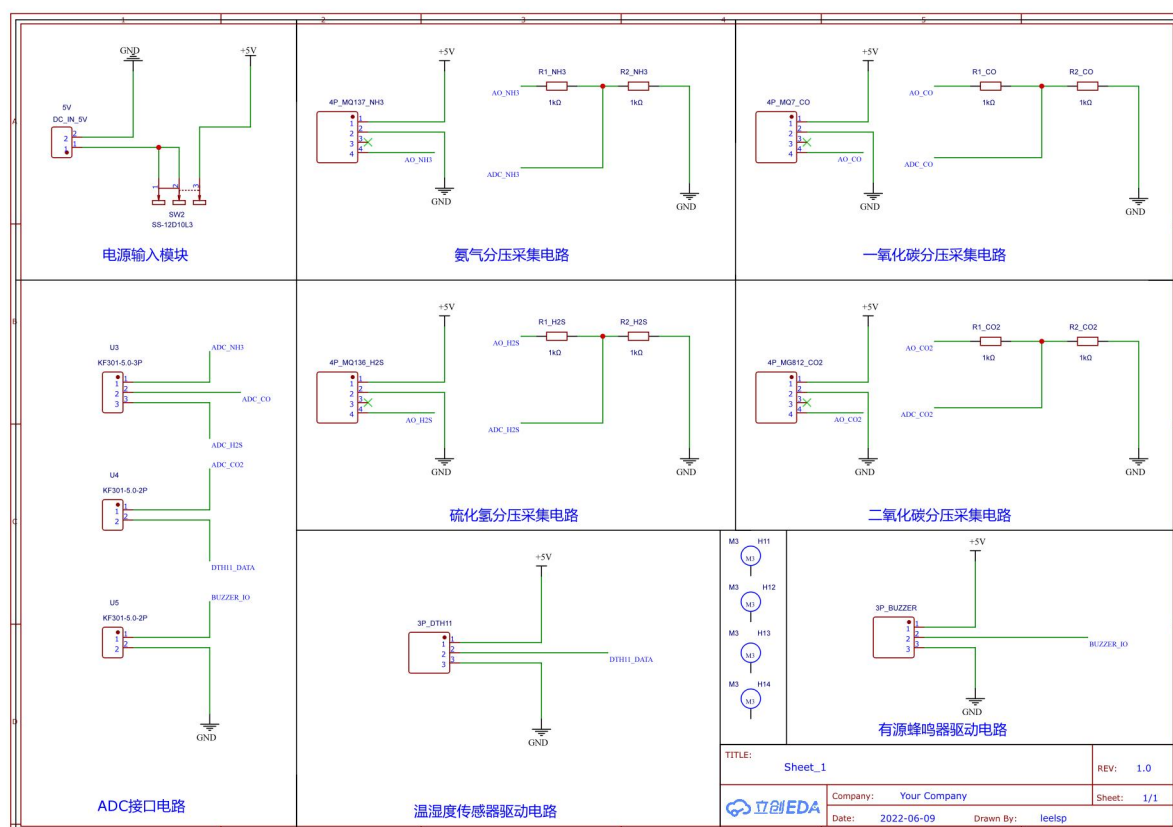
性传给单片机，数据采用校验和方式进行校验，有效的保证数据传输的准确性。DHT11 功耗很低，5V 电源电压下，工作平均最大电流 0.5mA。

报警电路选择有源蜂鸣器模块，内含驱动线路,无需外部驱动。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声音信号，有源蜂鸣器的工作发声原理图如图 2.1，只要一通电就会叫。

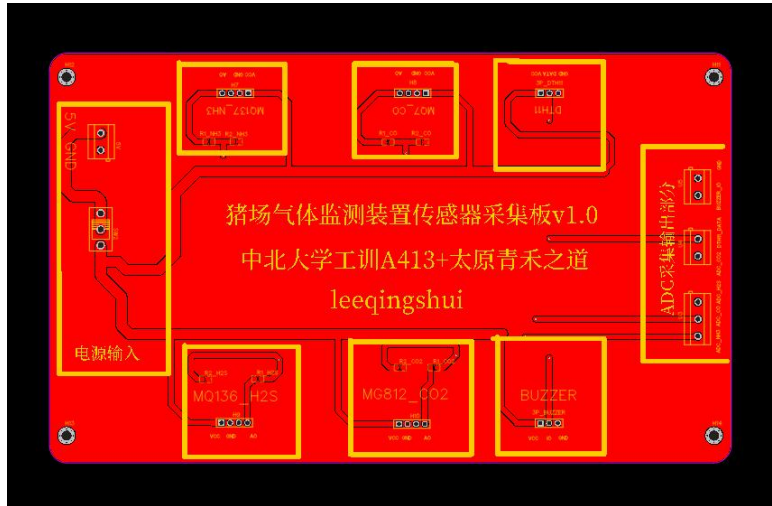
同时传感器都选用 4P 或者 3P 排母连接，适应对电磁干扰引起的衰减，以及对电磁干扰屏蔽的才能等。在可靠性方面要比相应的 SMT 元件高许多。无论是剧烈的拉拽、揉捏或热冲击，它都能承受，而不易脱离 PCB。可满足于不同的传输高电压及大电流的需求。



