

神舟号 STM32F407 开发板

使 用 说 明 书

郑州精畅科技有限责任公司制

简介

1、开发板检测

在收到快递之后，您第一步需要做的就是检测开发板是否完好。首先是包装，包装外观必须良好，然后我们打开包装盒，看到里面的神舟号STM32F407开发板和一个白色的12V1A电源包装盒。

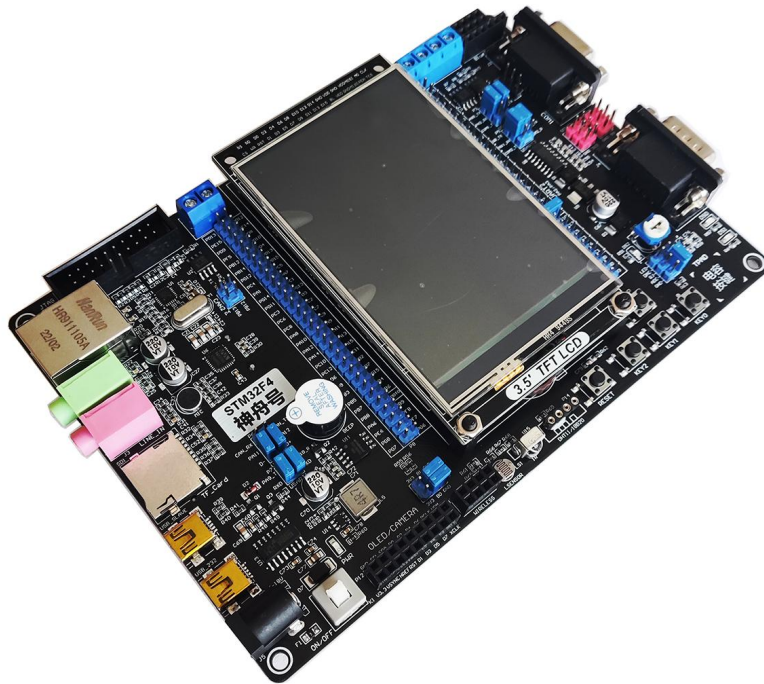
我们的开发板（包括液晶）用一个气泡袋包装起来了，另外有一个12V1A的适配器（用白色的盒子包起来了）。然后，取出所有配件，就可以看到整个套餐所含内容了，

我们神舟号STM32F407开发板标准套餐（假定配3.5寸电阻触摸屏）的所有内容，包括：神舟号STM32F407开发板底板1个、3.5寸触摸屏模块1个、12V1A电源适配器1个、红外遥控器1个、杜邦线5根、T口USB数据线1条。这些是神舟号STM32F407开发板的标配件。

其他套餐，大家根据自己拍下的内容，进行核对，我们一般会放有发货单，大家可以根据发货单自行核对。另外，如果您还购买了其他模块/芯片，请单独核对。

在确认接收到的开发板及配件外观没问题（主要看看LCD的触摸屏，是否有裂痕之后，请您开始检测开发板的硬件，是否存在问题（主要是LCD的问题，在运输过程中最容易受损）。

在出厂的时候，我们默认都是刷的神舟号综合测试实验，大家拿出开发板，先接上12V1A电源适配器给开发板供电（也可以是USB线供电，注意USB接口最好接USB_232这个端口），插上液晶（默认我们帮大家插好了），如果您有SD卡（或者购买了的话），可以插上SD卡。最后，按电源开关，给开发板上电，



此时开发板右下角的蓝色电源灯会亮，同时屏幕显示开发板检测信息，最后在检测正常之后，开发板的蜂鸣器会发出“滴”的一声，提示系统检测完毕，之后进入神舟号综合测试实验主界面。检测的时候如果您没有插SD卡的话，会提示SD CARD ERROR，这个是正常的，但是如果其他硬件检测错误，那么会提示对应的错误信息。

在进入到主界面之后，就可以通过点击屏幕的图标进行各项功能测试了。如果可以进入到主界面，并可以正常触摸，那么说明您的开发板就基本正常了，可以开始下面的学习了。

如果是开发板无法上电（电源指示灯不亮），请先检查12V1A电源适配器是否正常（适配器自带的指示灯是否亮了？），然后确认电源开关是否开启。

另外特别提醒：

1、如果您是用自己的液晶模块（或者以前购买的我们的液晶模块），发现触摸屏不准的时候，请先进行触摸屏校准。校准方法为：按下KEY0按下复位键（保持KEY0按下）松开复位键（KEY0还是保持按下），此时系统启动，等到Touch Check的时候，会弹出触摸屏校准界面，在校准界面，进行校准，成功后，您就可以正常使用了。

2、开发板右下角的TPAD（电容触摸），是一个电容触摸按键，可以用来作为退出键，有些界面（电子图书、数码相框、时钟、游戏、拨号和计算器等）必须用这个按键才能退出，请用手指轻轻触摸这里就可以退出了。

2、开始学习神舟号 STM32F407 开发板

在硬件检测完了之后,就可以开始学习神舟号 STM32F407 开发板了。先安装 MDK5.25, 安装方法见《MDK5.25 安装手册.pdf》。

再安装 CH340 驱动,用于串口下载代码。

如果买了 CMSIS-DAP 仿真器,则再安装 CMSIS-DAP 的驱动,用于仿真调试和代码下载。最后安装一个我们提供的串口调试助手 XCOM (或者使用 SSCOM3.3),用于观测一些串口数据。

另外有一个绿色软件 FLYMCU,用来串口下载代码,这也是很常用的一个软件,可以把该软件发一快捷方式到桌面。

以上软件的安装使用请参考:光盘开发环境搭建-软件安装这个视频。

在完成了这些之后,就可以跟着《STM32F4 开发指南》学习 STM32F407 了。

3、FAQ

1、USB 串口驱动安装后无法发现 USB 串口?

这个问题,先检查你的 USB 口有没有插对,我们开发板是需要把 USB 口插在 USB_232(即下方的那个 USB 口)才能实现 USB 串口。如果你已经是插对了口,那么有可能是 USB 线坏了,也有可能是板子有问题。此时你可以尝试先换过一根 USB 线试试,如果换线还是不行,请联系我们解决。

2、开发板上有 2 个 MiniUSB 接口,应该接哪个?

开发板的 2 个 MiniUSB 各有用途,他们不能通用。上方的 USB (USB_SLAVE) 是用来实现 STM32 与电脑的 USB 通信的,这个需要在 STM32 上面刷有 USB 协议的代码才可以用 (比如 USB 虚拟串口实验、USB 读卡器实验等就该插这个口)。而下方的 USB 口(USB_232)是用来实现 USB 转串口的,它并没有直接连接到 STM32F407,而是接 CH340 芯片,经过 CH340 将 USB 转为串口,再连接到 STM32F407。所以,这个 USB 口是用来实现串口通信,或者 ISP 下载代码用的。所以大家平常都应该插 USB_232 这个 MiniUSB 口。

3、触摸屏不准/反了,怎么办?

这问题一般是触摸屏没有校准,或者之前校准不对导致的,遇到此问题请先校准。校准办法可以通过下载:触摸屏实验,来校准。校准方法见 readme.txt。

最后,感谢您选择我们的产品,祝您身体健康、学习进步。

第一篇 硬件篇

实践出真知，要想学好 STM32F4，实验平台必不可少！本篇将详细介绍我们用来学习 STM32F4 的硬件平台：神舟号 STM32F4 开发板，通过该篇的介绍，你将了解到我们的学习平台神舟号 STM32F4 开发板的功能及特点。

为了让读者更好的使用神舟号 STM32F4 开发板，本篇还介绍了开发板的一些使用注意事项，请读者在使用开发板的时候一定要注意。

本篇将分为如下两章：

- 1，实验平台简介；
- 2，实验平台硬件资源详解；

第一章 实验平台简介

本章，主要向大家简要介绍我们的实验平台：神舟号 STM32F4 开发板。通过本章的学习，你将对我们后面使用的实验平台有个大概了解，为后面的学习做铺垫。

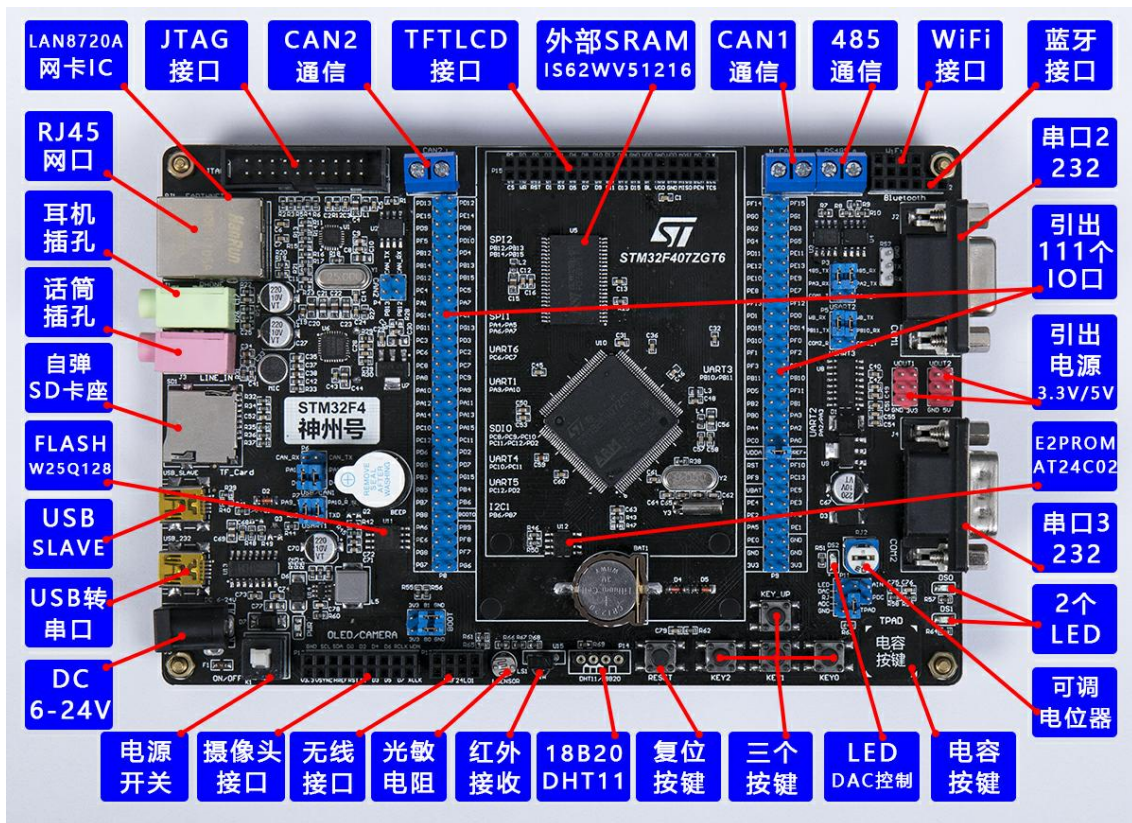
本章将分为如下两节：

- 1.1, 神舟号 STM32F4 开发板资源初探；
- 1.2, 神舟号 STM32F4 开发板资源说明；

1.1 神舟号 STM32F4 开发板资源初探

在神舟号 STM32F4 开发板之前，我们推出的两款 STM32F1 系列开发板：MiniSTM32 开发板和战神 STM32 开发板。而这款神舟号 STM32F4 开发板，则是推出的首款 CortexM4 开发板，下面我们开始介绍神舟号 STM32F4 开发板。

神舟号 STM32F4 开发板的资源图如图 1.1.1 所示：



从图 1.1.1 可以看出，神舟号 STM32F4 开发板，资源十分丰富，并把 STM32F407 的内部资源发挥到了极致，基本所有 STM32F407 的内部资源，都可以在此开发板上验证，同时扩充丰富的接口和功能模块，整个开发板显得十分大气。

开发板的外形尺寸为 111mm*163mm 大小，板子的设计充分考虑了人性化设计，并结合多年

的 STM32 开发板设计经验，同时听取了很多网友以及客户的建议，经过多次改进（面市之前，硬件改版超过 3 次，目前面市版本为 V2.0），最终确定了这样的设计。

神舟号 STM32F4 开发板板载资源如下：

- ◆ CPU： STM32F407ZGT6， LQFP144， FLASH： 1024K， SRAM： 192K；
- ◆ 外扩 SRAM： XM8A51216， 1M 字节
- ◆ 外扩 SPI FLASH： W25Q128， 16M 字节
- ◆ 1 个电源指示灯（蓝色）
- ◆ 2 个状态指示灯（DS0：红色， DS1：绿色）
- ◆ 1 个红外接收头，并配备一款小巧的红外遥控器
- ◆ 1 个 EEPROM 芯片， 24C02，容量 256 字节
- ◆ 1 个高性能音频编解码芯片， WM8978
- ◆ 1 个 2.4G 无线模块接口，支持 NRF24L01 无线模块
- ◆ 1 路 CAN 接口，采用 TJA1050 芯片
- ◆ 1 路 485 接口，采用 SP3485 芯片
- ◆ 2 路 RS232 串口（一公一母）接口，采用 SP3232 芯片
- ◆ 1 路单总线接口，支持 DS18B20/DHT11 等单总线传感器
- ◆ 1 个 WiFi 模块接口，支持 ESP-01 或者 ESP-01S 模块
- ◆ 1 个蓝牙模块接口，支持 HC-05 蓝牙模块
- ◆ 1 个光敏传感器
- ◆ 1 个标准的 2.4/2.8/3.5 寸 LCD 接口，支持电阻/电容触摸屏
- ◆ 1 个摄像头模块接口
- ◆ 1 个 OLED 模块接口
- ◆ 1 个 USB 串口，可用于程序下载和代码调试（USMART 调试）
- ◆ 1 个 USB SLAVE 接口，用于 USB 从机通信
- ◆ 1 个 USB HOST(OTG)接口，用于 USB 主机通信
- ◆ 1 个有源蜂鸣器
- ◆ 1 个 RS232/RS485 选择接口
- ◆ 1 个 RS232/模块选择接口
- ◆ 1 个 CAN/USB 选择接口
- ◆ 1 个串口选择接口

- ◆ 1 个 SD 卡接口 (TF 卡 支持最大 32G)
- ◆ 1 个百兆以太网接口 (RJ45)
- ◆ 1 个标准的 JTAG/SWD 调试下载口
- ◆ 1 个录音头 (MIC/咪头)
- ◆ 1 路立体声音频输出接口
- ◆ 1 路立体声录音输入接口
- ◆ 1 路扬声器输出接口, 可接 1W 左右小喇叭
- ◆ 1 组多功能端口 (DAC/ADC/PWM DAC/AUDIO IN/TPAD)
- ◆ 1 组 5V 电源供应/接入口
- ◆ 1 组 3.3V 电源供应/接入口
- ◆ 1 个参考电压设置接口
- ◆ 1 个直流电源输入接口 (输入电压范围: DC6~24V)
- ◆ 1 个启动模式选择配置接口
- ◆ 1 个 RTC 后备电池座, 并带电池
- ◆ 1 个复位按钮, 可用于复位 MCU 和 LCD
- ◆ 4 个功能按钮, 其中 KEY_UP(即 WK_UP)兼具唤醒功能
- ◆ 1 个电容触摸按键
- ◆ 1 个电源开关, 控制整个板的电源
- ◆ 独创的一键下载功能
- ◆ 除晶振占用的 IO 口外, 其余所有 IO 口全部引出

神舟号 STM32F4 开发板的特点包括:

- 1) **接口丰富。**板子提供十来种标准接口, 可以方便的进行各种外设的实验和开发。
- 2) **设计灵活。**板上很多资源都可以灵活配置, 以满足不同条件下的使用。我们引出了除晶振占用的 IO 口外的所有 IO 口, 可以极大的方便大家扩展及使用。另外板载一键下载功能, 可避免频繁设置 B0、B1 的麻烦, 仅通过 1 根 USB 线即可实现 STM32 的开发。
- 3) **资源充足。**主芯片采用自带 1M 字节 FLASH 的 STM32F407ZGT6, 并外扩 1M 字节 SRAM 和 16M 字节 FLASH, 满足大内存需求和大数据存储。板载高性能音频编解码芯片、百兆网卡、光敏传感器以及各种接口芯片, 满足各种应用需求。
- 4) **人性化设计。**各个接口都有丝印标注, 且用方框框出, 使用起来一目了然; 部分常用外设大丝印标出, 方便查找; 接口位置设计合理, 方便顺手。资源搭配合理, 物尽其用。

1.2 神舟号 STM32F4 开发板资源说明

资源说明部分，我们将分为两个部分说明：硬件资源说明和软件资源说明。

1.2.1 硬件资源说明

这里我们首先详细介绍神舟号 STM32F4 开发板的各个部分（图 1.1.1 中的标注部分）的硬件资源，我们顺序依次介绍。

1. NRF24L01 模块接口

这是开发板板载的 NRF24L01 模块接口（P13），只要插入模块，我们便可以实现无线通信，从而使得我们板子具备了无线功能，但是这里需要 2 个模块和 2 个开发板同时工作才可以。如果只有 1 个开发板或 1 个模块，是没法实现无线通信的。

2. W25Q128 128M FLASH

这是开发板外扩的 SPI FLASH 芯片（U11），容量为 128Mbit，也就是 16M 字节，可用于存储字库和其他用户数据，满足大容量数据存储要求。当然如果觉得 16M 字节还不够用，你可以把数据存放在外部 SD 卡。

3. SD 卡接口

这是开发板板载的一个标准 SD 卡接口（TF_CARD），采用小 TF 卡接口（SDIO 方式驱动，有了这个 SD 卡接口，就可以满足海量数据存储的需求。

4. 引出 IO 口（总共有两处）

这是开发板 IO 引出端口，总共有两组主 IO 引出口：P8 和 P9。分别采用 2*30 排针引出，总共引出 111 个 IO 口，

5. JTAG/SWD 接口

这是神舟号 STM32F4 开发板板载的 20 针标准 JTAG 调试口（JTAG），该 JTAG 口直接可以和 ULINK、JLINK 或者 STLINK 以及 DAP 等调试器（仿真器）连接，同时由于 STM32 支持 SWD 调试，这个 JTAG 口也可以用 SWD 模式来连接。

用标准的 JTAG 调试，需要占用 5 个 IO 口，有些时候，可能造成 IO 口不够用，而用 SWD 则只需要 2 个 IO 口，大大节约了 IO 数量，但他们达到的效果是一样的，所以我们强烈建议仿真器使用 SWD 模式！

6. CAN/USB 选择口

这是一个 CAN/USB 的选择接口（P6），因为 STM32 的 USB 和 CAN 是共用一组 IO（PA11 和 PA12），所以我们通过跳线帽来选择不同的功能，以实现 USB/CAN 的实验。

7. USB 串口/串口 1

这是 USB 串口同 STM32F407ZGT6 的串口 1 进行连接的接口（P7），标号 RXD 和 TXD 是 USB 转串口的 2 个数据口（对 CH340C 来说），而 PA9(TXD)和 PA10(RXD)则是 STM32 的串口 1 的两个数据口（复用功能下）。他们通过跳线帽对接，就可以和连接在一起了，从而实现 STM32 的程序下载以及串口通信。

设计成 USB 串口，是出于现在电脑上串口正在消失，尤其是笔记本，几乎清一色的没有串口。所以板载了 USB 串口可以方便大家下载代码和调试。而在板子上并没有直接连接在一起，则是出于使用方便的考虑。这样设计，你可以把神舟号 STM32F4 开发板当成一个 USB 转 TTL 串口，来和其他板子通信，而其他板子的串口，也可以方便地接到神舟号 STM32F4 开发板上。

8. STM32F407ZGT6

这是开发板的核心芯片（U10），型号为：STM32F407ZGT6。该芯片集成 FPU 和 DSP 指令，并具有 192KB SRAM、1024KB FLASH、12 个 16 位定时器、2 个 32 位定时器、2 个 DMA 控制器（共 16 个通道）、3 个 SPI、2 个全双工 I2S、3 个 IIC、6 个串口、2 个 USB（支持 HOST /SLAVE）、2 个 CAN、3 个 12 位 ADC、2 个 12 位 DAC、1 个 RTC（带日历功能）、1 个 SDIO 接口、1 个 FSMC 接口、1 个 10/100M 以太网 MAC 控制器、1 个摄像头接口、1 个硬件随机数生成器、以及 112 个通用 IO 口等。

9. USB HOST(OTG)

这是开发板板载的一个贴片式的 USB-A 座（USB_HOST，在开发板背面），由于 STM32F4 的 USB 是支持 HOST 的，所以我们可以通过这个 USB-A 座，连接 U 盘/USB 鼠标/USB 键盘等其他 USB 从设备，从而实现 USB 主机功能。不过特别注意，由于 USB HOST 和 USB SLAVE 是共用 PA11 和 PA12，所以两者不可以同时使用。

10. 后备电池接口

这是 STM32 后备区域的供电接口，可以用来给 STM32 的后备区域提供能量，在外部电源断电的时候，维持后备区域数据的存储，以及 RTC 的运行。

11. USB SLAVE

这是开发板板载的一个 MiniUSB 头（USB_SLAVE），用于 USB 从机（SLAVE）通信，一般用于 STM32 与电脑的 USB 通信。通过此 MiniUSB 头，开发板就可以和电脑进行 USB 通信了。注意：该接口不能和 USB HOST 同时使用。

开发板总共板载了 2 个 MiniUSB 头，一个（USB_232）用于 USB 转串口，连接 CH340C 芯片；另外一个（USB_SLAVE）用于 STM32 内带的 USB。同时开发板可以通过此 MiniUSB 头供电，板载两个 MiniUSB 头（不共用），主要是考虑了使用的方便性，以及可以给板子提供更大的电流（两个 USB

都接上)这两个因素。

12. USB 转串口

这是开发板板载的另外一个 MiniUSB 头 (USB_232)，用于 USB 连接 CH340C 芯片，从而实现 USB 转串口。同时，此 MiniUSB 接头也是开发板电源的主要提供口。

13. 小喇叭

这是开发板自带的一个 8Ω 1W 的小喇叭，安装在开发板的背面，可以用来播放音频。该喇叭由 WM8978 直接驱动，最大输出功率可达 0.9W。

14. OLED/摄像头模块接口

这是开发板板载的一个 OLED/摄像头模块接口 (P12)，如果是 OLED 模块，靠左插即可 (右边两个孔位悬空)。如果是摄像头模块，则刚好插满。通过这个接口，可以分别连接 2 个外部模块，从而实现相关实验。

15. 光敏传感器

这是开发板板载的一个光敏传感器 (LS1)，通过该传感器，开发板可以感知周围环境光线的变化，从而可以实现类似自动背光控制的应用。

16. 有源蜂鸣器

这是开发板的板载蜂鸣器 (BEEP)，可以实现简单的报警/闹铃。让开发板可以听得见。

17. 红外接收头

这是开发板的红外接收头 (U15)，可以实现红外遥控功能，通过这个接收头，可以接受市面常见的各种遥控器的红外信号，大家甚至可以自己实现万能红外解码。当然，如果应用得当，该接收头也可以用来传输数据。

神舟号 STM32F4 开发板给大家配备了一个小巧的红外遥控器，该遥控器外观如图 .2.1.1 所示：



18. 单总线接口

这是开发板的一个单总线接口 (P14)，该接口由 4 个镀金排孔组成，可以用来接 DS18B20 等单总线数字温度传感器。也可以用来接 DHT11 这样的单总线数字温湿度传感器。实现一个接口，多个功能。不用的时候，大家可以拆下上面的传感器，放到其他地方去用，使用上是十分方便灵活的。

19. 2 个 LED 灯

这是开发板板载的两个 LED 灯（DS0 和 DS1），DS0 是红色的，DS1 是绿色的，主要是方便大家识别。这里提醒大家不要停留在 51 跑马灯的思维，搞这么多灯，除了浪费 IO 口，实在是想不出其他什么优点。

我们一般的应用 2 个 LED 足够了，在调试代码的时候，使用 LED 来指示程序状态，是非常不错的一个辅助调试方法。神舟号 STM32F4 开发板几乎每个实例都使用了 LED 来指示程序的运行状态。

20. 复位按钮

这是开发板板载的复位按键（RESET），用于复位 STM32，还具有复位液晶的功能，因为液晶模块的复位引脚和 STM32 的复位引脚是连接在一起的，当按下该键的时候，STM32 和液晶一并被复位。

21. 启动选择端口

这是开发板板载的启动模式选择端口（BOOT），STM32 有 BOOT0（B0）和 BOOT1（B1）两个启动选择引脚，用于选择复位后 STM32 的启动模式，作为开发板，这两个是必须的。在开发板上，我们通过跳线帽选择 STM32 的启动模式。关于启动模式的说明，请看 2.1.8 小节。

22. 4 个按键

这是开发板板载的 4 个机械式输入按键（KEY0、KEY1、KEY2 和 KEY_UP），其中 KEY_UP 具有唤醒功能，该按键连接到 STM32 的 WAKE_UP（PA0）引脚，可用于待机模式下的唤醒，在不使用唤醒功能的时候，也可以做为普通按键输入使用。

其他 3 个是普通按键，可以用于人机交互的输入，这 3 个按键是直接连接在 STM32 的 IO 口上的。这里注意 KEY_UP 是高电平有效，而 KEY0、KEY1 和 KEY2 是低电平有效，大家在使用的时候留意一下。

23. 参考电压选择端口

这是 STM32 的参考电压选择端口（P9），我们默认是接开发板的 3.3V（VDDA）。如果大家想设置其他参考电压，只需要把你的参考电压源接到 Vref+和 GND 即可。

24. WiFi 模块接口

这是开发板板载的一个 ESP-01 模块接口（P1），可以接 ESP8266 模块，我们可以学习无线 WiFi 知识。

25. 触摸按钮

这是开发板板载的一个电容触摸输入按键（TPAD），利用电容充放电原理，实现触摸按键检测。

26. 电源指示灯

这是开发板板载的一颗蓝色的 LED 灯（PWR），用于指示电源状态。在电源开启的时候（通过板上的电源开关控制），该灯会亮，否则不亮。通过这个 LED，可以判断开发板的上电情况。

27. 多功能端口

这是 1 个由 6 个排针组成的一个接口（P11）。不过大家可别小看这 6 个排针，这可是本开发板设计的很巧妙的一个端口（由 P2 和 P12 组成），这组端口通过组合可以实现的功能有：ADC 采集、DAC 输出、PWM DAC 输出、外部音频输入、电容触摸按键、DAC 音频、PWM DAC 音频、DAC ADC 自测等，所有这些，你只需要 1 个跳线帽的设置，就可以逐一实现。

28. 扬声器接口

这是开发板预留的一个扬声器接口（SPK1），可以外接 1W（8Ω）左右的小喇叭，这样使用 WM8978 放音的时候，就可以直接推动喇叭输出音频了。不过，我们开发板自带了一个 8Ω 1W 的小喇叭，所以，这里即使不外接喇叭，也是可以听到声音的！

29. 耳机输出接口

这是开发板板载的音频输出接口（PHONE），该接口可以插 3.5mm 的耳机，当 WM8978 放音的时候，就可以通过在该接口插入耳机，欣赏音乐。

30. 录音输入接口

这是开发板板载的外部录音输入接口（LINE_IN），通过咪头我们只能实现单声道的录音，而通过这个 LINE_IN，我们可以实现立体声录音。

31. MIC（咪头）

这是开发板的板载录音输入接口（MIC），该咪头直接接到 WM8978 的输入上，可以用来实现录音功能。

32. 24C02 EEPROM

这是开发板板载的 EEPROM 芯片（U12），容量为 2Kb，也就是 256 字节。用于存储一些掉电不能丢失的重要数据，比如系统设置的一些参数/触摸屏校准数据等。有了这个就可以方便的实现掉电数据保存。

33. 蓝牙模块接口

这是开发板板载的一个通用蓝牙模块接口（P2），目前可以支持开发的 GPS 模块和蓝牙模块，直接插上对应的模块，就可以进行开发。后续我们将开发更多兼容该接口的其他模块，实现更强大的扩展性能。

34. 3.3V 电源输入/输出

这是开发板板载的一组 3.3V 电源输入输出红色排针（2*3）（VOUT1），用于给外部提供 3.3V 的电源，也可以用于从外部接 3.3V 的电源给板子供电。

大家在实验的时候可能经常会为没有 3.3V 电源而苦恼不已，有了神舟号 STM32F4 开发板，你

就可以很方便的拥有一个简单的 3.3V 电源（USB 供电的时候，最大电流不能超过 500mA，外部供电的时候，最大可达 1000mA）。

35. 5V 电源输入/输出

这是开发板板载的一组 5V 电源输入输出红色排针（2*3）（VOUT2），该排针用于给外部提供 5V 的电源，也可以用于从外部接 5V 的电源给板子供电。

同样大家在实验的时候可能经常会为没有 5V 电源而苦恼不已，充分考虑到了大家需求，有了这组 5V 排针，你就可以很方便的拥有一个简单的 5V 电源（USB 供电的时候，最大电流不能超过 500mA，外部供电的时候，最大可达 1000mA）。

36. 电源开关

这是开发板板载的电源开关（K1）。该开关用于控制整个开发板的供电，如果切断，则整个开发板都将断电，电源指示灯（PWR）会随着此开关的状态而亮灭。注意设计上为了保护电脑主板开关按下去是选择适配器 12V 的电源，弹起来选择是 USB 口供电的。

37. DC6~24V 电源输入

这是开发板板载的一个外部电源输入口（DC_IN），采用标准的直流电源插座。开发板板载了 DC-DC 芯片（RY8320），用于给开发板提供高效、稳定的 5V 电源。由于采用了 DC-DC 芯片，所以开发板的供电范围十分宽，大家可以很方便的找到合适的电源（只要输出范围在 DC6~24V 的基本都可以）来给开发板供电。在耗电比较大的情况下，比如用到 4.3 屏/7 寸屏/网口的时候，建议使用外部电源供电，可以提供足够的电流给开发板使用。

38. 以太网接口（RJ45）

这是开发板板载的网口（EARTHNET），可以用来连接网线，实现网络通信功能。该接口使用 STM32F4 内部的 MAC 控制器外加 PHY 芯片，实现 10/100M 网络的支持。

39. RS485 总线接口

这是开发板板载的 RS485 总线接口（RS485），通过 2 个端口和外部 485 设备连接。这里提醒大家，RS485 通信的时候，必须 A 接 A，B 接 B。否则可能通信不正常！

40. RS232/模块选择接口

这是开发板板载的一个 RS232（COM3）模块接口（U8）选择接口（P5），通过该选择接口，我们可以选择 STM32 的串口 3 连接在 COM3 还是连接在模块接口上面，以实现不同的应用需求。这样的设计还有一个好处，就是我们的开发板还可以充当 RS232 到 TTL 串口的转换（注意，这里的 TTL 高电平是 3.3V）。

41. RS232/485 选择接口

这是开发板板载的 RS232 (COM2) /485 选择接口 (P3)，因为 RS485 基本上就是一个半双工的串口，为了节约 IO，我们把 RS232 (COM1) 和 RS485 共用一个串口，通过 P3 来设置当前是使用 RS232 (COM1) 还是 RS485。这样的设计还有一个好处。就是我们的开发板既可以充当 RS232 到 TTL 串口的转换，又可以充当 RS485 到 TTL485 的转换。（注意，这里的 TTL 高电平是 3.3V）。

42. RS232 接口（公）

这是开发板板载的一个 RS232 接口 (COM2)，通过一个标准的 DB9 公头和外部的串口连接。通过这个接口，我们可以连接带有串口的电脑或者其他设备，实现串口通信。

43. LCD 接口

这是开发板板载的 LCD 模块接口，该接口兼容全系列 TFTLCD 模块，包括：2.4 寸、2.8 寸、3.5 寸、4.3 寸和 7 寸等 TFTLCD 模块，并且支持电阻/电容触摸功能。

44. 8M SRAM

这是开发板外扩的 SRAM 芯 (U5) 片，容量为 8M 位，也就是 1M 字节，这样，对大内存需求的应用（比如 GUI），就可以很好的实现了

45. RS232 接口（母）

这是开发板板载的另外一个 RS232 接口 (COM1)，通过一个标准的 DB9 母头和外部的串口连接。通过这个接口，我们可以连接带有串口的电脑或者其他设备，实现串口通信

46. CAN 接口

这是开发板板载的 CAN 总线接口 (CAN)，通过 2 个端口和外部 CAN 总线连接，即 CANH 和 CANL。这里提醒大家：CAN 通信的时候，必须 CANH 接 CANH，CANL 接 CANL，否则可能通信不正常！

1.2.2 软件资源说明

上面我们详细介绍了神舟号 STM32F4 开发板的硬件资源。接下来，我们将向大家简要介绍一下神舟号 STM32F4 开发板的软件资源。

神舟号 STM32F4 开发板提供的标准例程多达 59 个，一般的 STM32 开发板仅提供库函数代码，而我们则提供寄存器和库函数两个版本的代码（本手册以库函数版本作为介绍）。我们提供的这些例程，基本都是原创，拥有非常详细的注释，代码风格统一、循序渐进，非常适合初学者入门。而其他开发板的例程，大都是来自 ST 库函数的直接修改，注释也比较少，对初学者来说不那么容易入门。

神舟号 STM32F4 开发板的例程列表如表 1.2.2.1 所示：

编号	实验名字	编号	实验名字
1	跑马灯实验	31	DHT11 数字温湿度传感器实验
2	蜂鸣器实验	32	MPU6050 六轴传感器实验
3	按键输入实验	33	无线通信实验
4	串口通信实验	34	FLASH 模拟 EEPROM 实验
5	外部中断实验	35	摄像头实验
6	独立看门狗实验	36	外部 SRAM 实验
7	窗口看门狗实验	37	内存管理实验
8	定时器中断实验	38	SD 卡实验
9	PWM 输出实验	39	FATFS 实验
10	输入捕获实验	40	汉字显示实验
11	电容触摸按键实验	41	图片显示实验
12	OLED 实验	42	照相机实验
13	TFTLCD 实验	43	音乐播放器实验
14	USART 调试实验	44	录音机实验
15	RTC 实验	45	视频播放器实验
16	硬件随机数实验	46	FPU 测试 (Julia 分形) 实验
17	待机唤醒实验	47	DSP 测试实验
18	ADC 实验	48	手写识别实验
19	内部温度传感器实验	49	T9 拼音输入法实验
20	光敏传感器实验	50	串口 IAP 实验
21	DAC 实验	51	USB 读卡器 (Slave) 实验
22	PWM DAC 实验	52	USB 声卡 (Slave) 实验
23	DMA 实验	53	USB U 盘 (Host) 实验
24	IIC 实验	54	USB 鼠标键盘 (Host) 实验
25	SPI 实验	55	网络通信实验
26	485 实验	56	UCOSII 实验 1-任务调度
27	CAN 实验	57	UCOSII 实验 2-信号量和邮箱
28	触摸屏实验	58	UCOSII 实验 3-消息队列、信号量集和软件定时器
29	红外遥控实验	59	综合测试实验
30	DS18B20 数字温度传感器实验		

从上表可以看出，神舟号 STM32F4 开发板的例程基本上涵盖了 STM32F407ZGT6 的所有内部资源，并且外扩展了很多有价值的例程，比如：FLASH 模拟 EEPROM 实验、USART 调试实验、ucosii 实验、内存管理实验、IAP 实验、拼音输入法实验、手写识别实验、综合实验等。

而且从上表可以看出，例程安排是循序渐进的，首先从最基础的跑马灯开始，然后一步步深入，从简单到复杂，有利于大家的学习和掌握。所以，神舟号 STM32F4 开发板是非常适合初学

者的。当然，对于想深入了解 STM32 内部资源的朋友，神舟号 STM32F4 开发板也绝对是一个不错的选择。

1.2.3 神舟号 IO 引脚分配

为了让大家更快更好的使用我们的神舟号 STM32F407 开发板，这里特地将神舟号开发板主芯片：STM32F407ZGT6 的 IO 资源分配做了一个总表，以便大家查阅。神舟号 IO 引脚分配总表如表 1.2.3.1 所示：

神舟号 IO 资源分配表					
引脚	GPIO	连接资源		独立	连接关系说明
34	PA0	WK_UP		Y	1, 按键 KEY_UP 2, 可以做待机唤醒脚(WKUP)
35	PA1	RMII_REF_CLK		N	接 LAN8720 的 REFCLKO 脚
36	PA2	USART2_TX /RS485_RX	ETH_MDIO	N	1, RS232 串口 2(COM2)RX 脚(P9 设置) 2, RS485 RX 脚(P9 设置) 3, LAN8720 的 MDIO 脚
37	PA3	USART2_RX /RS485_TX	PWM_DAC	N	1, RS232 串口 2(COM2)TX 脚(P9 设置) 2, RS485 TX 脚(P9 设置) 3, PWM_DAC 输出脚
40	PA4	STM_DAC	DCMI_HREF	Y	1, DAC_OUT1 输出脚 2, OLED/CAMERA 接口的 HREF 引脚
41	PA5	STM_ADC		Y	ADC 输入引脚，同时做 TPAD 检测脚
42	PA6	DCMI_PCLK		Y	OLED/CAMERA 接口的 PCLK 脚
43	PA7	RMII_CRS_DV		N	接 LAN8720 的 CRS_DV 脚
100	PA8	DCMI_XCLK	REMOTE_IN	N	1, OLED/CAMERA 接口的 XCLK 脚 2, 接 HS0038 红外接收头
101	PA9	USART1_TX		Y	串口 1 TX 脚，默认连接 CH340 的 RX(P6 设置)
102	PA10	USART1_RX		Y	串口 1 RX 脚，默认连接 CH340 的 TX(P6 设置)
103	PA11	USB_D-	CAN_RX	Y	1, USB D-引脚(P11 设置) 2, CAN_RX 引脚(P11 设置)
104	PA12	USB_D+	CAN_TX	Y	1, USB D+引脚(P11 设置) 2, CAN_TX 引脚(P11 设置)
105	PA13	JTMS	SWDIO	N	JTAG/SWD 仿真接口, 没接任何外设
109	PA14	JTCK	SWDCLK	N	JTAG/SWD 仿真接口, 没接任何外设
110	PA15	JTDI	USB_PWR	N	1, JTAG 仿真口(JTDI) 2, USB_HOST 接口供电控制脚
46	PB0	T_SCK		Y	TFTLCD 接口触摸屏 SCK 信号
47	PB1	T_PEN		Y	TFTLCD 接口触摸屏 PEN 信号
48	PB2	BOOT1	T_MISO	N	1, BOOT1, 启动选择配置引脚(仅上电时用) 2, TFTLCD 接口触摸屏 MISO 信号