有源蜂鸣器发声原理图



传感器采集板原理图



传感器采集板 PCB 图



传感器采集板 3D 图（正面）



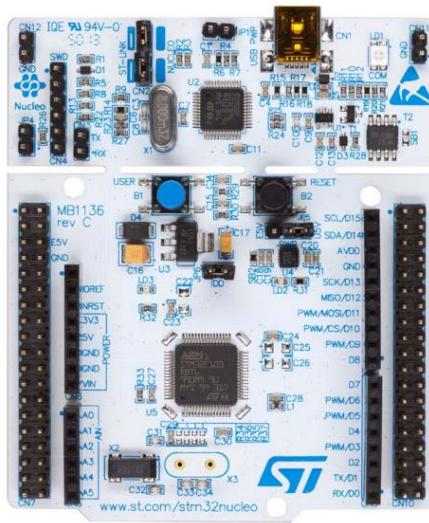
传感器采集板 3D 图（背面）

主控扩展板设计

主控扩展板由供电板提供稳定 5V 供电。

主控扩展板主要为了满足主控和外设功能的扩展性，主控选择 Nucleo-f103 开发板，其具有灵活的供电选项：ST-LINK、USB VBUS 或外部电源；具有 USB 重新枚举功能的板上 ST-LINK 调试器/编程器；大容量存储器、虚拟 COM 端口和调试端口；提供了全面的免费软件库和例程，可从 STM32Cube MCU 软件包获得支持多种集成开发环境（IDE），包括 IAR Embedded Workbench®、MDK-ARM，以及 STM32CubeIDE。为用户提供了一种可负担的灵活方法。

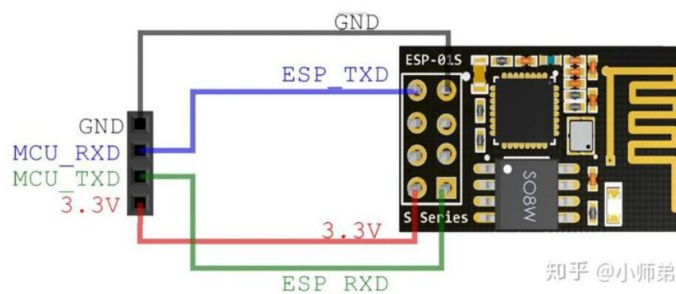
作为主控制器，监测养猪场硫化氢、氨、二氧化碳、一氧化碳浓度、温度、湿度和设备电压。



Nucleo-f103 开发板实物图

ESP8266 模块采用 MQTT 协议连接阿里云物联网平台，从而 Nucleo-f103 借助 ESP8266 进行数据实时传输和处理。

ESP8266 是一个完整且自成体系的 WiFi 网络解决方案，能够独立运行，也可以作为 slave 搭载于其他 Host 运行。ESP8266 在搭载应用并作为设备中唯一的应用处理器时，能够直接从外接闪存中启动。内置的高速缓冲存储器有利于提高系统性能，并减少内存需求。另外一种情况是，无线上网接入承担 WiFi 适配器的任务时，可以将其添加到任何基于微控制器的设计中，连接简单易行，只需通过 SPI/SDIO 接口或中央处理器 AHB 桥接口即可。ESP8266 强大的片上处理和存储能力，使其可通过 GPIO 口集成传感器及其他应用的特定设备，实现了最低前期的开发和运行中最少地占用系统资源。



ESP8266 连接图

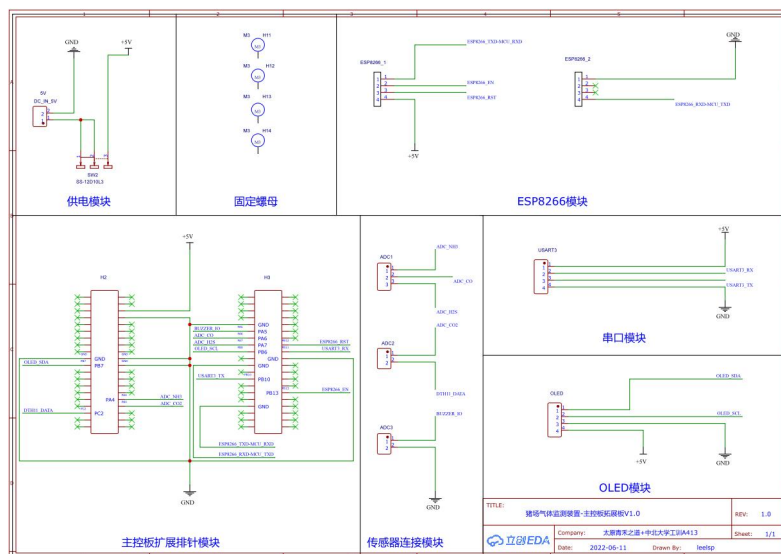
OLED 采用 0.96 寸 OLED 模块。由于同时具备自发光，不需背光源、对比度高、厚度薄、视角广、反应速度快、可用于挠曲性面板、使用温度范围广、构造及制程较简单等优异之特性，被认为是下一代的平面显示器新兴应用技术。