133	PB3	JTDO	SPI1_SCK	N	1, JTAG 仿真口 (JTDO) 2, W25Q128 和 WIRELESS 接口的 SCK 信号
134	PB4	JTRST	SPI1_MISO	N	1, JTAG 仿真口 (JTRST) 2, W25Q128 和 WIRELESS 接口的 MISO 信号
135	PB5		SPI1_MOSI	N	W25Q128 和 WIRELESS 接口的 MOSI 信号
136	PB6	DCMI_D5		Y	OLED/CAMERA 接口的 D5 脚
137	PB7	DCMI_VSYNC		Y	OLED/CAMERA 接口的 VSYNC 脚
139	PB8	IIC_SCL		N	接 24C02&MPU6050&WM8978 的 SCL
140	PB9	IIC_SDA		N	接 24C02&MPU6050&WM8978 的 SDA
69	PB10	USART3_TX		Y	1, RS232 串口 3 (COM3) RX 脚 (P10 设置) 2, MODULE 接口的 RXD 脚 (P10 设置)
70	PB11	USART3_RX		Y	1, RS232 串口 3 (COM3) TX 脚 (P10 设置) 2, MODULE 接口的 TXD 脚 (P10 设置)
73	PB12	I2S_LRCK		N	WM8978 的 LRCK 信号
74	PB13	I2S_SCLK		N	WM8978 的 SCLK 信号
75	PB14	F_CS		N	W25Q128 的片选信号
76	PB15	LCD_BL		Y	TFTLCD 接口背光控制脚
26	PC0	GBC_LED	3D_INT	N	1, MODULE 接口的 LED 引脚 2, MPU6050 模块的中断脚
27	PC1	ETH_MDC		N	接 LAN8720 的 MDC 脚
28	PC2	I2S_SDOUT		N	WM8978 的 SDOUT 信号
29	PC3	I2S_SDIN		N	WM8978 的 SDIN 信号
44	PC4	RMII_RXD0		N	接 LAN8720 的 RXD0 脚
45	PC5	RMII_RXD1		N	接 LAN8720 的 RXD1 脚
96	PC6	I2S_MCLK	DCMI_D0	N	1, WM8978 的 SDIN 信号 2, OLED/CAMERA 接口的 D0 脚
97	PC7		DCMI_D1	Y	OLED/CAMERA 接口的 D1 脚
98	PC8	SDIO_D0	DCMI_D2	N	1, SD 卡接口的 D0 2, OLED/CAMERA 接口的 D2 脚
99	PC9	SDIO_D1	DCMI_D3	N	1, SD 卡接口的 D1 2, OLED/CAMERA 接口的 D3 脚
111	PC10	SDIO_D2		N	SD 卡接口的 D2
112	PC11	SDIO_D3		N	SD 卡接口的 D3
113	PC12	SDIO_SCK	DCMI_D4	Y	1, SD 卡接口的 SCK 2, OLED/CAMERA 接口的 D4 脚
7	PC13	T_CS		Y	TFTLCD 接口触摸屏 CS 信号
8	PC14		RTC 晶振	N	接 32.768K 晶振, 不可用做 IO
9	PC15		RTC 晶振	N	接 32.768K 晶振, 不可用做 IO
114	PD0	FSMC_D2		N	FSMC 总线数据线 D2 (LCD/SRAM 共用)
115	PD1	FSMC_D3		N	FSMC 总线数据线 D3 (LCD/SRAM 共用)
116	PD2	SDIO_CMD		N	SD 卡接口的 CMD
117	PD3	ETH_RESET		N	接 LAN8720 的复位脚
118	PD4	FSMC_NOE		N	FSMC 总线 NOE (RD) (LCD/SRAM 共用)
119	PD5	FSMC_NWE		N	FSMC 总线 NWE (WR) (LCD/SRAM 共用)

122	PD6	DCMI_SCL		Y	OLED/CAMERA 接口的 SCL 脚
123	PD7	DCMI_SDA		Y	OLED/CAMERA 接口的 SDA 脚
77	PD8	FSMC_D13		N	FSMC 总线数据线 D13 (LCD/SRAM 共用)
78	PD9	FSMC_D14		N	FSMC 总线数据线 D14 (LCD/SRAM 共用)
79	PD10	FSMC_D15		N	FSMC 总线数据线 D15 (LCD/SRAM 共用)
80	PD11	FSMC_A16		N	FSMC 总线地址线 A17 (SRAM 专用)
81	PD12	FSMC_A17		N	FSMC 总线地址线 A18 (SRAM 专用)
82	PD13	FSMC_A18		N	FSMC 总线地址线 A19 (SRAM 专用)
85	PD14	FSMC_D0		N	FSMC 总线数据线 D0 (LCD/SRAM 共用)
86	PD15	FSMC_D1		N	FSMC 总线数据线 D1 (LCD/SRAM 共用)
141	PE0	FSMC_NBL0		N	FSMC 总线 NBL0 (SRAM 专用)
142	PE1	FSMC_NBL1		N	FSMC 总线 NBL1 (SRAM 专用)
1	PE2	KEY2		Y	接按键 KEY2
2	PE3	KEY1		Y	接按键 KEY1
3	PE4	KEY0		Y	接按键 KEY0
4	PE5	DCMI_D6		Y	OLED/CAMERA 接口的 D6 脚
5	PE6	DCMI_D7		Y	OLED/CAMERA 接口的 D7 脚
58	PE7	FSMC_D4		N	FSMC 总线数据线 D4 (LCD/SRAM 共用)
59	PE8	FSMC_D5		N	FSMC 总线数据线 D5 (LCD/SRAM 共用)
60	PE9	FSMC_D6		N	FSMC 总线数据线 D6 (LCD/SRAM 共用)
63	PE10	FSMC_D7		N	FSMC 总线数据线 D7 (LCD/SRAM 共用)
64	PE11	FSMC_D8		N	FSMC 总线数据线 D8 (LCD/SRAM 共用)
65	PE12	FSMC_D9		N	FSMC 总线数据线 D9 (LCD/SRAM 共用)
66	PE13	FSMC_D10		N	FSMC 总线数据线 D10 (LCD/SRAM 共用)
67	PE14	FSMC_D11		N	FSMC 总线数据线 D11 (LCD/SRAM 共用)
68	PE15	FSMC_D12		N	FSMC 总线数据线 D12 (LCD/SRAM 共用)
10	PF0	FSMC_A0		N	FSMC 总线地址线 A0 (SRAM 专用)
11	PF1	FSMC_A1		N	FSMC 总线地址线 A1 (SRAM 专用)
12	PF2	FSMC_A2		N	FSMC 总线地址线 A2 (SRAM 专用)
13	PF3	FSMC_A3		N	FSMC 总线地址线 A3 (SRAM 专用)
14	PF4	FSMC_A4		N	FSMC 总线地址线 A4 (SRAM 专用)
15	PF5	FSMC_A5		N	FSMC 总线地址线 A5 (SRAM 专用)
18	PF6	GBC_KEY		Y	接 MODULE 接口的 KEY 脚
19	PF7	LIGHT_SENSOR		N	接光敏传感器 (LS1)
20	PF8	BEEP		N	接蜂鸣器 (BEEP)
21	PF9	LED0		N	接 DS0 LED 灯 (红色)
22	PF10	LED1		N	接 DS1 LED 灯 (绿色)
49	PF11	T_MOSI		Y	TFTLCD 接口触摸屏 MOSI 信号
50	PF12	FSMC_A6		N	FSMC 总线地址线 A10 (SRAM/LCD 共用)
53	PF13	FSMC_A7		N	FSMC 总线地址线 A7 (SRAM 专用)
54	PF14	FSMC_A8		N	FSMC 总线地址线 A8 (SRAM 专用)
55	PF15	FSMC_A9		N	FSMC 总线地址线 A9 (SRAM 专用)
56	PG0	FSMC_A10		N	FSMC 总线地址线 A10 (SRAM 专用)
57	PG1	FSMC_A11		N	FSMC 总线地址线 A11 (SRAM 专用)

87	PG2	FSMC_A12		N	FSMC 总线地址线 A12 (SRAM 专用)
88	PG3	FSMC_A13		N	FSMC 总线地址线 A13 (SRAM 专用)
89	PG4	FSMC_A14		N	FSMC 总线地址线 A14 (SRAM 专用)
90	PG5	FSMC_A15		N	FSMC 总线地址线 A15 (SRAM 专用)
91	PG6	NRF_CE		Y	WIRELESS 接口的 CE 信号
92	PG7	NRF_CS		Y	WIRELESS 接口的 CS 信号
93	PG8	NRF_IRQ	RS485_RE	N	1, WIRELESS 接口 IRQ 信号 2, RS485 RE 引脚
124	PG9	DCMI_PWDN	1WIRE_DQ	N	1, OLED/CAMERA 接口的 PWDN 脚 2, 接单总线接口 (U12), 即 DHT11/DS18B20
125	PG10	FSMC_NE3		N	FSMC 总线的片选信号 3, 为外部 SRAM 片选信号
126	PG11	RMII_TX_EN		N	接 LAN8720 的 TXEN 脚
127	PG12	FSMC_NE4		Y	FSMC 总线的片选信号 4, 为 LCD 片选信号
128	PG13	RMII_TXD0		N	接 LAN8720 的 TXD0 脚
129	PG14	RMII_TXD1		N	接 LAN8720 的 TXD1 脚
132	PG15	DCMI_RESET		Y	OLED/CAMERA 接口的 RESET 脚

表 1.2.3.1 中, 引脚栏即 STM32F407ZGT6 的引脚编号; GPIO 栏则表示 GPIO; 连接资源栏表示了对应 GPIO 所连接到的网络; 独立栏, 表示该 IO 是否可以完全独立 (不接其他任何外设和上下拉电阻) 使用, 通过一定的方法, 可以达到完全独立使用该 IO, Y 表示可做独立 IO, N 表示不可做独立 IO; 连接关系栏, 则对每个 IO 的连接做了简单的介绍。

该表在: 光盘→2, 神舟号 STM32F4 开发板原理图 文件夹下有提供 Excel 格式, 并注有详细说明和使用建议, 大家可以打开该表格的 Excel 版本, 详细查看。

第二章 实验平台硬件资源详解

本章，我们将节向大家详细介绍神舟号 STM32F4 开发板各部分的硬件原理图，让大家对该开发板的各部分硬件原理有个深入理解，并向大家介绍开发板的使用注意事项，为后面的学习做好准备。

本章将分为如下两节：

1.1，开发板原理图详解；

1.2，开发板使用注意事项；

2.1 开发板原理图详解

2.1.1 MCU

神舟号 STM32F4 开发板选择的是 STM32F407ZGT6 作为 MCU，该芯片是 STM32F407 里面配置非常强大的了，它拥有的资源包括：集成 FPU 和 DSP 指令，并具有 192KBSRAM、1024KBFLASH、12 个 16 位定时器、2 个 32 位定时器、2 个 DMA 控制器（共 16 个通道）、3 个 SPI、2 个全双工 I2S、3 个 IIC、6 个串口、2 个 USB（支持 HOST/SLAVE）、2 个 CAN、3 个 12 位 ADC、2 个 12 位 DAC、1 个 RTC（带日历功能）、1 个 SDIO 接口、1 个 FSMC 接口、1 个 10/100M 以太网 MAC 控制器、1 个摄像头接口、1 个硬件随机数生成器、以及 112 个通用 IO 口等。该芯片的配置十分强悍，很多功能相对 STM32F1 来说进行了重大改进，比如 FSMC 的速度，F4 刷屏速度可达 3300W 像素/秒，而 F1 的速度则只有 500W 左右。

MCU 部分的原理图如图 2.1.1.1（因为原理图比较大，缩小下来可能有点看不清，请大家打开开发板光盘的原理图进行查看）所示：



这里主要讲解以下 3 个地方:

- 第 21 页 共 59 页

2.1.2 引出 IO 口

神舟号 STM32F4 开发板引出了 STM32F407ZGT6 的所有 IO 口，如图 2.1.2.1 所示：

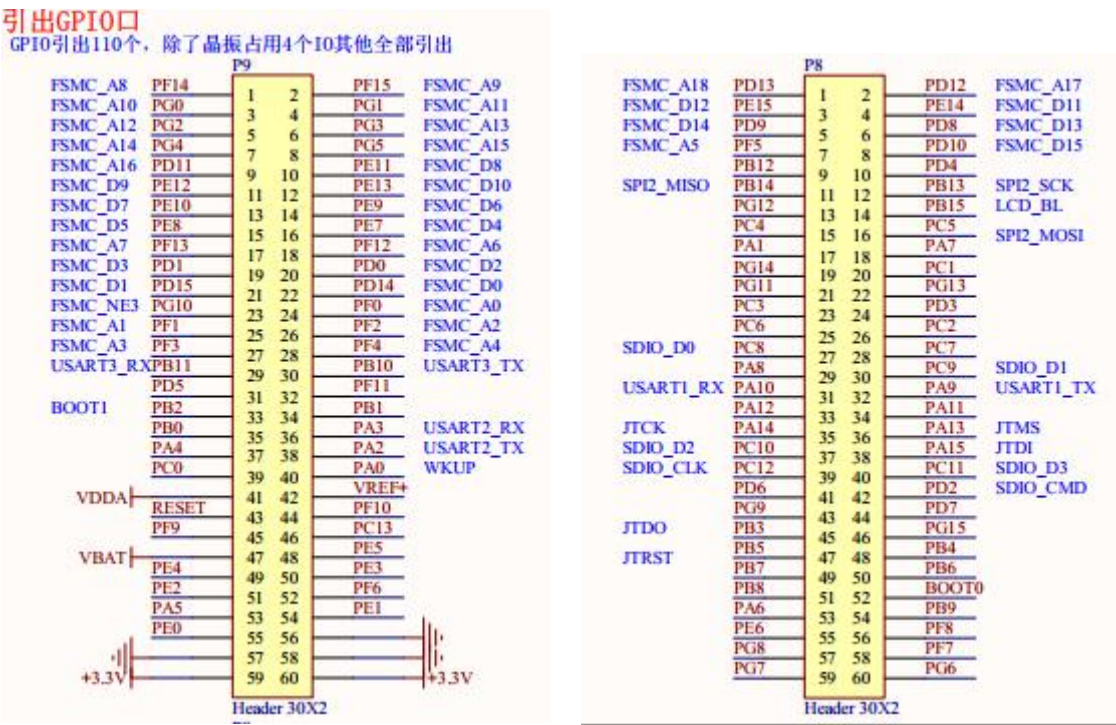


图 2.1.2.1 引出 IO 口

图中 P8 和 P9 为 MCU 主 IO 引出口，这 2 组排针共引出了 111 个 IO 口，STM32F407ZGT6 总共有 112 个 IO，除去 RTC 晶振占用的 2 个，还剩 110 个引出。

2.1.3 USB 串口/串口 1 选择接口

神舟号 STM32F4 开发板板载的 USB 串口和 STM32F407ZGT6 的串口是通过 P7 连接起来的，如图 2.1.3.1 所示：

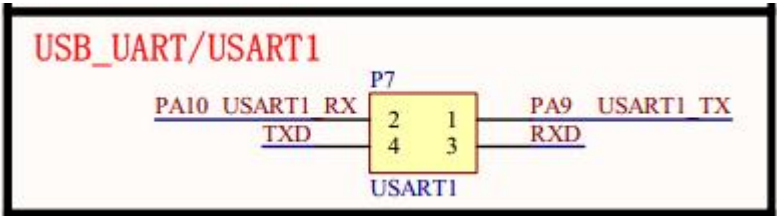


图 2.3.1.1USB 串口/串口 1 选择接口

图中 TXD/RXD 是相对 CH340C 来说的，也就是 USB 串口的发送和接受脚。而 USART1_RX 和 USART1_TX 则是相对于 STM32F407ZGT6 来说的。这样，通过对接，就可以实现 USB 串口和 STM32F407ZGT6 的串口通信了。同时，P7 是 PA9 和 PA10 的引出口。这样设计的好处就是使用上非常灵活。比如需要用到外部 TTL 串口和 STM32 通信的时候，只需要拔了跳线帽，通过杜邦线连接外

部 TTL 串口，就可以实现和外部设备的串口通信了；又比如我有个板子需要和电脑通信，但是电脑没有串口，那么你就可以使用开发板的 RXD 和 TXD 来连接你的设备，把我们的开发板当成 USB 转串口用了。

2.1.4 JTAG/SWD

神舟号 STM32F4 开发板板载的标准 20 针 JTAG/SWD 接口电路如图 2.1.4.1 所示：

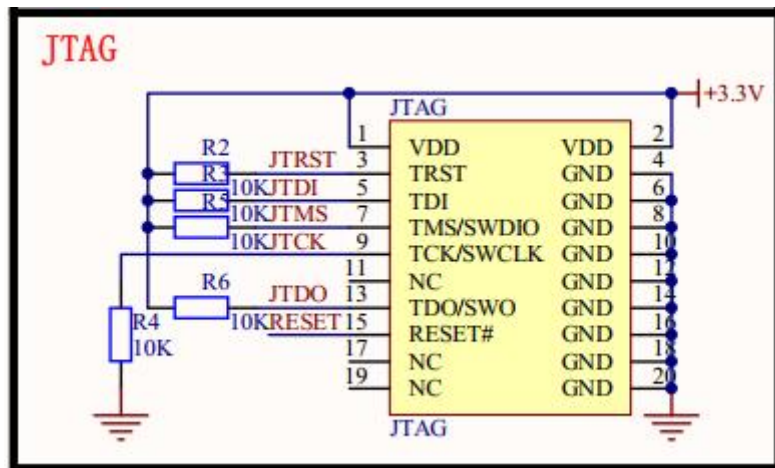


图 2.1.4.1 JTAG/SWD 接口

这里，我们采用的是标准的 JTAG 接法，但是 STM32 还有 SWD 接口，SWD 只需要最少 2 根线（SWCLK 和 SWDIO）就可以下载并调试代码了，这同我们使用串口下载代码差不多，而且速度非常快，能调试。所以建议大家在设计产品的时候，可以留出 SWD 来下载调试代码，而摒弃 JTAG。STM32 的 SWD 接口与 JTAG 是共用的，只要接上 JTAG，你就可以使用 SWD 模式了（其实并不需要 JTAG 这么多线），当然，你的调试器必须支持 SWD 模式，JLINKV7/V8、ULINK2 和 STLINK 以及 DAP 等都支持 SWD 调试。

特别提醒，JTAG 有几个信号线用来接其他外设了，但是 SWD 是完全没有接任何其他外设的，所以在使用的時候，**推荐大家一律使用 SWD 模式！！**

2.1.5 SRAM

神舟号 STM32F4 开发板外扩了 1M 字节的 SRAM 芯片，如图 2.1.5.1 所示，注意图中的地址线标号，是以 IS62WV51216BLL 为模版的，但是和 IS62WV51216BLL 的 datasheet 标号有出入，不过，因为地址的唯一性，这并不会影响我们使用 IS62WV51216BLL（**特别提醒：地址线可以乱，但是数据线必须一致！！**），因此，该原理图对这两个芯片都是可以正常使用的。

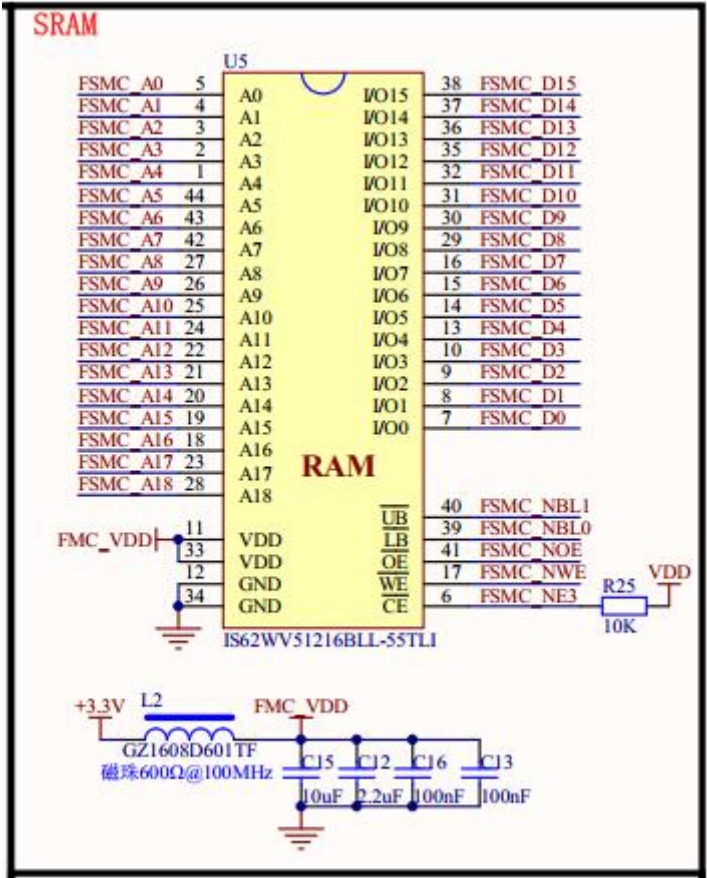


图 2.1.5.1 外扩 SRAM

图中 U5 为外扩的 SRAM 芯片，型号为：IS62WV51216BLL，容量为 1M 字节，该芯片挂在 STM32 的 FSMC 上。这样大大扩展了 STM32 的内存（芯片本身有 192K 字节），从而在需要大内存的场合，神舟号 STM32F4 开发板也可以胜任。

2.1.6 LCD 模块接口

神舟号 STM32F4 开发板板载的 LCD 模块接口电路如图 2.1.6.1 所示：

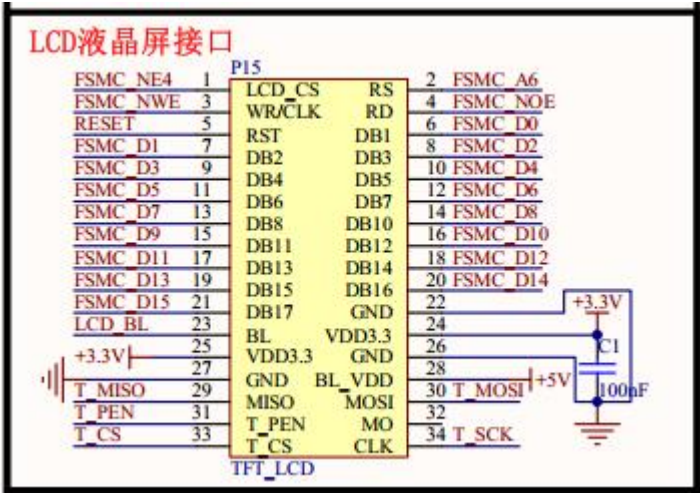


图 2.1.6.1LCD 模块接口

图中 TFT_LCD 是一个通用的液晶模块接口，支持全系列 TFTLCD 模块，包括：2.4 寸、2.8 寸、3.5 寸、4.3 寸和 7 寸等尺寸的 TFTLCD 模块。LCD 接口连接在 STM32F407ZGT6 的 FSMC 总线上面，可以显著提高 LCD 的刷屏速度。

图中的 T_MISO/T_MOSI/T_PEN/T_CS/T_CS 用来实现对液晶触摸屏的控制（支持电阻屏和电容屏）。LCD_BL 则控制 LCD 的背光。液晶复位信号 RESET 则是直接连接在开发板的复位按钮上，和 MCU 共用一个复位电路。

2.1.7 复位电路

神舟号 STM32F4 开发板的复位电路如图 2.1.7.1 所示：

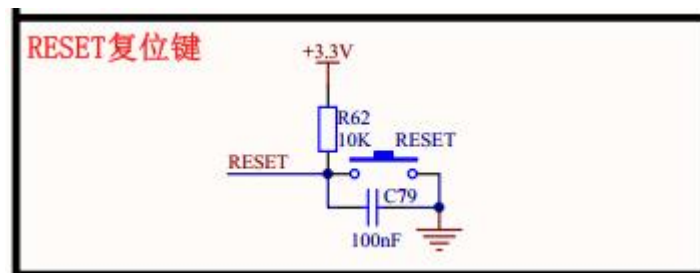


图 2.1.7.1 复位电路

因为 STM32 是低电平复位的，所以我们设计的电路也是低电平复位的，这里的 R62 和 C79 构成了上电复位电路。同时，开发板把 TFT_LCD 的复位引脚也接在 RESET 上，这样这个复位按钮不仅可以用来复位 MCU，还可以复位 LCD。

2.1.8 启动模式设置接口

神舟号 STM32F4 开发板的启动模式设置端口电路如图 2.1.8.1 所示：

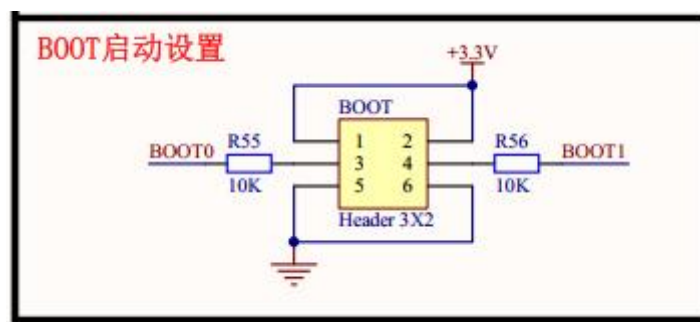


图 2.1.8.1 启动模式设置接口

上图的 BOOT0 和 BOOT1 用于设置 STM32 的启动方式，其对应启动模式如表 2.1.8.1 所示：

BOOT0	BOOT1	启动模式	说明
0	X	用户闪存存储器	用户闪存存储器，也就是FLASH启动
1	0	系统存储器	系统存储器启动，用于串口下载
1	1	SRAM启动	SRAM启动，用于在SRAM中调试代码

表 2.1.8.1BOOT0、BOOT1 启动模式表

按照表 2.1.8.1，一般情况下如果我们想用串口下载代码，则必须配置 BOOT0 为 1，BOOT1 为 0，而如果想让 STM32 一按复位键就开始跑代码，则需要配置 BOOT0 为 0，BOOT1 随便设置都可以。这里神舟号 STM32F4 开发板专门设计了一键下载电路，通过串口的 DTR 和 RTS 信号，来自动配置 BOOT0 和 RST 信号，因此不需要用户来手动切换他们的状态，直接串口下载软件自动控制，可以非常方便的下载代码。

2.1.9 RS232 串口

神舟号 STM32F4 开发板板载了一公一母两个 RS232 接口，电路原理图如图 2.1.9.1 所示：

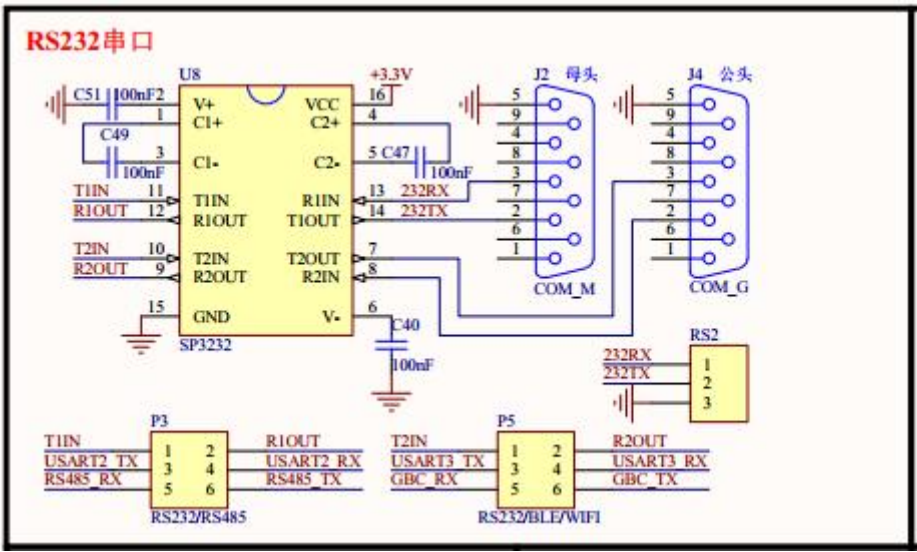


图 2.1.9.1RS232 串口

因为 RS232 电平不能直接连接到 STM32，所以需要有一个电平转换芯片。这里我们选择的是 SP3232（也可以用 MAX3232）来做电平转接，同时图中的 P3 用来实现 RS232(COM1)/RS485 的选择，P5 用来实现 RS232(COM2)模块接口的选择，以满足不同实验的需要。

图中 USART2_TX/USART2_RX 连接在 MCU 的串口 2 上(PA2/PA3)，所以这里的 RS232(COM1)/RS485 都是通过串口 2 来实现的。图中 RS485_TX 和 RS485_RX 信号接在 SP3485 的 DI 和 RO 信号上。

而图中的 USART3_TX/USART3_RX 则是连接在 MCU 的串口 3 上(PB10/PB11)，所以 RS232(COM2)/模块接口都是通过串口 3 来实现的。图中 GBC_RX 和 GBC_TX 连接在 WiFi 模块和蓝牙模块接口上面。

因为 P3/P5 的存在，其实还带来另外一个好处，就是我们可以把开发板变成一个 RS232 电平转

换器，或者 RS485 电平转换器，比如你买的核心板，可能没有板载 RS485/RS232 接口，通过连接神舟号 STM32F4 开发板的 P3/P5 端口，就可以让你的核心板拥有 RS232/RS485 的功能。

2.1.10 RS485 接口

神舟号 STM32F4 开发板板载的 RS485 接口电路如图 2.1.10.1 所示：

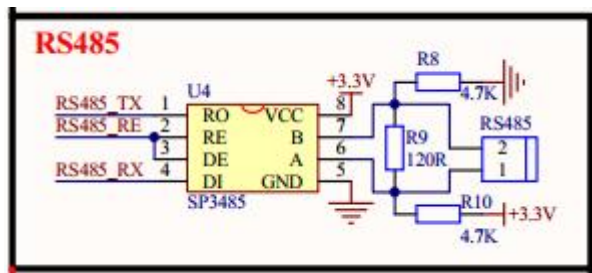


图 2.1.10.1 RS485 接口

RS485 电平也不能直接连接到 STM32，同样需要电平转换芯片。这里我们使用 SP3485 来做 485 电平转换，其中 R9 为终端匹配电阻，而 R8 和 R10 则是两个偏置电阻，以保证静默状态时，485 总线维持逻辑 1。

RS485_RX/RS485_TX 连接在 P3 上面，通过 P3 跳线来选择是否连接在 MCU 上面，RS485_RE 则是直接连接在 MCU 的 IO 口（PG8）上的，该信号用来控制 SP3485 的工作模式（高电平为发送模式，低电平为接收模式）。

另外，**特别注意**：RS485_RE 和 NRF_IRQ 共同接在 PG8 上面，在同时用到这两个外设的时候，需要注意下。

2.1.11 CAN/USB 接口

神舟号 STM32F4 开发板板载的 CAN 接口电路以及 STM32 USB 接口电路如图 2.1.11.1 所示：

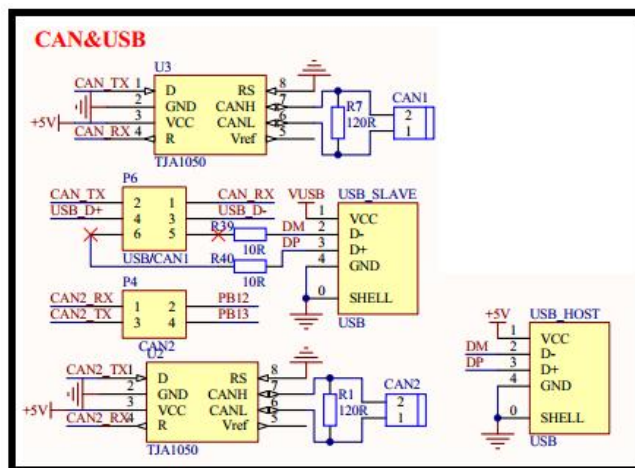


图 2.1.11.1 CAN/USB 接口

CAN 总线电平也不能直接连接到 STM32，同样需要电平转换芯片。这里我们使用 TJA1050 来做 CAN 电平转换，其中 R1（120Ω）为终端匹配电阻。

USB_D+/USB_D-连接在 MCU 的 USB 口（PA12/PA11）上，同时，因为 STM32 的 USB 和 CAN 共用这组信号，所以我们通过 P6 来选择使用 USB 还是 CAN。

图中共有 2 个 USB 口：USB_SLAVE 和 USB_HOST，前者是用来做 USB 从机通信的，后者则是用来做 USB 主机通信的。

USB_SLAVE 可以用来连接电脑，实现 USB 读卡器或声卡等 USB 从机实验。另外，该接口还具有供电功能，VUSB 为开发板的 USB 供电电压，通过这个 USB 口，就可以给整个开发板供电了。

USB HOST（在开发板背面）可以用来接如：U 盘、USB 鼠标、USB 键盘和 USB 手柄等设备，实现 USB 主机功能。该接口可以对从设备供电，且供电可控，通过 USB_PWR 控制，该信号连接在 MCU 的 PA15 引脚上与 JTDI 共用 PA15，所以用 JTAG 仿真的时候，USB_PWR 就不受控了，这也是我们推荐大家使用 SWD 模式而不用 JTAG 模式的另外一个原因。

2.1.12 EEPROM

神舟号 STM32F4 开发板板载的 EEPROM 电路如图 2.1.12.1 所示：

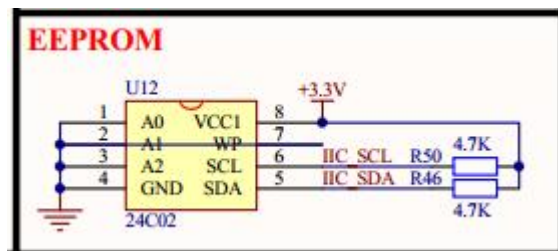


图 2.1.12.1 EEPROM

EEPROM 芯片我们使用的是 24C02，该芯片的容量为 2Kb，也就是 256 个字节，对于我们普通应用来说是足够的。当然，你也可以选择换大的芯片，因为我们的电路在原理上是兼容 24C02~24C512 全系列 EEPROM 芯片的。

这里我们把 A0~A2 均接地，对 24C02 来说也就是把地址位设置成了 0 了，写程序的时候要注意这点。IIC_SCL 接在 MCU 的 PB8 上，IIC_SDA 接在 MCU 的 PB9 上，这里我们虽然接到了 STM32 的硬件 IIC 上，但是我们并不提倡使用硬件 IIC，因为 STM32 的 IIC 是鸡肋！请谨慎使用。IIC_SCL/IIC_SDA 总线上总共挂了 2 个器件：24C02 和 WM8978，后续我们将向大家介绍另外两个器件。

2.1.13 光敏传感器

神舟号 STM32F4 开发板板载了一个光敏传感器，可以用来感应周围光线的变化，该部分电路如

图 2.1.13.1 所示：

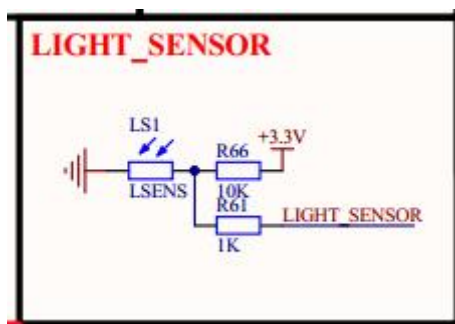


图 2.1.13.1 光敏传感器电路

图中的 LS1 就是光敏传感器，其实就是一个光敏二极管，周围环境越亮，电流越大，反之电流越小，即可等效为一个电阻，环境越亮阻值越小，反之越大，从而通过读取 LIGHT_SENSOR 的电压，即可知道周围环境光线强弱。LIGHT_SENSOR 连接在 MCU 的 ADC3_IN5（ADC3 通道 5）上面，即 PF7 引脚。

2.1.14 SPI FLASH

神舟号 STM32F4 开发板板载的 SPI FLASH 电路如图 2.1.14.1 所示：

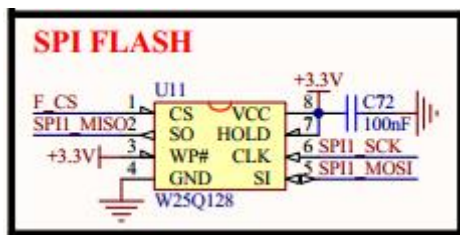


图 2.1.14.1 SPIFLASH 芯片

SPI FLASH 芯片型号为 W25Q128，该芯片的容量为 128Mb，也就是 16M 字节。该芯片和 NRF24L01 共用一个 SPI（SPI1），通过片选来选择使用某个器件，在使用其中一个器件的时候，请务必禁止另外一个器件的片选信号。