有单色和双色两种可选，单色为纯蓝色，而双色则为黄蓝双色。单色模块每个像素点只有亮与不亮两种情况，没有颜色区分；尺寸小，显示尺寸为 0.96 寸，而模块的尺寸仅为 27mm*26mm 大小；高分辨率，该模块的分辨率为 128*64；多种接口方式，该模块提供了总共 4 种接口包括：6800、8080 两种并行接口方式、4 线的穿行 SPI 接口方式、IIC 接口方式；不需要高压，直接接 3.3V 就可以工作了。

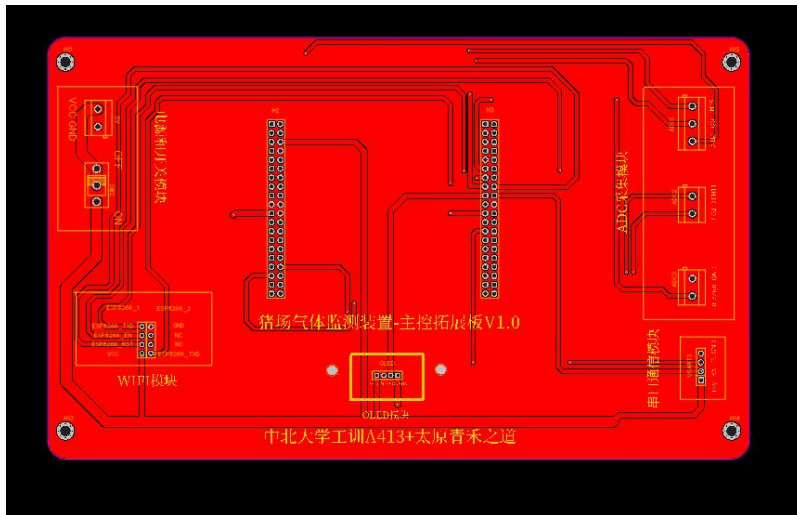
序号	模块引脚	功能
1	VCC	电源正, 3.3V-5V
2	GND	电源地
3	SCL	IIC总线时钟信号
4	SDA	IIC总线数据信号