图中 F_CS 连接在 MCU 的 PB14 上，SPI1_SCK/SPI1_MOSI/SPI1_MISO 则分别连接在 MCU 的 PB3/PB5/PB4 上，其中 PB3/PB4 又是 JTAG 的 JTDO 和 JTRST 信号，所以在 JTAG 仿真的时候，SPI 就用不了了，但是用 SWD 仿真，则不存在任何问题，所以我们推荐大家使用 SWD 仿真！

2.1.15 WiFi 模块接口

神舟号 STM32F4 开发板板载的 ESP8266 模块接口电路如图 2.1.15.1 所示：

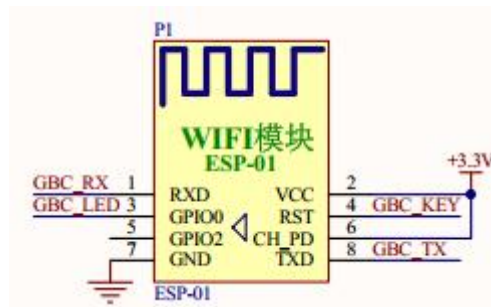


图 2.1.15.1 ESP-01 模块接口

支持 ESP8266 的 WiFi 模块（ESP-01 和 ESP-01S）。可以直接学习无线 WiFi 网络知识。

2.1.16 温湿度传感器接口

神舟号 STM32F4 开发板板载的温湿度传感器接口电路如图 2.1.16.1 所示：

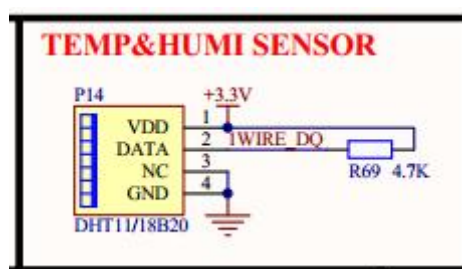


图 2.1.16.1 温湿度传感器接口

该接口支持 DS18B20/DS1820/DHT11 等单总线数字温湿度传感器。1WIRE_DQ 是传感器的数据线，该信号连接在 MCU 的 PG9 上，特别注意：该引脚同时还接到了摄像头模块的 DCMI_PWDN 信号上面，他们不能同时使用，但可以分时复用。

2.1.17 红外接收头

神舟号 STM32F4 开发板板载的红外接收头电路如图 2.1.17.1 所示：

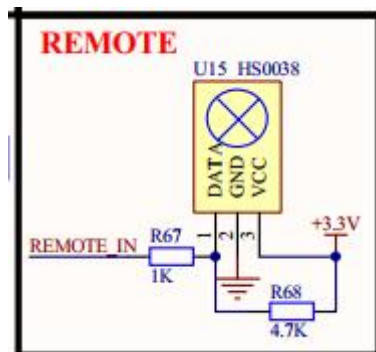


图 2.1.17.1 红外接收头

HS0038 是一个通用的红外接收头，几乎可以接收市面上所有红外遥控器的信号，有了它，就可

以用红外遥控器来控制开发板了。REMOTE_IN 为红外接收头的输出信号，该信号连接在 MCU 的 PA8 上。特别注意：PA8 同时连接了 DCMI_XCLK，如果要用到 DCMI_XCLK 的时候，HS0038 就不能同时使用了，但可以分时复用。

2.1.18 无线模块接口

神舟号 STM32F4 开发板板载的无线模块接口电路如图 2.1.18.1 所示：

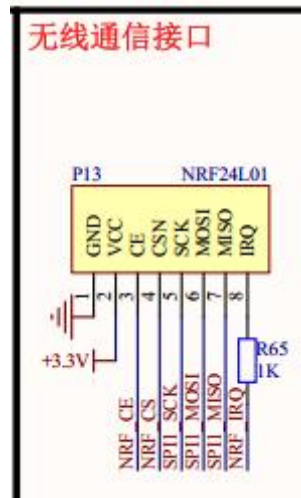


图 2.1.18.1 无线模块接口

该接口用来连接 NRF24L01 等 2.4G 无线模块，从而实现开发板与其他设备的无线数据传输（注意：NRF24L01 不能和蓝牙/WIFI 连接）。NRF24L01 无线模块的最大传输速度可以达到 2Mbps，传输距离最大可以到 30 米左右（空旷地，无干扰）。

NRF_CE/NRF_CS/NRF_IRQ 连接在 MCU 的 PG6/PG7/PG8 上，而另外 3 个 SPI 信号则和 SPIFLASH 共用，接 MCU 的 SPI1。这里需要注意的是 PG8 还接了 RS485 的 RE 信号，所以在使用 NRF24L01 中断引脚的时候，不能和 RS485 同时使用，不过，如果没用到 NRF24L01 的中断引脚，那么 RS485 和 NRF24L01 模块就可以同时使用了。

2.1.19 LED 灯

神舟号 STM32F4 开发板板载总共有 3 个 LED，其原理图如图 2.1.19.1 所示：

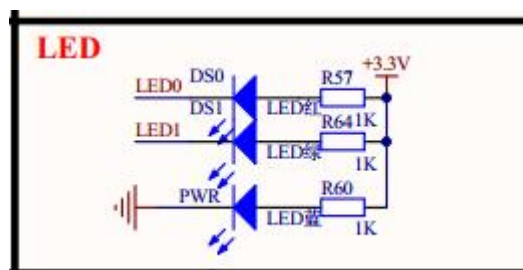


图 2.1.19.1 LED

其中 PWR 是系统电源指示灯，为蓝色。LED0(DS0)和 LED1(DS1)分别接在 PF9 和 PF10 上。为了方便大家判断，我们选择了 DS0 为红色的 LED，DS1 为绿色的 LED。

2.1.20 按键

神舟号 STM32F4 开发板板载总共有 4 个输入按键，其原理图如图 2.1.20.1 所示：

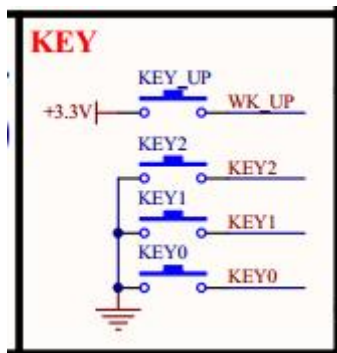


图 2.1.20.1 输入按键

KEY0、KEY1 和 KEY2 用作普通按键输入，分别连接在 PE4、PE3 和 PE2 上，这里并没有使用外部上拉电阻，但是 STM32 的 IO 作为输入的时候，可以设置上下拉电阻，所以我们使用 STM32 的内部上拉电阻来为按键提供上拉。

KEY_UP 按键连接到 PA0(STM32 的 WKUP 引脚)，它除了可以用作普通输入按键外，还可以用作 STM32 的唤醒输入。注意：这个按键是高电平触发的。

2.1.21 TPAD 电容触摸按键

神舟号 STM32F4 开发板板载了一个电容触摸按键，其原理图如图 2.1.21.1 所示：

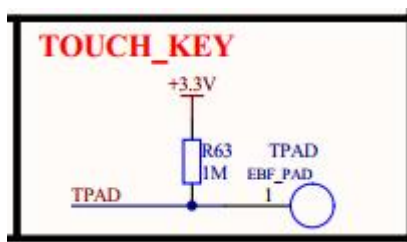


图 2.1.21.1 电容触摸按键

图中 1M 电阻是电容充电电阻，TPAD 并没有直接连接在 MCU 上，而是连接在多功能端口 (P12) 上面，通过跳线帽来选择是否连接到 STM32。多功能端口，我们将在 2.1.26 节介绍。电容触摸按键的原理我们将在后续的实战篇里面介绍。

2.1.22 OLED/摄像头模块接口

神舟号 STM32F4 开发板板载了一个 OLED/摄像头模块接口，其原理图如图 2.1.22.1 所示：

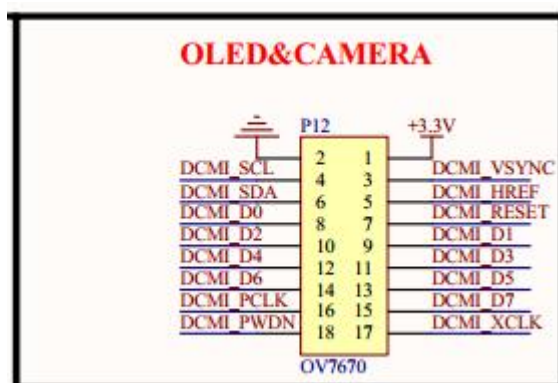


图 2.1.22.1 OLED/摄像头模块接口

图中 P12 是接口可以用来连接 OLED 模块或者摄像头模块。如果是 OLED 模块，则 DCMI_PWDN 和 DCMI_XCLK 不需要接（在板上靠左插即可），如果是摄像头模块，则需要用到全部引脚。

其中，DCMI_SCL/DCMI_SDA/DCMI_RESET/DCMI_PWDN/DCMI_XCLK 这 5 个信号是不属于 STM32F4 硬件摄像头接口的信号，通过普通 IO 控制即可，分别接在 MCU 的：PD6/PD7/PG15/PG9/PA8 上面。特别注意：DCMI_PWDN 和 1WIRE_DQ 信号共用 PG9 这个 IO，所以摄像头和 DS18B20/DHT11 不能同时使用，但是可以分时复用。另外，DCMI_XCLK 和 REMOTE_IN 共用，在用到 DCMI_XCLK 信号的时候，则红外接收和摄像头不可同时使用，不过同样是可以分时复用的。

其他信号全接在 STM32F4 的硬件摄像头接口上，DCMI_VSYNC/DCMI_HREF/DCMI_D0/DCMI_D1/DCMI_D2/DCMI_D3/DCMI_D4/DCMI_D5/DCMI_D6/DCMI_D7/DCMI_PCLK 分别连接在：PB7/PA4/PC6/PC7/PC8/PC9/PC11/PB6/PE5/PE6/PA6 上。特别注意：这些信号和 DAC1 输出以及 SD 卡，I2S 音频等有 IO 共用，所以在使用 OLED 模块或摄像头模块的时候，不能和 DAC1 的输出、SD 卡使用和 I2S 音频播放等三个功能同时使用，只能分时复用。

2.1.23 有源蜂鸣器

神舟号 STM32F4 开发板板载了一个有源 5V 蜂鸣器，其原理图如图 2.1.23.1 所示：

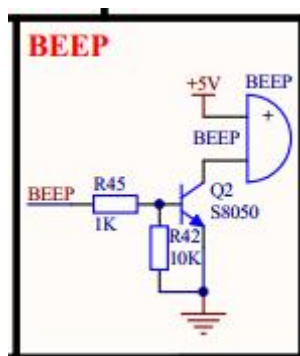


图 2.1.23.1 有源蜂鸣器

有源蜂鸣器是指自带了震荡电路的蜂鸣器，这种蜂鸣器一接上电就会自己震荡发声。而如果是无源蜂鸣器，则需要外加一定频率（2~5KHz）的驱动信号，才会发声。这里我们选择使用有源蜂鸣器，方便大家使用。

图中 Q2 是用来扩流，R42 则是一个下拉电阻，避免 MCU 复位的时候，蜂鸣器可能发声的现象。BEEP 信号直接连接在 MCU 的 PF8 上面，PF8 可以做 PWM 输出，所以大家如果想玩高级点（如：控制蜂鸣器“唱歌”），就可以使用 PWM 来控制蜂鸣器。

2.1.24 SD 卡接口

神舟号 STM32F4 开发板板载了一个 SD 卡（TF 卡）接口，其原理图如图 2.1.24.1 所示：

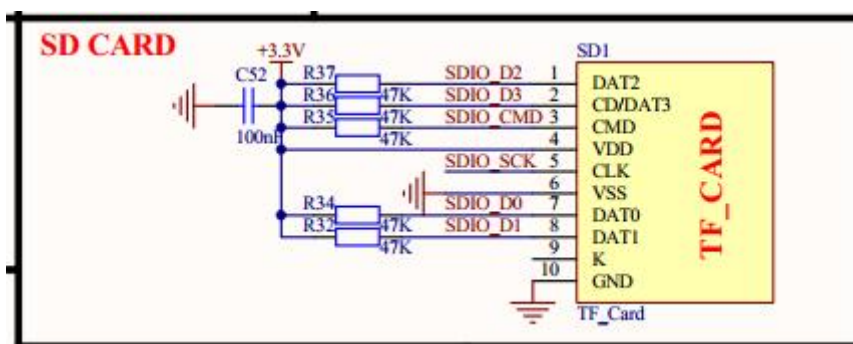


图 2.1.24.1 SD 卡接口

图中 SD_CARD 为 SD 卡接口，SD 卡采用 4 位 SDIO 方式驱动，理论上最大速度可以达到 24MB/S，非常适合需要高速存储的情况。图中：SDIO_D0/SDIO_D1/SDIO_D2/SDIO_D3/SDIO_SCK/SDIO_CMD 分别连接在 MCU 的 PC8/PC9/PC10/PC11/PC12/PD2 上面。特别注意：SDIO 和 OLED/摄像头的部分 IO 有共用，所以在使用 OLED 模块或摄像头模块的时候，只能和 SDIO 分时复用，不能同时使用。

2.1.25 蓝牙模块接口

神舟号 STM32F4 开发板板载了蓝牙模块接口，其原理图如图 2.1.25.1 所示：

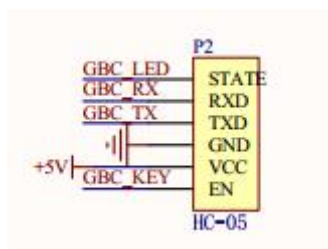


图 2.1.25.1 蓝牙模块接口

2.1.26 多功能端口

神舟号 STM32F4 开发板板载的多功能端口，是由 P11 构成的一个端口，其原理图如图 2.1.26.1 所示：

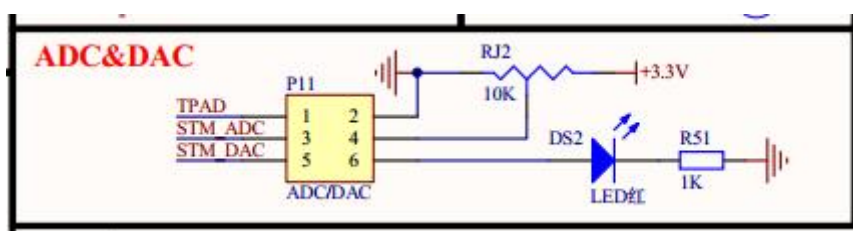


图 2.1.26.1 多功能端口

从上图，大家可能还看不出这个多功能端口的全部功能，别担心，下面我们会详细介绍。

首先介绍左侧的 P11，其中 TPAD 为电容触摸按键信号，连接在电容触摸按键上。STM_ADC 和 STM_DAC 则分别连接在 PA5 和 PA4 上，用于 ADC 采集或 DAC 输出。当需要电容触摸按键的时候，我们通过跳线帽短接 TPAD 和 STM_ADC，就可以实现电容触摸按键（利用定时器的输入捕获）。STM_DAC 信号则既可以用作 DAC 输出，也可以用作 ADC 输入，因为 STM32 的该管脚同时具有这两个复用功能。特别注意：STM_DAC 与摄像头的 DCMI_HREF 共用 PA4，所以他们不可以同时使用，但是可以分时复用。

我们再来看看 P11，PWM_DAC 连接在 MCU 的 PA3，是定时器 2/5 的通道 4 输出，后面跟一个二阶 RC 滤波电路，其截止频率为 33.8KHz。经过这个滤波电路，MCU 输出的方波就变为直流信号了。PWM_AUDIO 是一个音频输入通道，它连接到 WM8978 的 AUX 输入，可通过配置 WM8978，输出到耳机/扬声器。特别注意：PWM_DAC 和 USART2_RX 共用 PA3，所以 PWM_DAC 和串口 2 的接收，不可以同时使用，不过，可以分时复用。

2.1.27 以太网接口（RJ45）

神舟号 STM32F4 开发板板载了一个以太网接口(RJ45)，其原理图如图 2.1.27.1 所示：

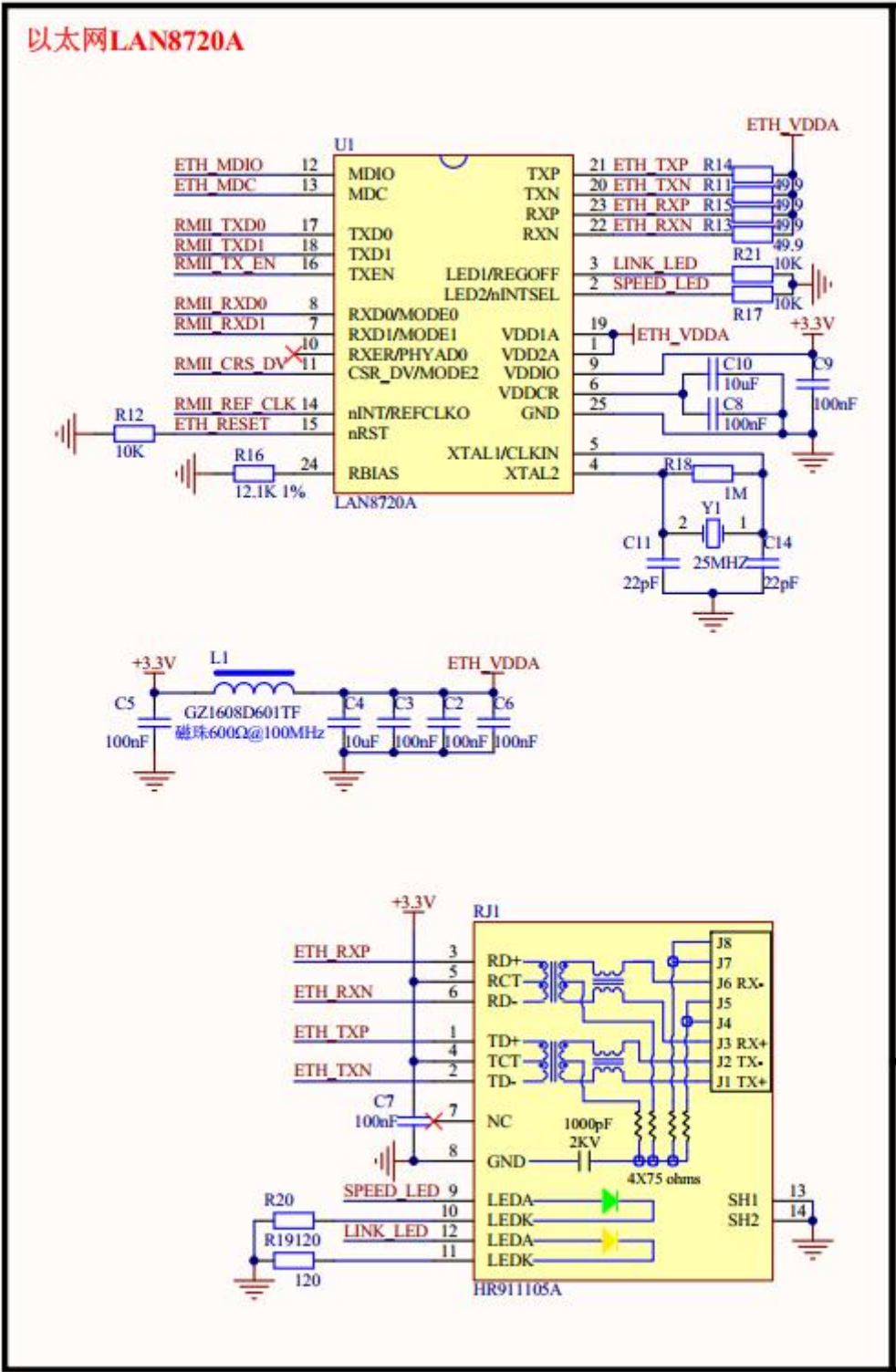


图 2.1.27.1 以太网接口电路

STM32F4 内部自带网络 MAC 控制器，所以只需要外加一个 PHY 芯片，即可实现网络通信功能。这里我们选择的是 LAN8720A 这颗芯片作为 STM32F4 的 PHY 芯片，该芯片采用 RMII 接口与 STM32F4 通信，占用 IO 较少，且支持 autmdix（即可自动识别交叉/直连网线）功能。板载一个自带网络变压器的 RJ45 头（HR91105A），一起组成一个 10M/100M 自适应网卡。

图中：ETH_MDIO/ETH_MDC/RMII_TXD0/RMII_TXD1/RMII_TX_EN/RMII_RXD0/RMII_RXD1/RMII_CRS_DV/RMII_REF_CLK/ETH_RESET 分别接在 MCU 的：PA2/PC1/PG13/PG14/PG11/PC4/PC5/PA7/PA1/PD3 上。特别注意：网络部分 ETH_MDIO 与 USART2_TX 共用 PA2，所以网络和串口 2 的发送，不可以同时使用，但是可以分时复用。

2.1.28 I2S 音频编解码器

神舟号 STM32F4 开发板板载 WM8978 高性能音频编解码芯片，其原理图如图 2.1.28.1 所示：

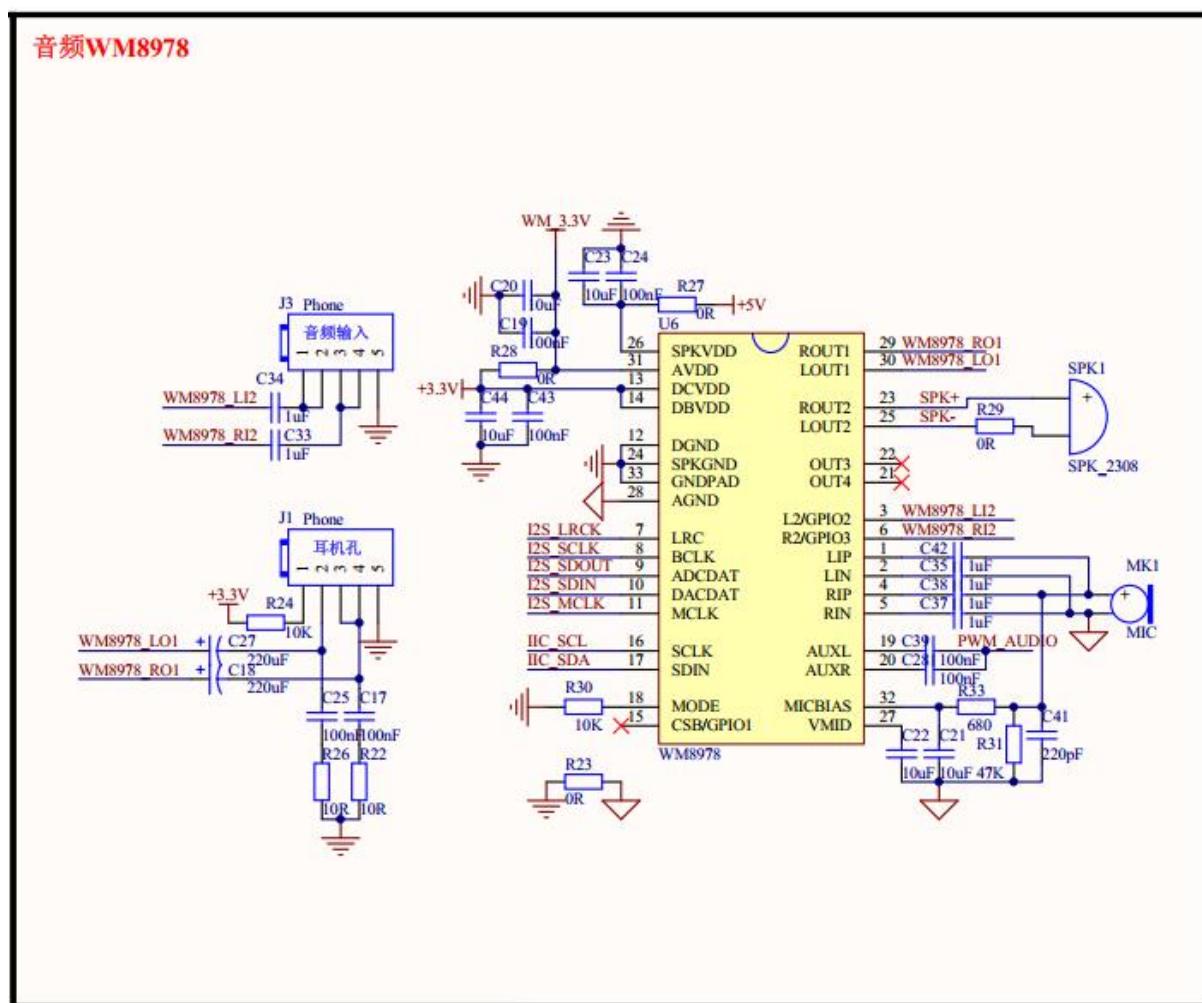


图 2.1.28.1 I2S 音频编解码芯片

WM8978 是一颗低功耗、高性能的立体声多媒体数字信号编解码器。该芯片内部集成了 24 位高性能 DAC&ADC，可以播放最高 192K@24bit 的音频信号，并且自带段 EQ 调节，支持 3D 音效等功能。不仅如此，该芯片还结合了立体声差分麦克风的前置放大与扬声器、耳机和差分、立体声线输出的驱动，减少了应用时必需的外部组件，直接可以驱动耳机（16 Ω @40mW）和喇叭（8 Ω /0.9W），无需外加功放电路。

图中，SPK-和 SPK+连接了一个板载的 8 Ω 1W 小喇叭（在开发板背面），以获得更好音质。MIC

是板载的咪头，可用于录音机实验，实现录音。PHONE 是 3.5mm 耳机输出接口，可以用来插耳机。LINE_IN 则是线路输入接口，可以用来外接线路输入，实现立体声录音。

该芯片采用 I2S 接口与 MCU 连接，图中：I2S_LRCK/I2S_SCLK/I2S_SDOUT/I2S_SDIN/I2S_MCLK/IIC_SCL/IIC_SDA 分别接在 MCU 的：PB12/PB13/PC2/PC3/PC6/PB8/PB9 上。特别注意：I2S_MCLK 和 DCMI_D0 共用 PC6，所以 I2S 音频播放和 OLED 模块/摄像头模块不可以同时使用。另外，IIC_SCL 和 IIC_SDA 是与 24C02/MPU6050 等共用一个 IIC 接口。

2.1.29 电源

神舟号 STM32F4 开发板板载的电源供电部分，其原理图如图 2.1.29.1 所示：

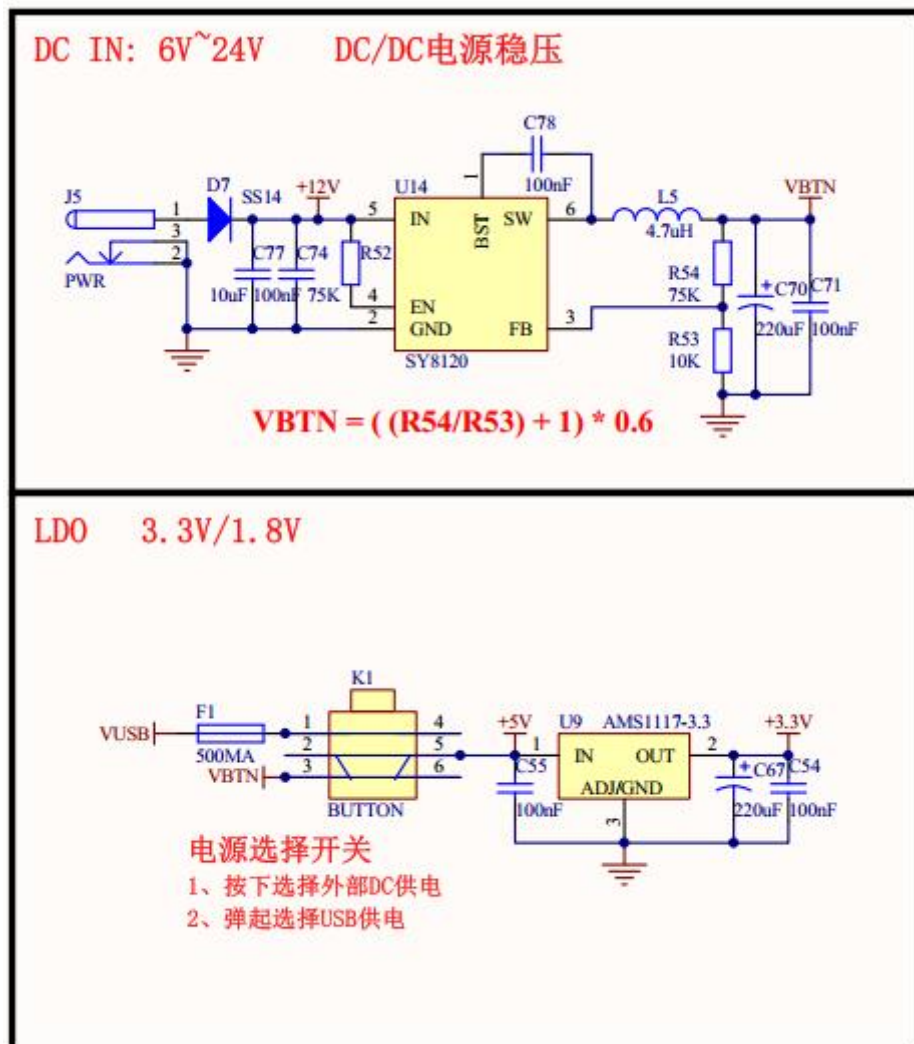


图 2.1.29.1 电源

图中，总共有 3 个稳压芯片：U14/U9，DC_IN 用于外部直流电源输入，经过 U14 DC-DC 芯片转换为 5V 电源输出，其中 D7 是防反接二极管，避免外部直流电源极性搞错的时候，烧坏开发板。K1 为开发板的总电源开关，F1 为 500ma 自恢复保险丝，用于保护 USB。U9 为 3.3V 稳压芯片，给开发

板提供 3.3V 电源，

这里还有 USB 供电部分没有列出来，其中 VUSB 就是来自 USB 供电部分，我们将在相应章节进行介绍。

2.1.30 电源输入输出接口

神舟号 STM32F4 开发板板载了两组简单电源输入输出接口，其原理图如图 2.1.30.1 所示：

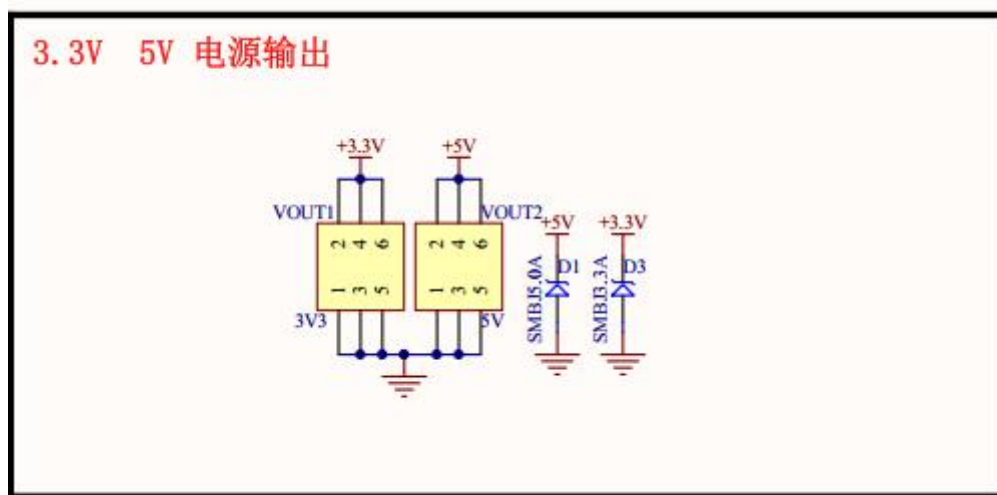


图 2.1.30.1 电源

图中，VOUT1 和 VOUT2 分别是 3.3V 和 5V 的电源输入输出接口，有了这 2 组接口，我们可以通过开发板给外部提供 3.3V 和 5V 电源了，虽然功率不大（最大 1000ma），但是一般情况都够用了，大家在调试自己的小电路板的时候，有这两组电源还是比较方便的。同时这两组端口，也可以用来由外部给开发板供电。

图中 D1 和 D3 为 TVS 管，可以有效避免 VOUT 外接电源/负载不稳的时候（尤其是开发板外接电机/继电器/电磁阀等感性负载的时候），对开发板造成的损坏。同时还能一定程度防止外接电源接反，对开发板造成的损坏。

2.1.31 USB 串口

神舟号 STM32F4 开发板板载了一个 USB 串口，其原理图如图 2.1.31.1 所示：

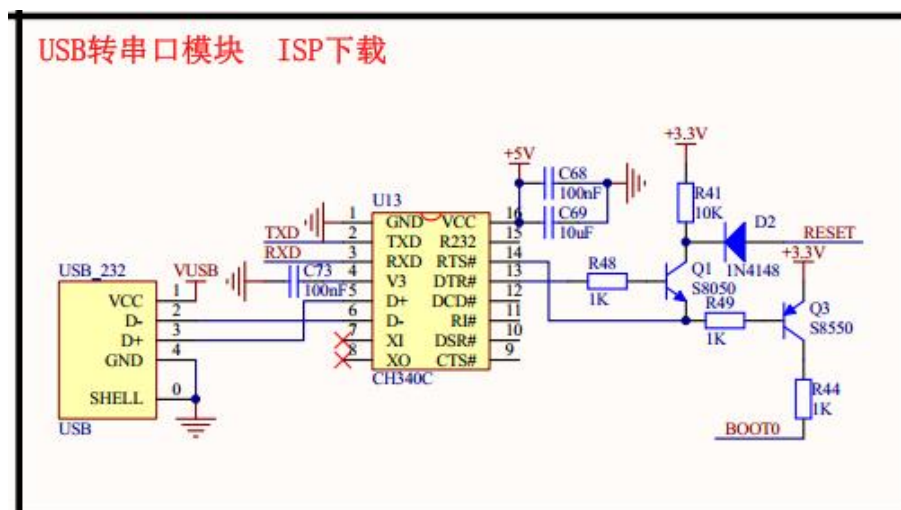


图 2.1.31.1 USB 串口

USB 转串口，我们选择的是 CH340C，是国内芯片公司南京沁恒的产品，稳定性经测试还不错，所以还是多支持下国产。

图中 Q1 和 Q3 的组合构成了我们开发板的一键下载电路，只需要在 flymcu 软件设置：DTR 的低电平复位，RTS 高电平进 BootLoader。就可以一键下载代码了，而不需要手动设置 B0 和按复位了。其中，RESET 是开发板的复位信号，BOOT0 则是启动模式的 B0 信号。

一键下载电路的具体实现过程：首先，mcuisp 控制 DTR 输出低电平，则 DTR_N 输出高，然后 RTS 置高，则 RTS_N 输出低，这样 Q3 导通了，BOOT0 被拉高，即实现设置 BOOT0 为 1，同时 Q1 也会导通，STM32F4 的复位脚被拉低，实现复位。然后，延时 100ms 后，mcuisp 控制 DTR 为高电平，则 DTR_N 输出低电平，RTS 维持高电平，则 RTS_N 继续为低电平，此时 STM32F4 的复位引脚，由于 Q1 不再导通，变为高电平，STM32F4 结束复位，但是 BOOT0 还是维持为 1，从而进入 ISP 模式，接着 mcuisp 就可以开始连接 STM32F4，下载代码了，从而实现一键下载。

USB_232 是一个 MiniUSB 座，提供 CH340C 和电脑通信的接口，同时可以给开发板供电，VUSB 就是来自电脑 USB 的电源，USB_232 是本开发板的主要供电口。

2.2 开发板使用注意事项

为了让大家更好的使用神舟号 STM32F4 开发板，我们在这里总结该开发板使用的时候尤其要注意的一些问题，希望大家在使用的时候多多注意，以减少不必要的问题。

1，开发板一般情况是由 USB_232 口供电，在第一次上电的时候由于 CH340C 在和电脑建立连接的过程中，导致 DTR/RTS 信号不稳定，会引起 STM32 复位 2~3 次左右，这个现象是正常的，后续按复位键就不会出现这种问题了。

2, 1 个 USB 供电最多 500mA, 且由于导线电阻存在, 供到开发板的电压, 一般都不会有 5V, 如果使用了很多大负载外设, 比如 4.3 寸屏、网络、摄像头模块等, 那么可能引起 USB 供电不够, 所以如果是使用 4.3 屏的朋友, 或者同时用到多个模块的时候, 建议大家使用一个独立电源供电。如果没有独立电源, 建议可以同时插 2 个 USB 口, 并插上 JTAG, 这样供电可以更足一些。

3, JTAG 接口有几个信号(JTDI/JTDO/JTRST)被 USB_PWR(USBHOST)/SPI1(W25Q128 和 NRF24L01)占用了, 所以在调试这几个模块的时候, 请大家选择 SWD 模式, 其实最好就是一直用 SWD 模式。

4, 当你想使用某个 IO 口用作其他用处的时候, 请先看看开发板的原理图, 该 IO 口是否有连接在开发板的某个外设上, 如果有, 该外设的这个信号是否会对你的使用造成干扰, 先确定无干扰, 再使用这个 IO。比如 PF8 就不怎么适合再用做其他输出, 因为他接了蜂鸣器, 如果你输出高电平就会听到蜂鸣器的叫声了。

5, 开发板上的跳线帽比较多, 大家在使用某个功能的时候, 要先查查这个是否需要设置跳线帽, 以免浪费时间。

6, 当液晶显示白屏的时候, 请先检查液晶模块是否插好(拔下来重新插试试), 如果还不行, 可以通过串口看看 LCD ID 是否正常, 再做进一步的分析。

7, 开发板的 USB SLAVE 和 USB HOST 共用同一个 USB 口, 所以, 他们不可以同时使用。使用的时候多加注意。

第三章 DAP 编程器仿真器

3.1 产品特点

精畅科技推出的 CMSIS DAP 仿真器主要是针对 Cortex-M0/M1/M2/M3/M4/M7 内核的芯片，用于程序的下载和调试，产品具有如下特性：

- 无版权限制，不会丢失固件
- 无须驱动，即插即用(WIN11 系统下)
- 自带虚拟串口，目标设备可以通过仿真器与计算机实现串口通信
- 使用SWD 接口或者 JTAG 接口与设备连接，无需硬件复位线，简化连接

输出3.3V和5V 电源，直接给目标板供电



3.2 引脚排列

CMSIS DAP 仿真器通过 2.54 间距的双排针引出信号线和电源线，引出的电源为 3.3V 和 5V，用户可通过杜邦线与目标板进行连接，实现下载、调试、供电和串口通信。引脚排列如下：

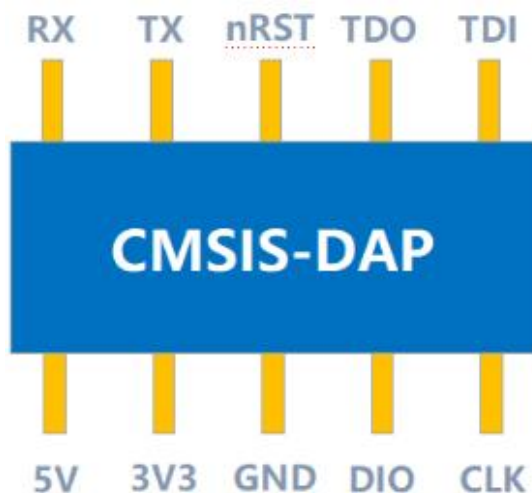


图 1.2 接口示意图

名称	说明	名称	说明
RX	串口接收引脚	5V	5V 电源，可对目标板供电
TX	串口发送引脚	3V3	3.3V 电源，可对目标板供电
nRST	复位引脚	GND	电源地
TDO	JTAG 数据串行输出	DIO	即 SWDIO，数据线
TDI	JTAG 数据串行输入	CLK	即 SWCLK，时钟线

3.3 性能测试

在 MDK5.23 下,使用 CMSIS DAP 仿真器进行下载测试,SWD 接口速度为 10M,测试芯片为 STM32F429,使用不同大小的程序进行测试,和 Jlink V8 做对比,数据如下:

	33K	92K	192K	700K
CMSIS DAP	4s	9s	16s	55s
Jlink V8	3s	7s	12s	33s

更换测试芯片为 STM32F103ZGT6, 数据如下:

	29K	76K	91K	294K
CMSIS DAP	5s	10s	11s	35s
Jlink V8	4s	8s	9s	27s

由于 CMSIS DAP 仿真器很多版本, 以上的测试数据是基于精畅科技的 DAP 仿真器测试得来的, 结果仅供参考。

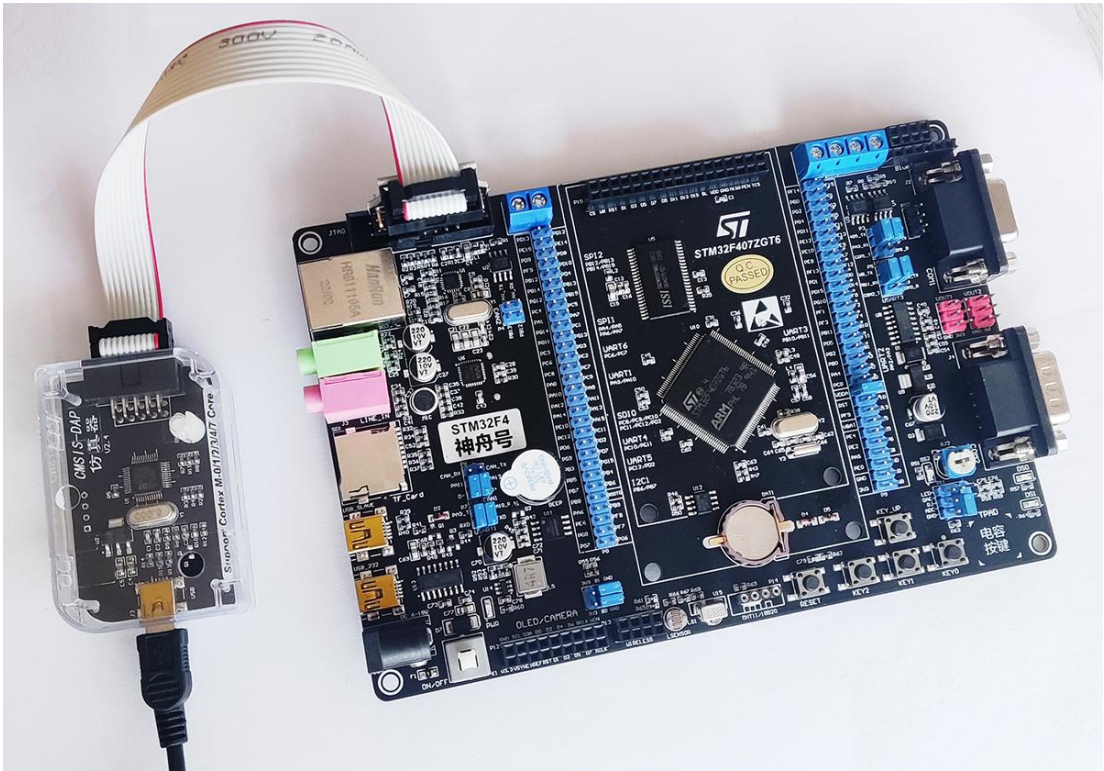
3.4 使用说明

CMSIS DAP 插入电脑后, 会自动识别为 USB HID 设备和串口设备, 在 win10 系统下无需安装驱动即可正常使用, 若使用的是 win7 或 win8 系统, 则需要手动添加串口驱动, 具体安装方法将在第三章进行介绍。



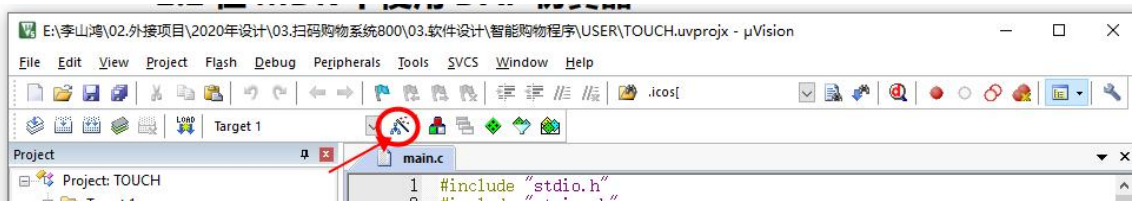
图 2.1 设备管理器示意图

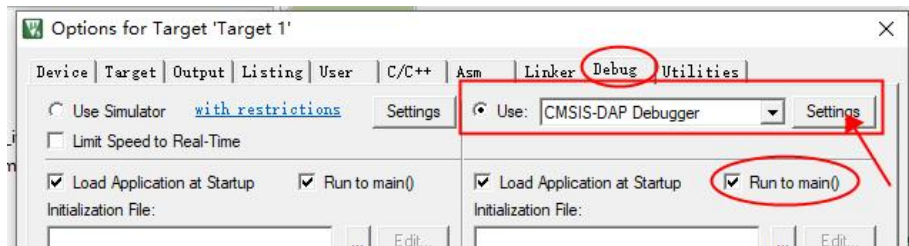
在使用之前，请确保已经进行了正确的连线，下图为使用转接排线连接图示



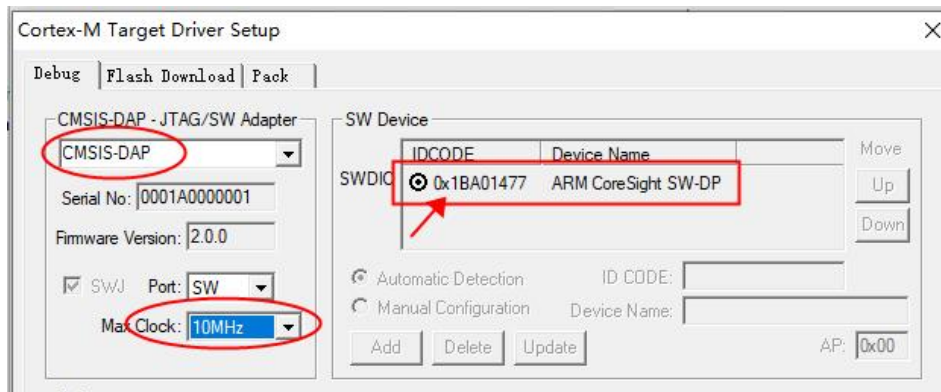
3.4.1 在 MDK 下使用 DAP 仿真器

在 MDK 中，进行简单的配置即可使用 CMSIS DAP 仿真器，首先点击“魔术棒”按钮：

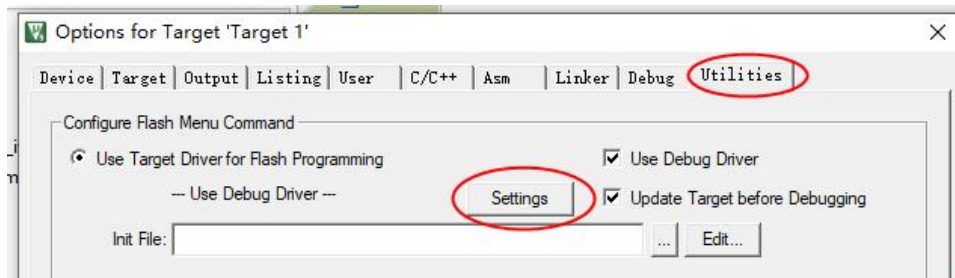




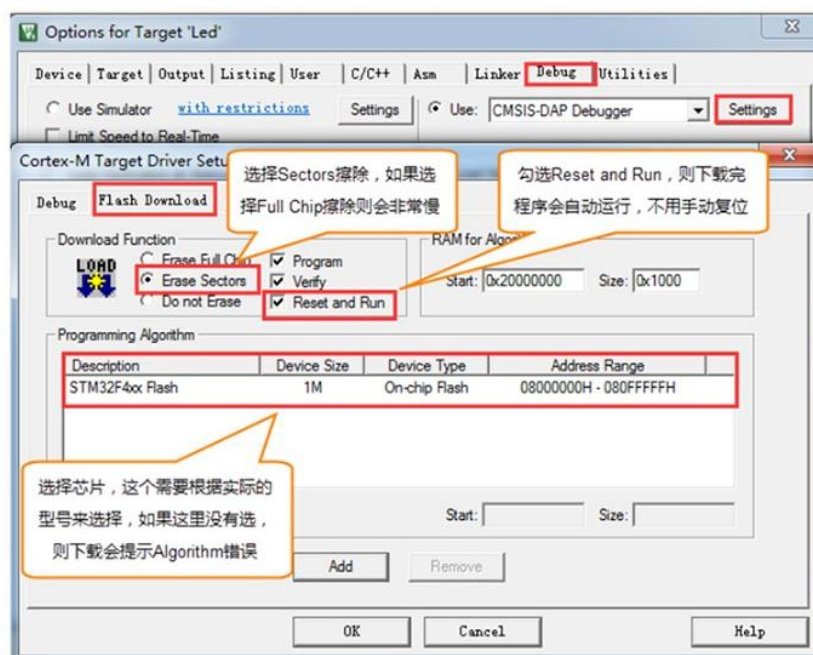
在下拉框里选择已经识别到的仿真器设备，设置为 SW 模式，其它使用默认设置即可，如果设备正常连接会在右边显示目标芯片的 IDCODE，通信速度可以设置为最大 10M。



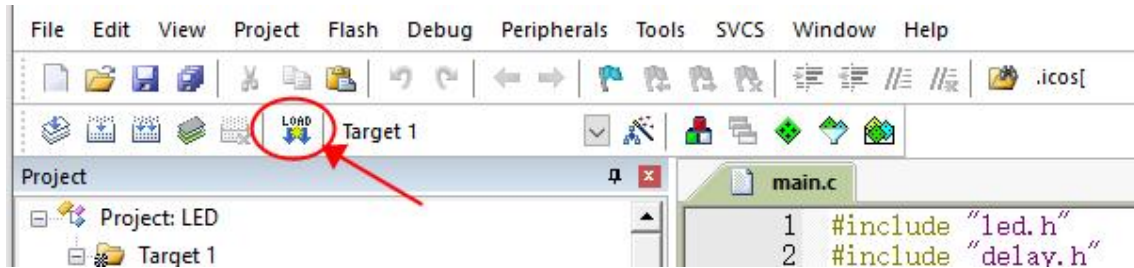
接着点击 Utilities 选项卡，点击 Settings，进行相关的下载设置：



在弹出的选项卡中，勾选 Reset and Run 选项，选择芯片对应的型号。

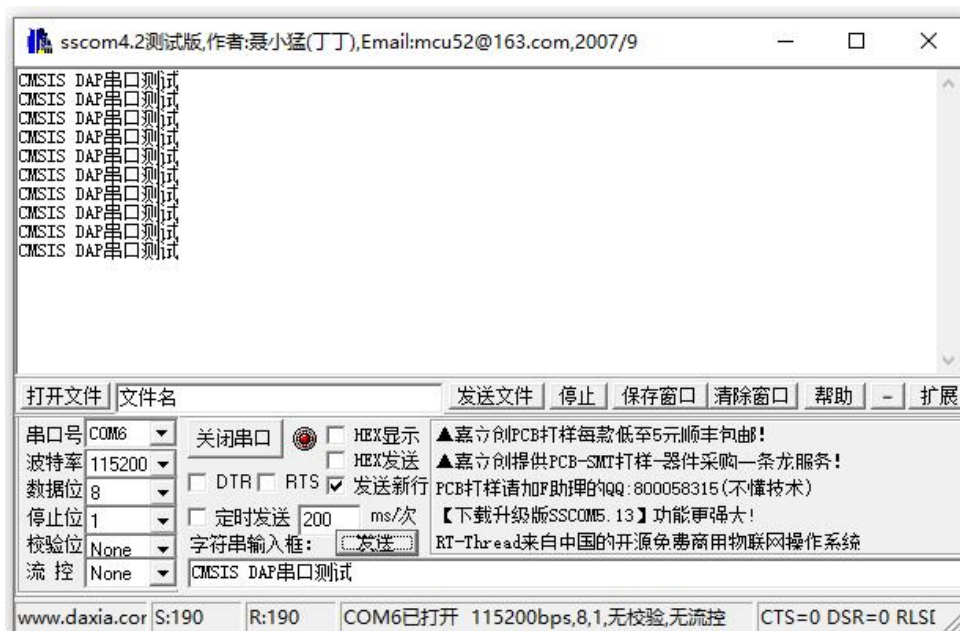


设置完毕之后，回到主界面，在菜单栏点击下载按钮即可下载程序，同时也可以使用 MDK 的调试功能对目标设备进行调试。



3.4.2 使用虚拟串口功能

CMSIS DAP 的串口使用方法和一般的 USB 转串口设备无异，先是在设备管理器中找到串口的设备号，然后打开串口软件选择相应的串口设备即可。

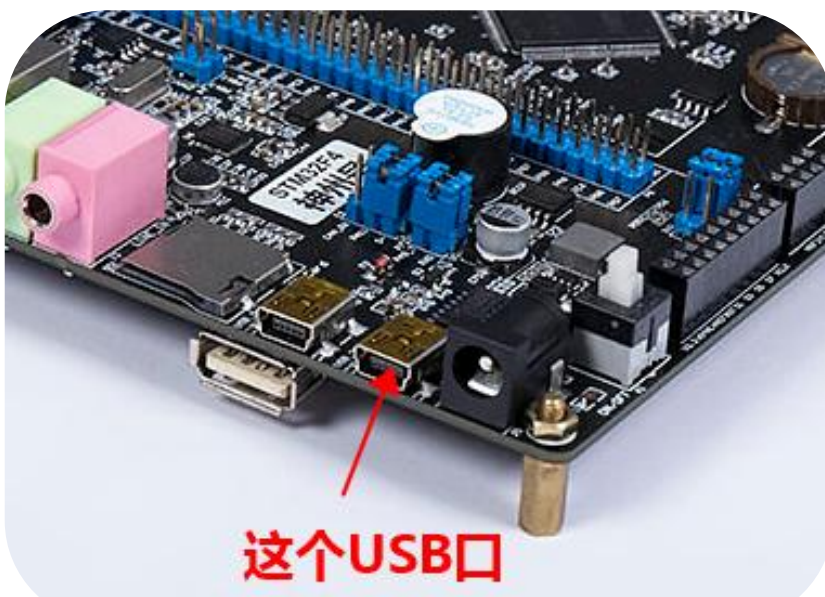


第二篇 常见问题汇总

Q1: 找不到 USB 转串口?

A: 这个问题可能有几个原因:

- 1, 开发板没供电。这种情况, 请检查开发板右下角的蓝色电源灯, 是否亮了? 如果没亮, 请按电源开关。电源指示灯一定要亮, 电脑才能找到 USB 转串口。
- 2, 插错 USB 口了。要使用 USB 转串口, USB 线一定要插开发板左下角的 USB_232 接口, 才可以。如图 Q1.1 所示:



Q1.1 USB 转串口插头插法

- 3, USB 驱动没有安装。请安装 CH340C 驱动, CH340 的驱动在光盘资料: 5. 常用软件\软件\CH340 驱动(USB 串口驱动)_XP_WIN7 共用 文件夹里面。

Q2: flymcu 无法下载代码?

A: 这个问题非常常见。常见的问题原因有如下几个:

- 1, flymcu 软件的 DTR, RTS 设置错了。正确的设置是: **DTR 的低电平复位, RTS 的高电平进 BootLoader**, 如图 Q2.1 所示:



Q2.1 flymcu 设置

图Q2.1 所示：**不要勾选：使用 RamIsp、不要勾选：编程到 FLASH 时写选项字节，DTR,RTS 不要设置错了，串口号要选对。**

- 2, 串口选错了。这个一定要检查好, 你的开发板 USB 转串口是哪一个? 在设备管理器里面, 可以查看 USB 转串口的类型, 如图 Q2.2 所示:



图 Q2.2 我电脑的 USB 转 TTL 串口端口

图Q2.2 中, 可以看到计算机总共有 2 个COM 口, 而很明显我们可以知道, 开发板的USB 转串口, 是 COM3。因为开发板是 USB 转串口, 所以肯定是 USB-SERIAL, 另外用的是 CH340G 芯片, 所以必定是 COM3 了。**特别注意:** 你的电脑不一定是 COM3, 但是 USB-SERIAL CH340, 这个字符串一定是一样的, 请根据具体情况选择!

Q3: CMSIS DAP 仿真器 能否给开发板供电?

A: 不推荐。因为开发板功耗比较大, DAP 如果又给自己供电又给开发板供电, 有可能导致 DAP 固件丢失。

Q4: CMSIS DAP 仿真器 不能下载代码?

A: 请参考: 开发板光盘资料: “9. CMSIS DAP 仿真器资料”

Q5: flymcu 下载代码后, 不运行?

A: 这个问题可能有如下 2 种情况:

- 1, flymcu 没有勾选：编程后执行。这个，勾选这个选项，重新下载即可，或者按开发板的复位也行。
- 2, 开发板的 B0, B1 没有都接 GND。检查开发板的 B0, B1 是不是都接 GND 了，一定要都接GND 才可以。

Q6: CAN 实验，普通模式，无法发送数据？

A: CAN 实验必须要 2 个或 2 个以上节点才可以正常通信。CAN 不像串口和 485，单个节点也可以发送（不管对方收没收到），CAN 一定要 2 个，或者以上节点，并且程序没问题，才可以发送成功。

Q7: 综合测试实验，提示：Font error/System File Error?

A: 这个问题，一般是由于存放在 SPI FLASH 的SYSTEM 文件夹意外丢失导致的。必须重新拷贝SYSTEM 文件夹到 SPI FLASH，才可以进入主界面了。解决办法有 2 个：

方案 1: 准备一个 SD 卡（小卡 TF），并拷贝 SYSTEM 文件夹（**注意：这个 SYSTEM 文件夹不是开发板例程里的 SYSTEM 文件夹，而是光盘根目录→SD 卡根目录文件→SYSTEM 文件夹**）到 SD 卡根目录，然后 SD 卡插入开发板，按复位重启，然后开发板会自动更新文件。

方案 2: 按 KEY2 不放，然后按复位，松开复位（此过程 KEY2 一直按住不放）直到红字提示：**Erase all system files?**，松开 KEY2，然后按 KEY2，选择擦除所有文件。然后，等待系统重启。然后将 USB 线插入 USB_SLAVE 接口，然后根据提示，按 KEY1 选择使用 USB 更新 SPI FLASH Files，等待系统提示：**USB Connected** 后，电脑会找到一个：STM32F1 的磁盘，然后将：**光盘根目录→SD 卡根目录文件→SYSTEM 文件夹**，拷贝到 STM32F1 磁盘根目录。等待拷贝完成后，按 KEY0，此时系统自动重启，并自动更新所有文件（全自动，无需用户干预）等所有操作完成后，即可进入主界面。

Q8: 综合测试实验，触摸屏不准/无法触摸？

A: 这种情况，请校准一下触摸屏即可。

校准方法：按住 KEY0 不放，按复位，松开复位（此过程 KEY0 一直按住不放）等加载到 Touch Check 时，系统进入**触摸校准**界面（仅 2.8/3.5 等电阻屏有）松开 KEY0，然后，用笔尖依次点击屏幕显示十字架的最中心，即可完成校准，校准完成，屏幕会提示：**Touch Screen Adjust OK**，表示校准完成。

Q9: 串口实验（实验 3），发送数据无返回？

A: 这个问题一般可能是 2 个原因引起的：

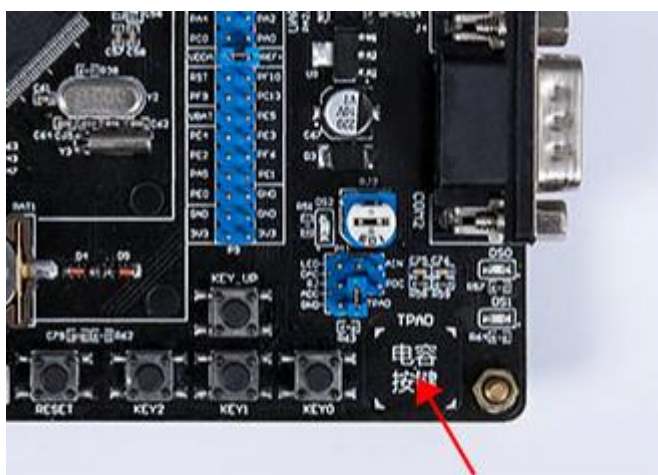
- 1, 串口选错了。请选择正确的串口（方法见 Q2.2）
- 2, 没有发送新行（或者输入回车符）用 SSCOM 和 XCOM 的朋友，注意，一定要勾选：发送新行，才可以，如图 Q9.1 所示、



图 Q9.1 勾选发送新行

Q10: 综合测试实验，某些界面无法退出？

A: 综合测试实验，某些界面，比如画图、记事本、电子图书等，在屏幕上面，没有返回按钮，这个时候，我们可以按开发板右下角的 TPAD 触摸按键，即可退出当前界面，回到主界面。如图 Q10.1 所示：



手指轻轻按这个区域就会返回

图 Q10.1 综合实验万能“返回按钮”

TPAD 是一个电容触摸按键，手指轻轻一摸，即可实现按键输入，在综合实验，任何界面下，都可以通过“按”这个按键返回。

Q11: 下载程序后，LCD 不亮/黑屏？

A: 这个问题，是因为你下载的代码，根本就没用到 LCD，所以 LCD 就不会亮了。我们提供的例程，**TFTLCD 显示实验 以前的所有例程**，下载进去，LCD 都不会亮。因为这些例程，都没用到 LCD。

Q12: 综合实验，为什么液晶一会亮，一会暗，感觉在闪烁？

A: 这个问题，是因为综合实验开启了自动背光导致的，因为开发板自带了光敏电阻，可以检测环境光线强度，从而控制 LCD 背光亮度，环境光越强，背光越亮，反之越暗。所以，这是正常现象，如果不想要此功能，可以在综合实验→系统设置→背光设置 里面，将背光设置不为 0 即可（0=开启自动背光）

第三篇 STM32 程序工程结构

一、组成工程结构的文件夹

CORE、HARDWARE、OBJ、STM32F10x_FWLib、SYSTEM、USER

由以上的文件夹组成一个程序工程结构

二、工程中每个文件夹里面的内容介绍

- 1、“CORE”：存放 STM32 启动文件、核心文件等
- 2、“HARDWARE”：存放 STM32 外设的驱动文件，如 led, key, lcd, adc, dac 等
- 3、“OBJ”：存放 MDK 编译产生的中间文件和 hex 文件
- 4、“STM32F40x_FWLib”：存放 STM32 库文件
- 5、“SYSTEM”：存放滴答延时驱动、串口 1 驱动以及系统文件
- 6、“USER”：存放工程的打开文件以及主函数 main.c 和中中断服务文件

三、说明文档“README.TXT”

说明本实验例程要准备的实验器材、实验目的、实验资源以及实验现象。

附录 神舟号 STM32F4 开发板原理图

1

2

3

4

STM32F4神舟号_原理图_V2.0

目 录

Page 1	目 录
Page 2	历史版本
Page 3	主电源/USB232
Page 4	MCU/SRAM/液晶屏/引出IO/JTAG/SWD/复位键
Page 5	RS232/蓝牙/wifi/无线/摄像头/RS485/DHT11/LED/SD卡/ CAN/USB/红外/按键/SPI-FLASH/EEPROM/蜂鸣器/光照/ADC
Page 6	以太网LAN8720A/音频WM8978

1

2

3

4

Title

STM32F4神舟号_原理图

Size

A4

Date:

2021-04-09

File:

E:\STM32F4神舟号原理图\STM32F4神舟号\asdboc

Number

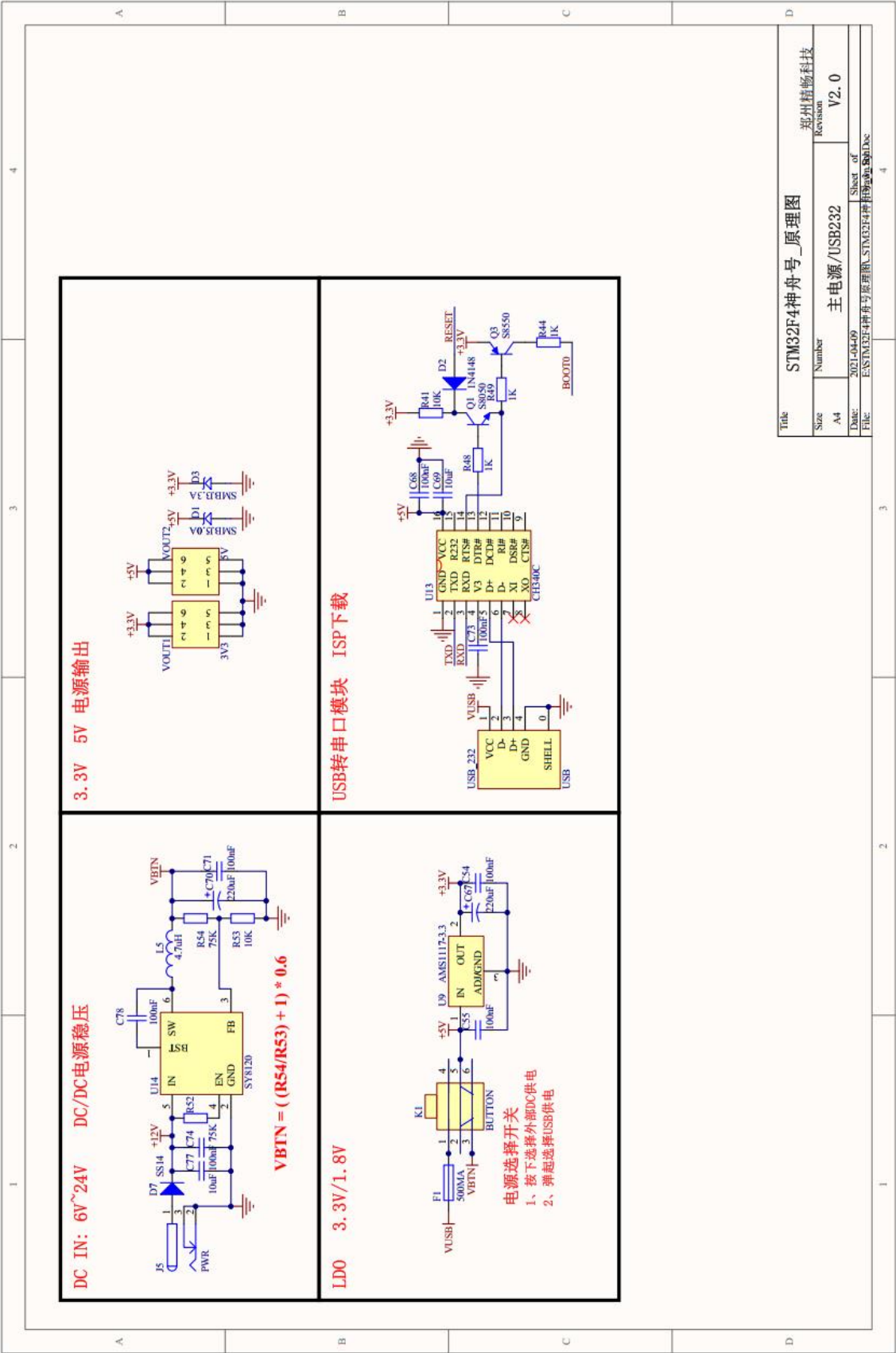
Revision

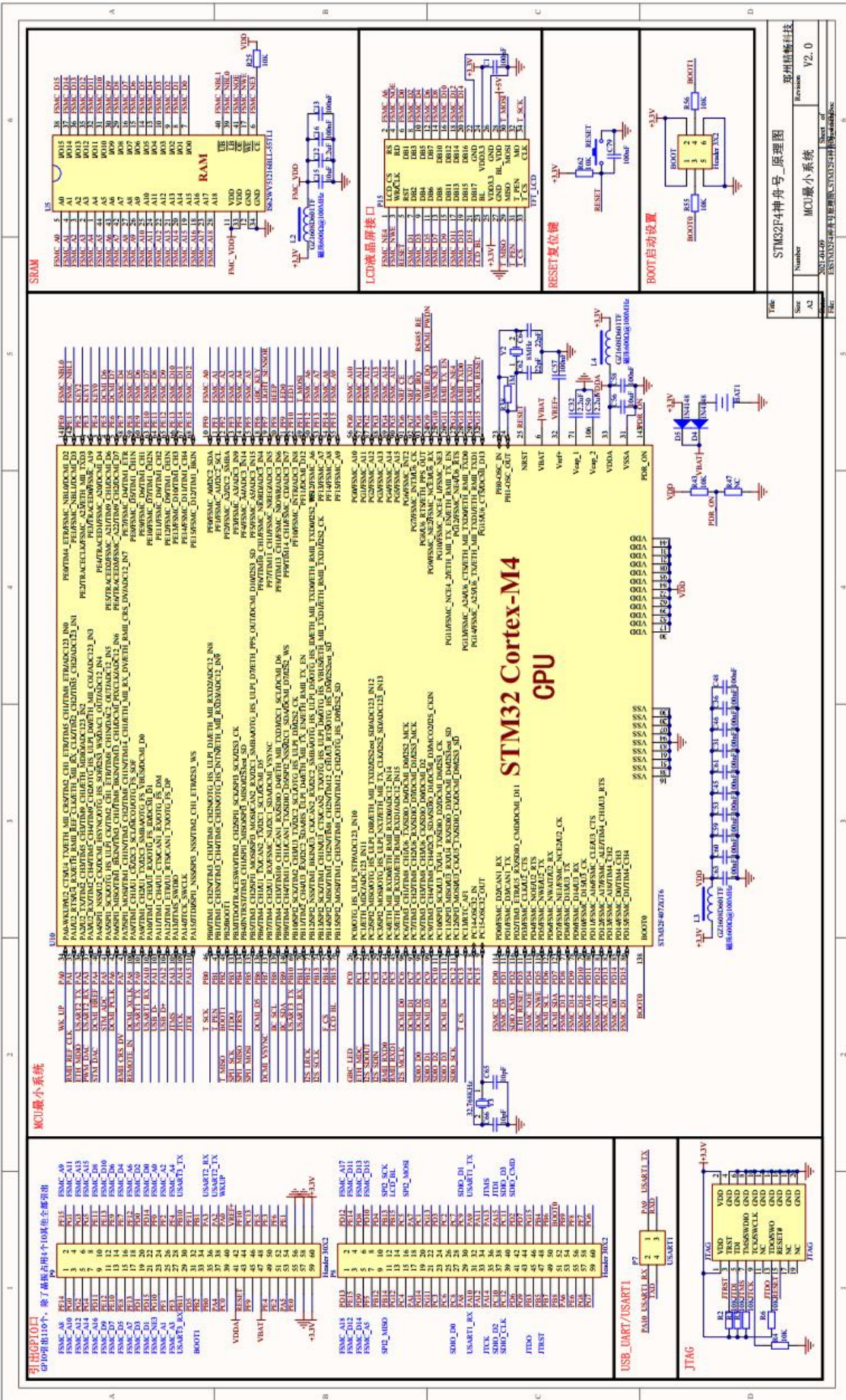
V2.0

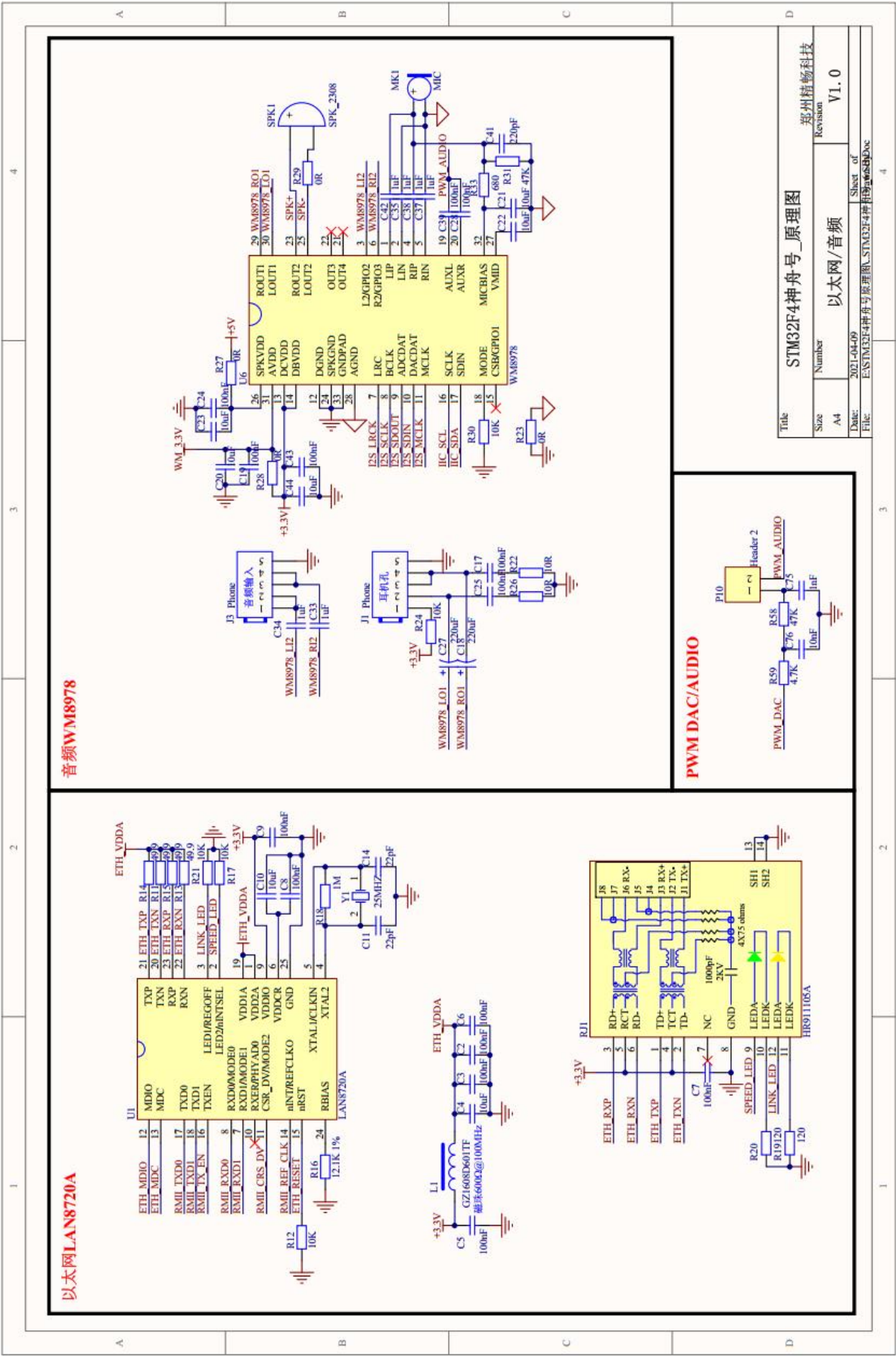
Sheet of

4

郑州精畅科技







郑州精畅科技有限责任公司

QQ 讨论群: **1030179400**

技术咨询 QQ: **3614918773** (李工)

E-mail: 3614918773@qq.com