OLED 引脚配置图

串口通信模块直接通过 4P 排针连接外部设备，用于获取远程采集设备的数据。传感器采集模块通过螺钉式接线端子连接外部设备。



主控扩展板原理图



主控扩展板 PCB 图

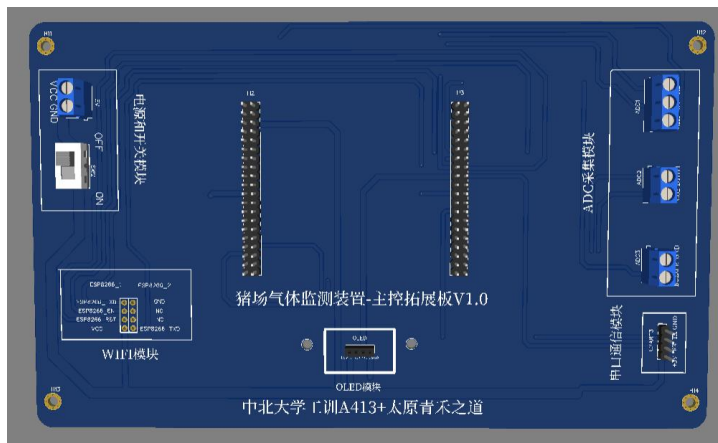


图 3.6 主控扩展板 3D 图（正面）

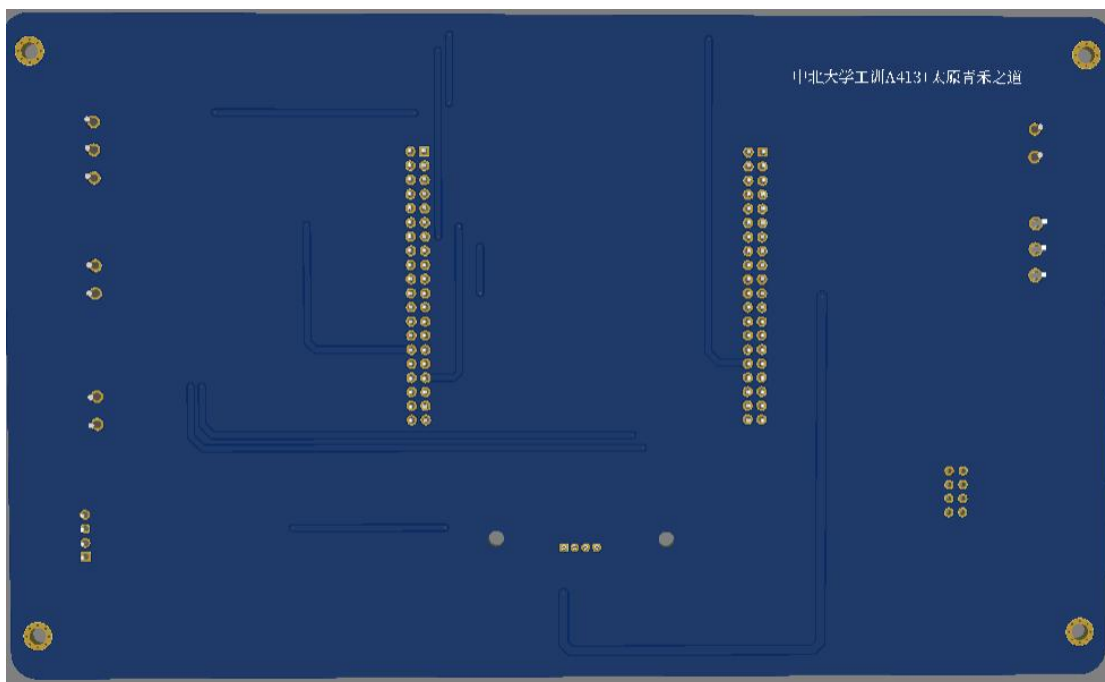


图 3.7 主控扩展板 3D 图（背面）

程序部分

1.1 简介:

Nucleo-f103rb 开发板作为主控制器，监测硫化氢、氨、二氧化碳、养猪场一氧化碳浓度、温度、湿度和设备电压通过气体传感器、温度以及湿度传感器和内部电压传感器。ESP826601S 模块采用 MQTT 协议连接阿里云物联网平台，实现数据的实时显示和处理。

1.2.代码介绍:

软件分为三层:硬件层、外设驱动层(硬件文件夹)、系统层(任务文件夹)

(1) 硬件层

芯片外设初始化，由 STM32CubeMX 自动生成

(2) 外围驱动层

包含外部模块的初始化和相关操作功能(ADC1\DTH11\ESP8266\蜂鸣器\OLED)

(3) 系统层

包括创建任务和消息队列以及通过消息队列在任务之间通信的功能

1.3. 工作过程:

(1) 芯片外设初始化

(2) 初始化外部模块

(3) 开启任务调度功能

组件的相关引脚及配置

2.1.MCU: NUCLEO-F103RB

时基源: TIM1

RCC BYPASS Clock Source RCC_OSC_IN PD0-OSC_IN

RCC BYPASS Clock Source RCC_OSC_OUT PD1-OSC_OUT

RCC Crystal/Ceramic Resonator RCC_OSC32_IN PC14-OSC32_IN

RCC Crystal/Ceramic Resonator RCC_OSC32_OUT PC15-OSC32_OUT

SYS Serial Wire SYS_JTCK-SWCLK PA14
SYS Serial Wire SYS_JTMS-SWDIO PA13

2.2. OLED :

OLED0561_ADD 0x78

I2C1 I2C I2C1_SCL PB6

I2C1 I2C I2C1_SDA PB7

主功能:

I2C Speed Mode 标准模式

I2C 时钟速率(Hz) 100000

特点:

Clock No Stretch Mode 禁用

主地址长度选择 7 位

双地址确认禁用

主从地址 0

通用呼叫地址检测禁用

2.3.ESP8266:

USART1

USART1 Asynchronous USART1_RX PA10

USART1 Asynchronous USART1_TX PA9

引脚使用:

PB12 GPIO_Output ESP8266_RST

PB13 GPIO_Output ESP8266_CH_PD

基本参数:

波特率 115200

字长 8 位(包括奇偶校验)

奇偶校验没有

停止位 1

先进的参数:

数据方向接收和发送

超过 16 个样本

2.4.USART3-USB T0 TTL:

引脚使用:

USART3 Asynchronous USART3_RX PB11

USART3 Asynchronous USART3_TX PB10

基本参数:

波特率 115200

字长 8 位(包括奇偶校验)

奇偶校验没有

停止位 1

先进的参数:

数据方向接收和发送

超过 16 个样本

2.5. DTH11-Data:

PC2 GPIO_Output DTH11

2.6.Buzzer && LED2:

PA5 GPIO_Output

2.7.ADC:

ADC1 IN4 ADC1_IN4 PA4

ADC1 IN6 ADC1_IN6 PA6

ADC1 IN7 ADC1_IN7 PA7

ADC1 IN8 ADC1_IN8 PB0

Channel1 DMA 请求设置:

模式:循环*

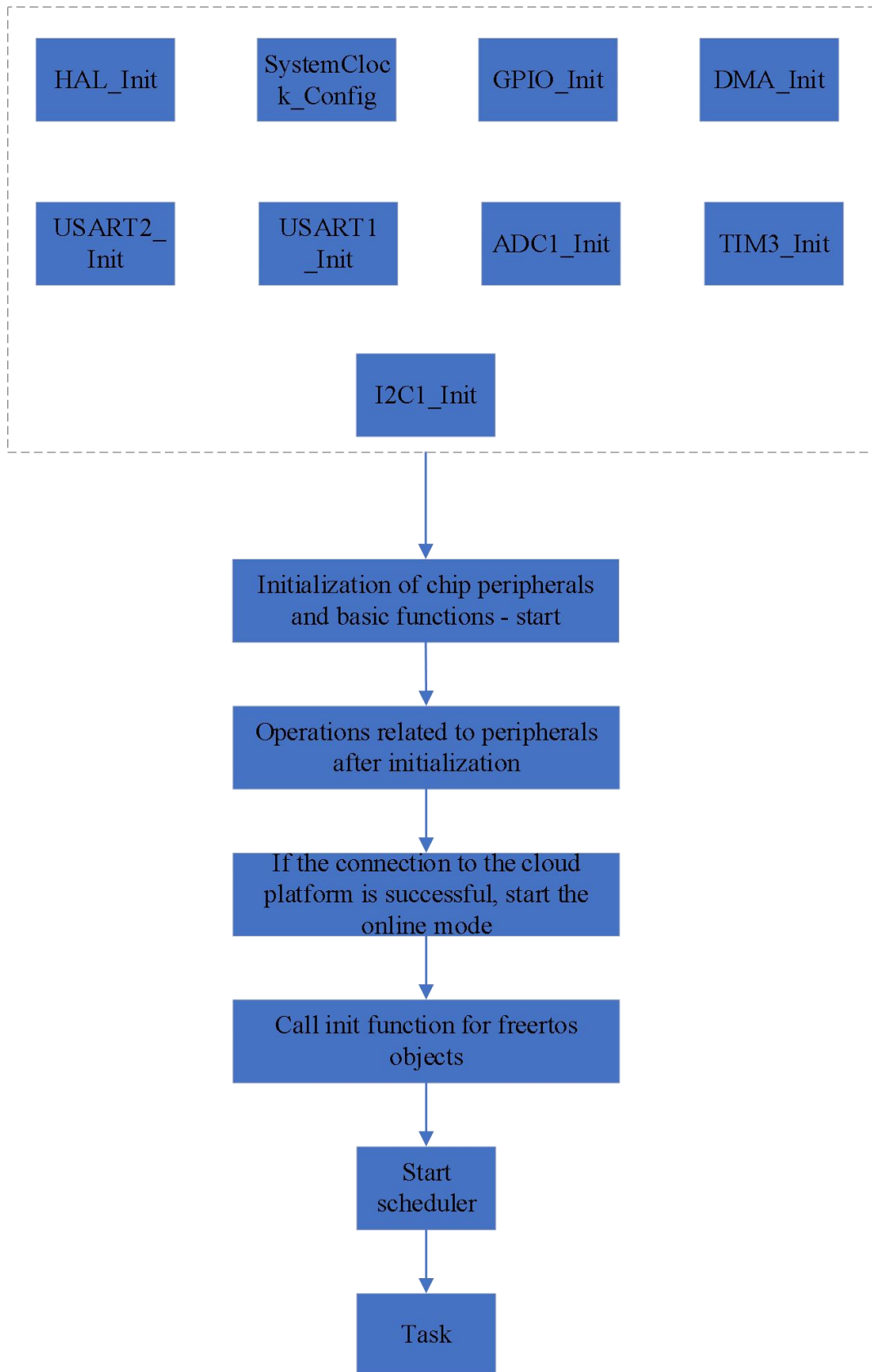
外围增加:禁用

内存增量:启用*

外围数据宽度:半字

内存数据宽度:半字

Software structure diagram



主要过程

HAL_Init → SystemClock_Config → GPIO_Init → DMA_Init → USART2_Init → USART1_Init → USART3_Init → ADC1_Init → TIM3_Init → I2C1_Init → Initialization of chip peripherals and basic functions - start → Operations related to peripherals after initialization → If the connection to the cloud platform is successful, start the online mode → Call init function for freertos objects (in freertos.c) → Start scheduler → Task.c

文件调用关系

文件介绍:

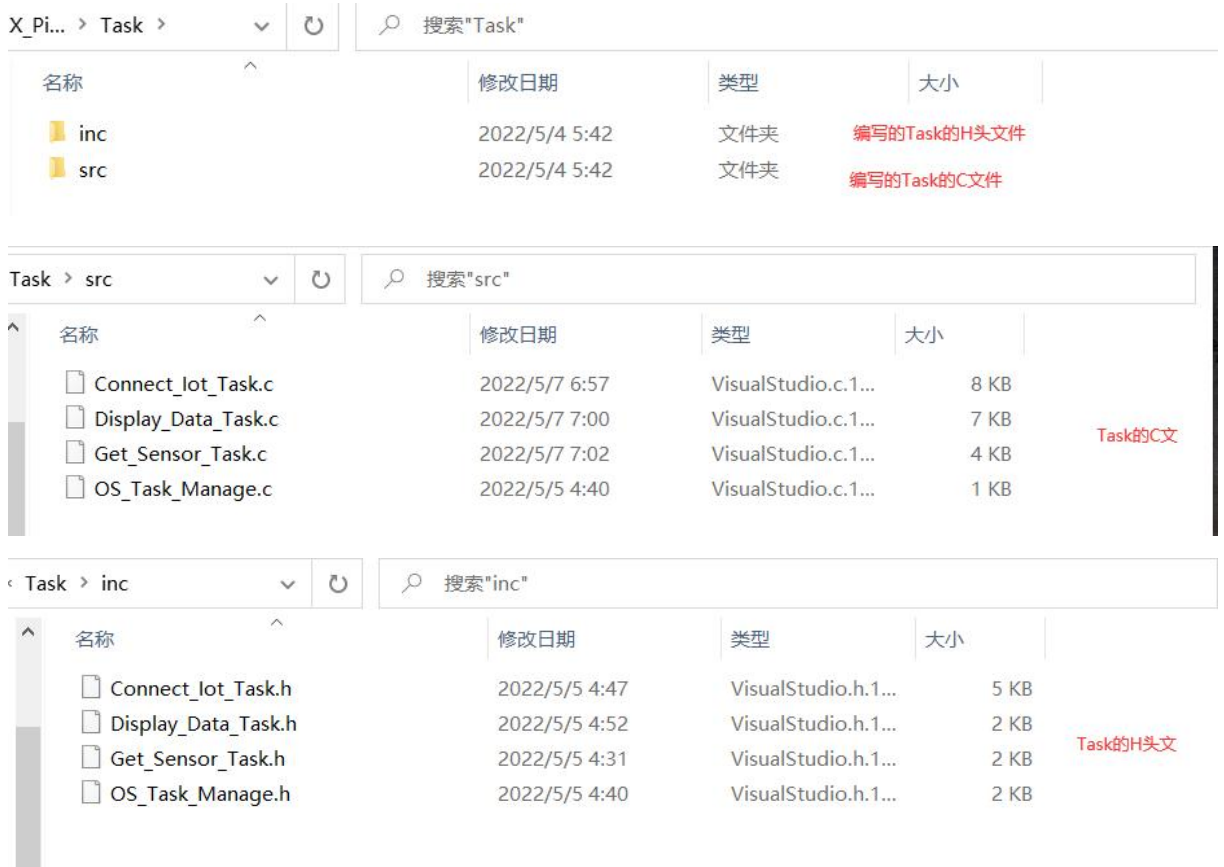
整个程序文件夹:

Common	2022/4/19 21:16	文件夹	
Core	2022/4/19 20:06	文件夹	底层文件
Doc	2022/4/30 17:02	文件夹	程序使用指南
Drivers	2022/4/19 20:06	文件夹	驱动程序
Hardware	2022/4/20 8:24	文件夹	编写的应用程序
MDK-ARM	2022/6/28 22:36	文件夹	工程文件
Middlewares	2022/4/19 20:06	文件夹	操作系统相关文件
Task	2022/5/2 0:55	文件夹	

文件夹	Task 文件	包含 Task 任务的相关 C 文件以及头文件
	Middlewares 文件	包含操作系统的相关文件
	MDK-ARM 文件	包含工程文件，我们可以直接打开工程文件对所编写的程序进行操作，由 Keil5 编译器进行编写
	Hardware 文件	包含需要用到的外设所编写的程序文件，包括 C 文件以及 H 头文件
	Drivers 文件	包含 HAL 库的驱动文件
	Core 文件	包含开发板相关库的驱动文件
	Common 文件	包含我们对数据类型编写的一个规范文件

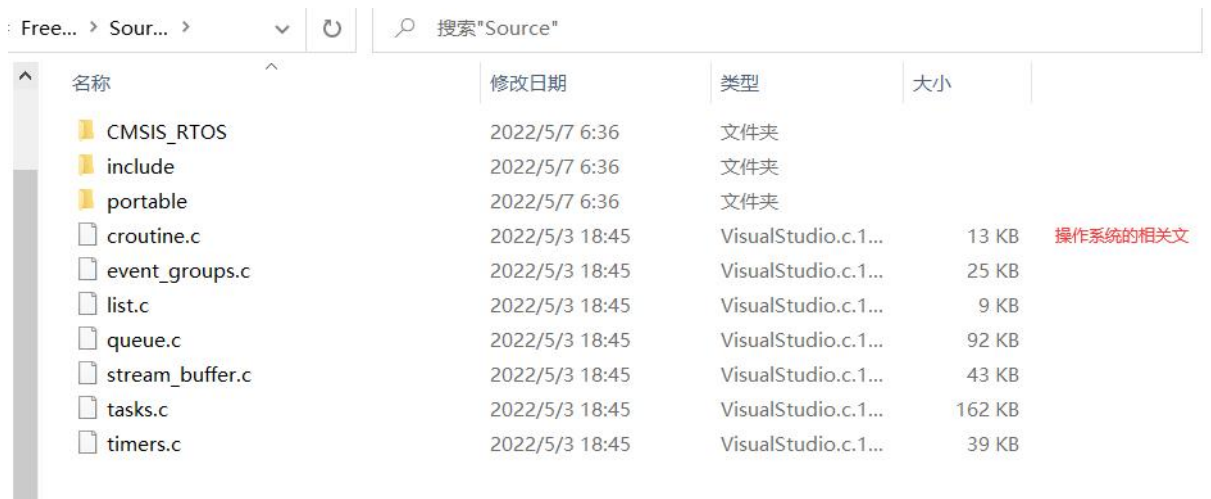
Task 文件介绍:

文件夹中包含 Task 任务的相关 C 文件以及头文件。



Middlewares 文件介绍:

文件夹中包含操作系统的相关文件。



MDK-ARM 文件介绍:

文件夹中包含工程文件，我们可以直接打开工程文件对所编写的程序进行操作，由 Keil5 编译器进行编写。

名称	修改日期	类型	大小
DebugConfig	2022/6/28 20:37	文件夹	
MX_Pig_V1.0	2022/5/7 6:51	文件夹	
RTE	2022/4/19 20:17	文件夹	
EventRecorderStub.scvd	2022/4/20 10:52	SCVD 文件	1 KB
MX_Pig_V1.0.uvguix.lee	2022/5/7 7:55	LEE 文件	177 KB
MX_Pig_V1.0.uvguix.lx	2022/6/28 22:36	LX 文件	90 KB
MX_Pig_V1.0.uvoptx	2022/6/28 22:36	UVOPTX 文件	30 KB
MX_Pig_V1.0.uvproj	2022/6/28 22:36	Microvision5 Project	26 KB
startup_stm32f103xb.lst	2022/5/7 6:37	VisualStudio.lst...	38 KB
startup_stm32f103xb.s	2022/5/7 6:36	VisualStudio.s.1...	13 KB

生成的相关文件

工程文件可直接打开

Hardware 文件介绍:

文件夹中包含需要用的外设所编写的程序文件，包括 C 文件以及 H 头文件。

inc	2022/5/2 21:53	文件夹	编写的H头文件
src	2022/5/2 21:53	文件夹	编写的C文件

名称	修改日期	类型	大小
ADC_Operation	2022/5/2 2:06	文件夹	
BUZZER	2022/5/2 21:53	文件夹	
DTH11	2022/5/2 3:15	文件夹	
ESP8266	2022/5/2 2:01	文件夹	
OLED	2022/5/2 2:00	文件夹	
UART_Printf	2022/4/20 8:23	文件夹	

相应的C文件在对应的文件夹中

名称	修改日期	类型	大小
ADC_Operation.h	2022/5/7 7:15	VisualStudio.h.1...	9 KB
BUZZER.h	2022/5/2 22:27	VisualStudio.h.1...	2 KB
DTH11.h	2022/5/3 18:54	VisualStudio.h.1...	2 KB
ESP8266.h	2022/5/7 7:23	VisualStudio.h.1...	7 KB
OLED.h	2022/5/7 7:18	VisualStudio.h.1...	3 KB
UART_Printf.h	2022/5/2 1:48	VisualStudio.h.1...	2 KB

头文件在Inc文件夹中

Drivers 文件介绍:

文件夹中包含 HAL 库的驱动文件。

名称	修改日期	类型	大小
CMSIS	2022/5/7 6:36	文件夹	包含HAL库的驱动文件
STM32F1xx_HAL_Driver	2022/5/7 6:36	文件夹	

Core 文件介绍:

文件夹中包含开发板相关库的驱动文件。

名称	修改日期	类型	大小
Inc	2022/5/7 6:36	文件夹	开发板相关库的驱动文件
Src	2022/5/7 6:36	文件夹	

名称	修改日期	类型	大小
adc.c	2022/5/7 6:36	VisualStudio.c.1...	5 KB
dma.c	2022/5/7 6:36	VisualStudio.c.1...	2 KB
freertos.c	2022/5/7 6:51	VisualStudio.c.1...	11 KB
gpio.c	2022/5/7 6:36	VisualStudio.c.1...	4 KB
i2c.c	2022/5/7 6:36	VisualStudio.c.1...	3 KB
main.c	2022/5/7 6:36	VisualStudio.c.1...	12 KB
stm32f1xx_hal_msp.c	2022/5/7 6:36	VisualStudio.c.1...	3 KB
stm32f1xx_hal_timebase_tim.c	2022/5/7 6:36	VisualStudio.c.1...	4 KB
stm32f1xx_it.c	2022/5/7 6:36	VisualStudio.c.1...	8 KB
system_stm32f1xx.c	2022/4/13 15:18	VisualStudio.c.1...	15 KB
tim.c	2022/5/7 6:36	VisualStudio.c.1...	3 KB
usart.c	2022/5/7 6:36	VisualStudio.c.1...	7 KB

名称	修改日期	类型	大小
adc.h	2022/5/7 6:36	VisualStudio.h.1...	2 KB
ASCII.h	2022/5/1 14:35	VisualStudio.h.1...	13 KB
dma.h	2022/5/7 6:36	VisualStudio.h.1...	2 KB
FreeRTOSConfig.h	2022/5/7 6:36	VisualStudio.h.1...	6 KB
gpio.h	2022/5/7 6:36	VisualStudio.h.1...	2 KB
i2c.h	2022/5/7 6:36	VisualStudio.h.1...	2 KB
main.h	2022/5/7 6:36	VisualStudio.h.1...	4 KB
stm32f1xx_hal_conf.h	2022/5/7 6:36	VisualStudio.h.1...	16 KB
stm32f1xx_it.h	2022/5/7 6:36	VisualStudio.h.1...	3 KB
tim.h	2022/5/7 6:36	VisualStudio.h.1...	2 KB
usart.h	2022/5/7 6:36	VisualStudio.h.1...	2 KB

头文件

Common 文件介绍:

文件夹中包含我们对数据类型编写的一个规范文件。

名称	修改日期	类型	大小
numtype.h	2022/5/5 5:40	VisualStudio.h.1...	2 KB

编写数据类型的规范

函数介绍

Main.c 中的相关函数介绍:

SystemClock_Config:

- 根据指定的参数初始化 RCC 振荡器
- 在 RCC_OscInitTypeDef 结构中。
- 初始化 CPU、AHB、APB 总线时钟

BSP_Init:

- 初始化后的外设相关操作
- ADC 校准、LED 闪烁、CMD UART1_IT
- 如果 BSP init 成功，返回 OPERATION_SUCCESS

1. 指示灯闪烁 5 次，蜂鸣器工作:BSP 初始化开始

2. 完成 OLED 屏幕初始化
3. OLED 屏幕提示初始化开始
4. 通过 DMA 传输在常规组上启动 ADC 转换
5. ESP8266 初始化
6. LED2 闪烁间隔，蜂鸣器工作 5 次，表示操作开始。表示外设初始化成功

Error_Handler

- 用户可以添加自己的实现来报告 HAL 错误返回状态

OLED.c 中的相关函数:

- ✓ 该文件提供了配置的代码，在 Oled 上显示信息
 - ✓ 这个文件包括:OLED 初始化，重置，显示字符和数字功能
1. WriteCmd: 写命令函数
 2. WriteDat: 写数据函数
 3. OLED_Init: 写数据 functionOled 初始化功能
 4. OLED_SetPos: 设置起始点坐标
 5. OLED_Fill: 全屏 OLED 填充
 6. OLED_CLS: 清除屏幕
 7. OLED_ON: 清除屏幕进入 OLED 屏幕
 8. OLED_OFF: OLED 关闭
 9. OLED_ShowStr: OLED 闭式显示 ASCII 字符，6*8 和 8*16 可选
 10. OLED_ShowChar: OLED 显示字符
 11. OLED_ShowNum: 显示数字

ADC.c 中的相关函数:

- ✓ 该文件提供了 ADC1 配置 GetGas 的代码
1. ADC_GetGas_Init: 初始化 ADC 相关外设:ADC GPIO 口和 DMA 通道
 2. ADC_Get_Gas: 获得四个通道采集的电压值
 3. ADC_Get_NH3: 返回计算出来的气体浓度

4. ADC_Get_CO2: 返回计算出来的气体浓度
5. ADC_Get_CO: 返回计算出来的气体浓度
6. ADC_Get_H2S: 返回计算出来的气体浓度
7. ADC_Get_Voltage: 返回计算出的电压

DTH11.c:

- ✓ 该文件提供了 DTH11 配置的代码
1. DHT11_IO_OUT: 设置 DTH11 接口的工作模式为推挽输出模式
 2. DHT11_RST: DHT11 端口复位并发送启动信号
 3. Dht11_Check: 端口重置并发送启动信号

BUZZER.c 中的相关函数:

- ✓ 该文件提供了 BUZZER 配置的代码
- Buzzer_Work: 蜂鸣器功能, 通过延迟和 GPIO 口翻转

C 文件总结介绍:

Main. c	初始化时钟、相关外设以及编写主程序
OLED. c	配置的代码, 在 Oled 上显示信息
ADC. c	提供了 ADC1 配置 GetGas 的代码
DTH11. c	提供了 DTH11 配置的代码
BUZZER. c	提供了 BUZZER 配置的代码

文件调用关系图:

