

E32R40T&E32N40T

4.0 寸 ESP32-32E 显示模块 用户手册



目 录

1. 资料包说明.....	3
2. 软件使用说明.....	4
3. 硬件使用说明.....	4
3.1. 显示模块硬件资源简介.....	4
3.2. 显示模块原理图详解.....	9
3.3. 显示模块使用注意事项.....	18

1. 资料包说明

资料目录如下图所示：

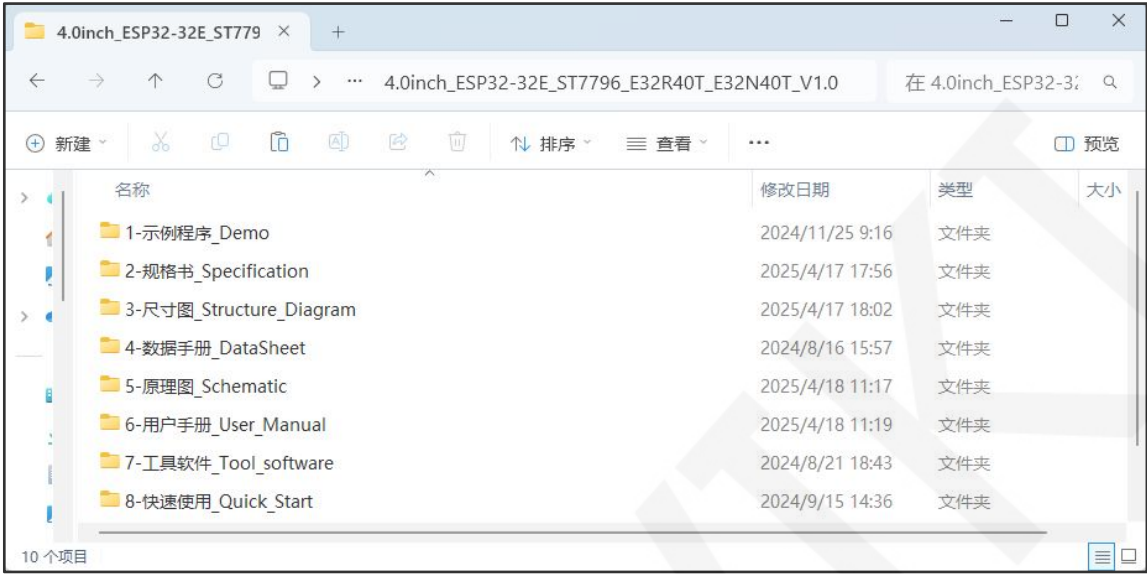


图 1.1 产品资料包目录

目录	内容说明
1-示例程序_Demo	示例程序代码、示例程序依赖的第三方软件库、第三方软件库替换文件、软件开发环境搭建说明文档以及示例程序说明文档。
2-规格书_Specification	显示模块产品规格书、液晶屏规格书以及液晶屏显示驱动 IC 初始化代码。
3-尺寸图_Structure_Diagram	显示模块产品尺寸图和产品 3D 图
4-数据手册_DataSheet	液晶屏显示驱动 ST7796 数据手册、电阻触摸屏驱动 XPT2046 数据手册、ESP32 主控数据手册以及硬件设计指导文档、USB 转串口 IC (CH340C) 数据手册、音频功放芯片 FM8002E 数据手册、5V 转 3.3V 稳压管数据手册以及电池充电管理芯片 TP4054 数据手册。
5-原理图_Schematic	产品硬件原理图、ESP32-32E 模块 IO 资源分配表、原理图和 PCB 元器件封装
6-用户手册_User_Manual	产品用户说明文档
7-工具软件_Tool_software	WIFI 和蓝牙测试 APP 以及调试工具、USB 转串口驱动、ESP32 Flash 下载工具软件、字符取模软件、图片取模软件、JPG 图片处理软件以及串口调试工具。
8-快速使用_Quick_Start	需要烧录的 bin 文件、flash 下载工具及使用说明文档。

2. 软件使用说明

显示模块软件开发步骤如下：

- A、搭建 ESP32 平台软件开发环境；
- B、如有需要，则导入第三方软件库作为开发基础；
- C、打开需要调试的软件项目工程，也可以新建一个软件项目工程；
- D、给显示模块上电，编译并下载调试的程序，然后查看软件运行效果；
- E、软件效果没达到预期，继续修改程序代码，然后编译并下载，直到效果到达预期；

以上步骤的详细说明见“1-示例程序_Demo”目录下的各个说明文档。

3. 硬件使用说明

3.1. 显示模块硬件资源简介

模块硬件资源如下两图所示：



图 3.1 模块硬件资源 1

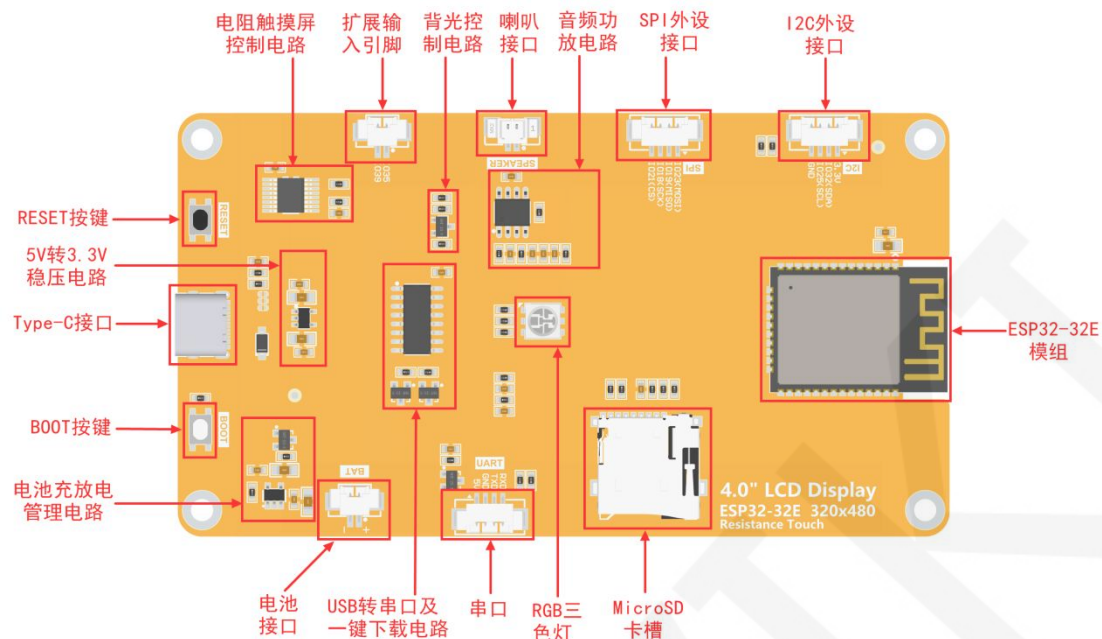


图 3.2 模块硬件资源 2

硬件资源说明如下：

1) 液晶屏

液晶屏显示尺寸为 4.0 寸，驱动 IC 为 ST7796，分辨率为 320x480。使用 4 线制 SPI 通信接口和 ESP32 相连。

A、ST7796 控制器简介

ST7796 控制器支持的最大分辨率为 320*480，拥有一个 345600 字节大小的 GRAM。同时支持 8 位、9 位、16 位、18 位以及 24 位并口数据总线，还支持 3 线制和 4 线制 SPI 串口。由于并行控制需要大量的 IO 口，所以最常用的还是 SPI 串口控制。ST7796 还支持 65K、262K、16.7M RGB 颜色显示，显示色彩很丰富，同时支持旋转和滚动显示以及视频播放，显示方式多样。

ST7796 控制器使用 16bit (RGB565) 来控制一个像素点显示，因此可以每个像素点显示颜色多达 65K 种。像素点地址设置按照行列的顺序进行，递增递减方向由扫描方式决定。ST7796 显示方法按照先设置地址再设置颜色值进行。

B、SPI 通信协议简介

4 线制 SPI 总线写模式时序如下图所示：

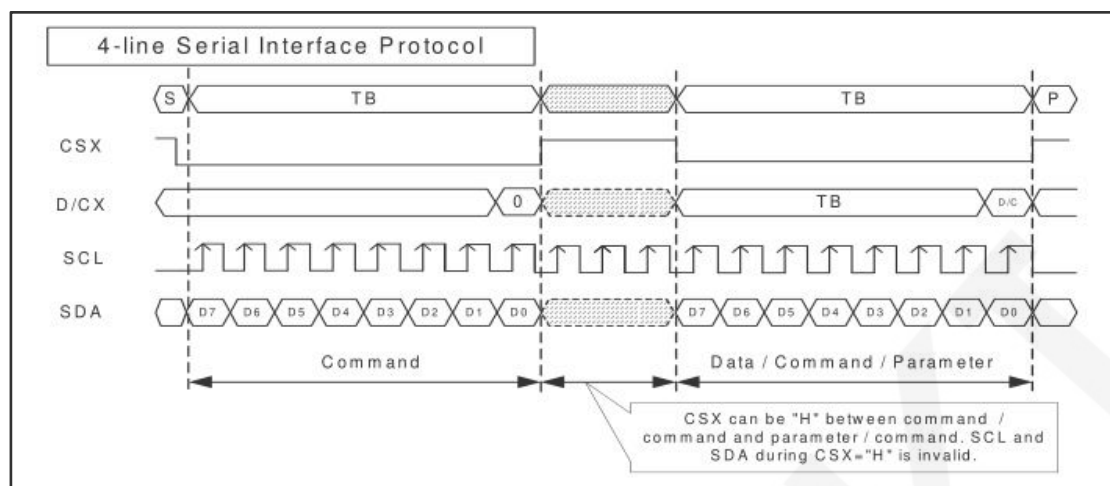


图 3.3 4 线制 SPI 总线写模式时序

CSX 为从机片选，仅当 CSX 为低电平时，芯片才会被使能。

D/CX 为芯片的数据/命令控制引脚，当 DCX 为低电平时写命令，为高电平时写数据

SCL 为 SPI 总线时钟，每个上升沿传输 1bit 数据；

SDA 为 SPI 传输的数据，一次传输 8bit 数据，数据格式如下图所示：

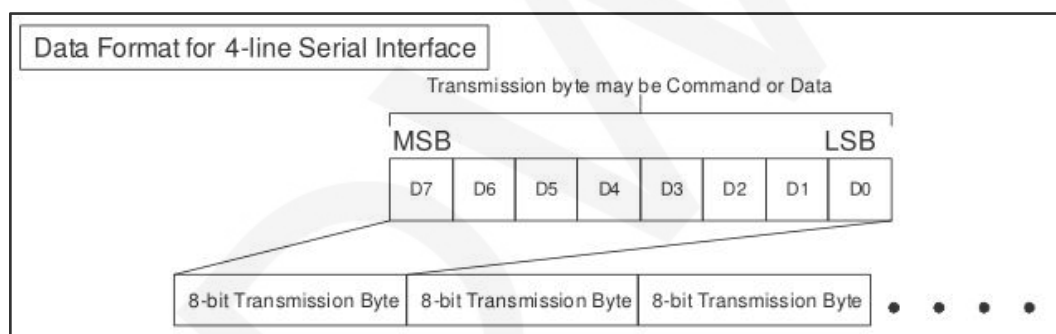


图 3.4 4 SPI 传输数据格式

高位在前，先传输。

对于 SPI 通信而言，数据是有传输时序的，即时钟相位 (CPHA) 与时钟极性 (CPOL) 组合：

CPOL 的高低决定串行同步时钟的空闲状态电平，CPOL = 0，为低电平。CPOL 对传输协议没有很多的影响；

CPHA 的高低决定串行同步时钟是在第一时钟跳变沿还是第二个时钟跳变沿数据被采集，当 CPHA = 0，在第一个跳变沿进行数据采集；

这两者组合就成为四种 SPI 通信方式，国内通常使用 SPI0，即 CPHA = 0，CPOL = 0。

2) 电阻触摸屏

电阻触摸屏尺寸为 4.0 寸，通过 XL、XR、YU、YD 四个引脚和 XPT2046 控住 IC 相连。

3) ESP32-32E 模组

此模组内置 ESP32-D0WD-V3 芯片，Xtensa 双核 32 位 LX6 微处理器，支持高达 240MHz 的时钟频率。拥有 448KB ROM, 520KB SRAM, 16KB RTC SRAM, 4MB QSPI Flash。支持 2.4GHz WIFI、蓝牙 V4.2 以及低功耗蓝牙模组。外置 26 个 GPIO，支持 SD 卡、UART、SPI、SDIO、I2C、LED PWM、电机 PWM、I2S、IR、脉冲计数器、GPIO、电容式触摸传感器、ADC、DAC、TWAI 等外设。

4) MicroSD 卡槽

使用 SPI 通信方式和 ESP32 连接，支持各种容量的 MicroSD 卡。

5) RGB 三色灯

红、绿、蓝三种颜色 LED 灯，可用于指示程序运行状态。

6) 串口

外接串口模块，用于模块串口通信。

7) USB 转串口及一键下载电路

核心器件是 CH340C，一端连接电脑端 USB，一端连接 ESP32 串口，从而实现 USB 转 TTL 串口。另外还附加了一键下载电路，就是下载程序时，可以自动进入下载模式，无需在通过外部触摸。

8) 电池接口

两针接口，一针为正极，一针为负极，接入电池供电和充电。

9) 电池充放电管理电路

核心器件是 TP4054，此电路可控制电池充电电流，是电池安全充电到饱和状态，同时也可以安全控制电池放电。

10) BOOT 按键

显示模块通电后，按下会将 I00 拉低。如果在模块上电的瞬间或者 ESP32 复位时，拉低 I00 就会进入下载模式。其他情况可以做普通按键使用。

11) Type-C 接口

显示模块的主供电接口和程序下载接口。连接 USB 转串口及一键下载电路，可用于供电、下载以及串口通信。

12) 5V 转 3.3V 稳压电路

核心器件是 ME6217C33M5G LDO 稳压管。此稳压电路支持 2V~6.5V 宽电压输入，支持 3.3V 稳定电压输出，最大输出电流为 800mA，完全可以满足显示模块的电压和电流需求。

13) RESET 按键

显示模块通电后，按下会将 ESP32 复位引脚拉低（默认状态为拉高），从而实现复位功能。

14) 电阻触摸屏控制电路

核心器件是 XPT2046，通过 SPI 方式和 ESP32 通信。此电路为电阻触摸屏和 ESP32 主控的桥梁，负责将触摸屏上的数据传输给 ESP32 主控，从而获取触摸点的坐标。

15) 扩展输入引脚

ESP32 模组上两个没有使用的输入 IO 口，引出来给外设使用。

16) 背光控制电路

核心器件是 BSS138 场效应管。此电路一端连着 ESP32 主控上的背光控制引脚，一端连着液晶屏背光 LED 灯的负极。背光控制引脚拉高，背光亮，反之则灭。

17) 喇叭接口

接线端子需垂直接入。用于接入单声道喇叭和扬声器。

18) 音频功放电路

核心器件是 FM8002E 音频功放 IC。此电路一端连接 ESP32 音频 DAC 值输出引脚，一端连接喇叭接口。此电路功能就是驱动小功率喇叭或者扬声器发声。5V 供电时，最大驱动功率为 1.5W（负载 8 欧）或者 2W（负载 4 欧）。

19) SPI 外设接口

4 线的水平接口。将一个没使用的片选引脚和 MicroSD 卡使用的 SPI 接口引脚引出，可以外接 SPI 设备，也可以做普通 IO 口使用。

20) I2C 外设接口

4 线的水平接口。将两个没使用的引脚引出做 I2C 接口，可以外接 IIC 设备，也可以做普通 IO 口使用。

3.2. 显示模块原理图详解

1) Type-C 接口电路

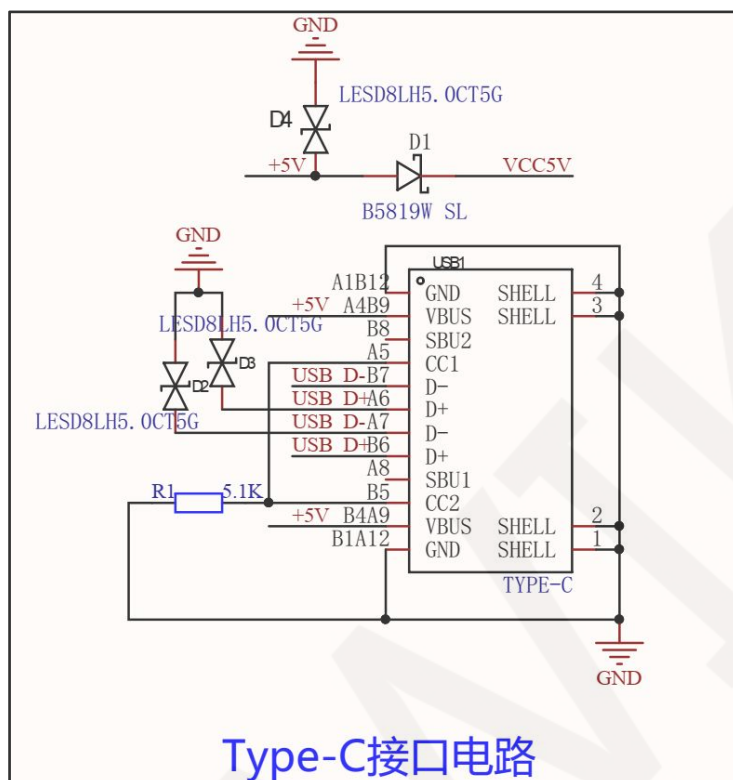


图 3.5 Type-C 接口电路

此电路中，D1 为肖特基二极管，用来防止电流反向。D2~D4 为静电浪涌保护二极管，防止因电压过大或者短路损坏显示模块。R1 为下拉电阻。USB1 为 Type-C 母座。显示模块通过 USB1 接入 Type-C 供电、下载程序以及串口通信。其中+5V 和 GND 为电源正电压和地信号，USB_D-和 USB_D+为 USB 差分信号，传输到板载 USB 转串口电路。

2) 5V 转 3.3V 稳压电路

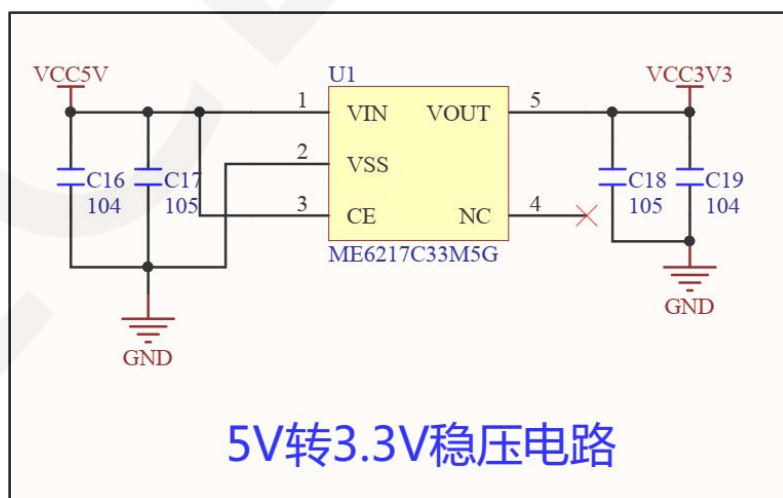


图 3.6 稳压电路

此电路中，C16~C19 为旁路滤波电容，用来维持输入电压和输出电压稳定。U1 为 5V 转 3.3V LDO，其型号为 ME6217C33M5G。因为显示模块上大部分电路需要 3.3V 供电，而 Type-C 接口输入电源基本上是 5V，所以需要稳压转换电路。

3) 电阻触摸屏控制电路

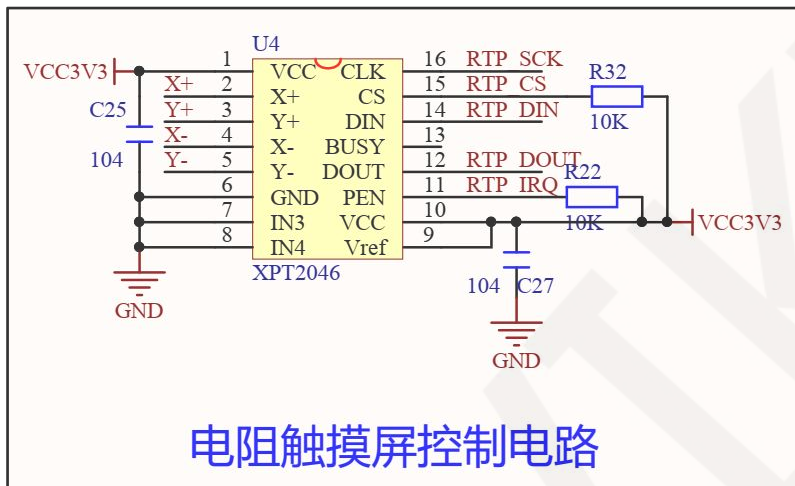


图 3.7 电阻触摸屏控制电路

此电路中，C25、C27 为旁路滤波电容，用来维持输入电压稳定。R22、R32 为上拉电阻，用来维持引脚默认状态为高电平。U4 为 XPT2046 控制 IC，此 IC 的作用就是通过 X+、X-、Y+、Y- 四个引脚获取电阻触摸屏触摸点的坐标电压值，然后经过 ADC 转换，将 ADC 值传输给 ESP32 主控。ESP32 主控再将 ADC 值转换为显示屏的像素坐标值。XPT2046 通过 SPI 总线和 ESP32 主控通信，因为其和显示屏共用 SPI 总线，所以得通过 CS 引脚控制使能状态。PEN 引脚为触摸中断引脚，发生触摸事件时，输入低电平。

4) USB 转串口以及一键下载电路

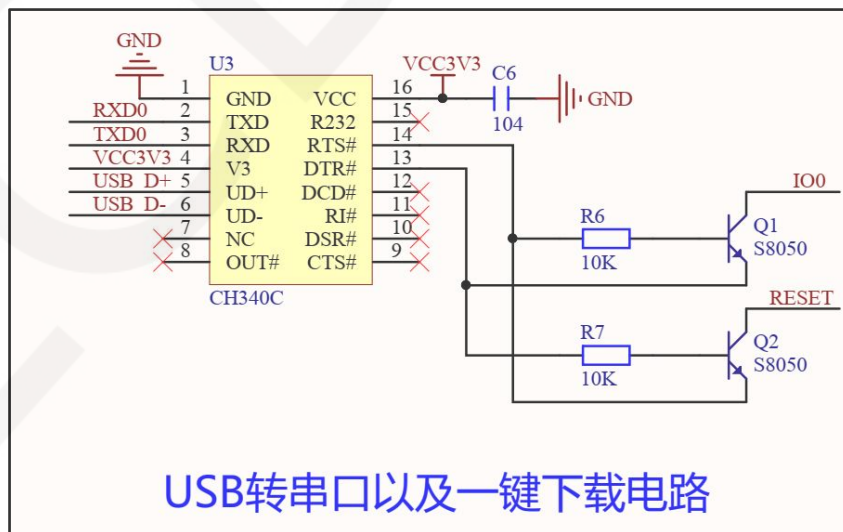


图 3.8 USB 转串口以及一键下载电路

此电路中，U3 为 CH340C USB 转串口 IC，其无需外接晶振，方便电路设计。C6 为旁路滤波电容，用来维持输入电压稳定。Q1、Q2 为 NPN 型三极管，R6、R7 为三极管基极限流电阻。此电路的作用就是实现 USB 转串口和一键下载功能。USB 信号通过 UD+ 和 UD- 引脚输入和输出，转换后通过 RXD、TXD 引脚传输到 ESP32 主控。一键下载电路原理：

- A、CH340C 的 RST 和 DTR 引脚默认都输出高电平，此时 Q1、Q2 三极管都没导通，ESP32 主控的 I00 引脚和复位引脚都被上拉至高电平；
- B、CH340C 的 RST 和 DTR 引脚都输出低电平，此时 Q1、Q2 三极管依然都没导通，ESP32 主控的 I00 引脚和复位引脚依然都被上拉至高电平；
- C、CH340C 的 RST 引脚保持电平不变，将 DTR 引脚输出高电平，此时 Q1 依然截止，Q2 导通，ESP32 主控的 I00 引脚依然被拉高，而复位引脚被拉低，ESP32 进入复位状态。
- D、CH340C 的 RST 引脚输出高电平，DTR 引脚输出低电平，此时 Q1 导通，Q2 截止，ESP32 主控的复位引脚因为所接电容充电，所以不会立即变为高电平，ESP32 还处于复位状态，而 I00 引脚立即被拉低，此时就会进入下载模式。

5) 音频功放电路

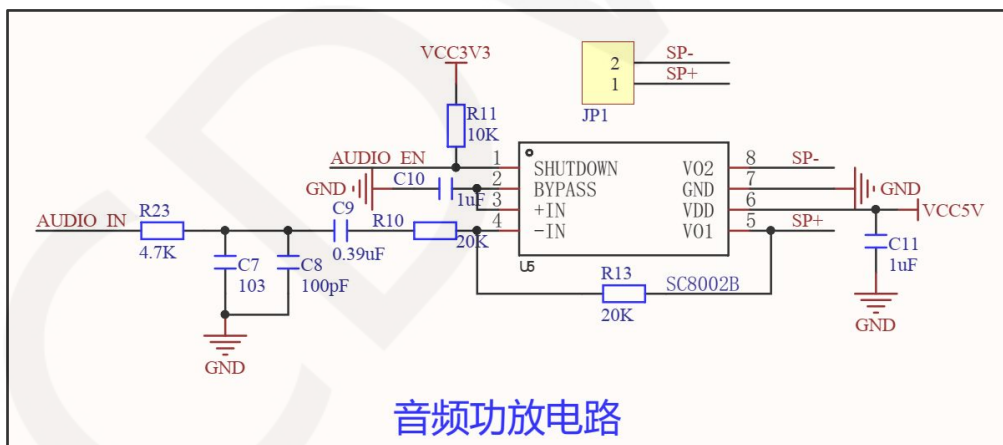


图 3.9 音频功放电路

此电路中 R23、C7、C8、C9 构成 RC 滤波电路，R10、R13 为运算放大器增益调节电阻，R13 阻值不变时，R10 阻值越小，外接扬声器音量越大。C10、C11 为输入耦合电容。R11 为上拉电阻。JP1 为喇叭/扬声器接口。U5 为 FM8002E 音频功放 IC。音频 DAC 信号 AUDIO_IN 输入后，经过 FM8002E 增益放大后，由 V01、V02 引脚输出到喇叭/扬声器。SHUTDOWN 为 FM8002E 使能引脚，低电平使能，默认状态为高电平。

6) ESP32-32E 主控电路

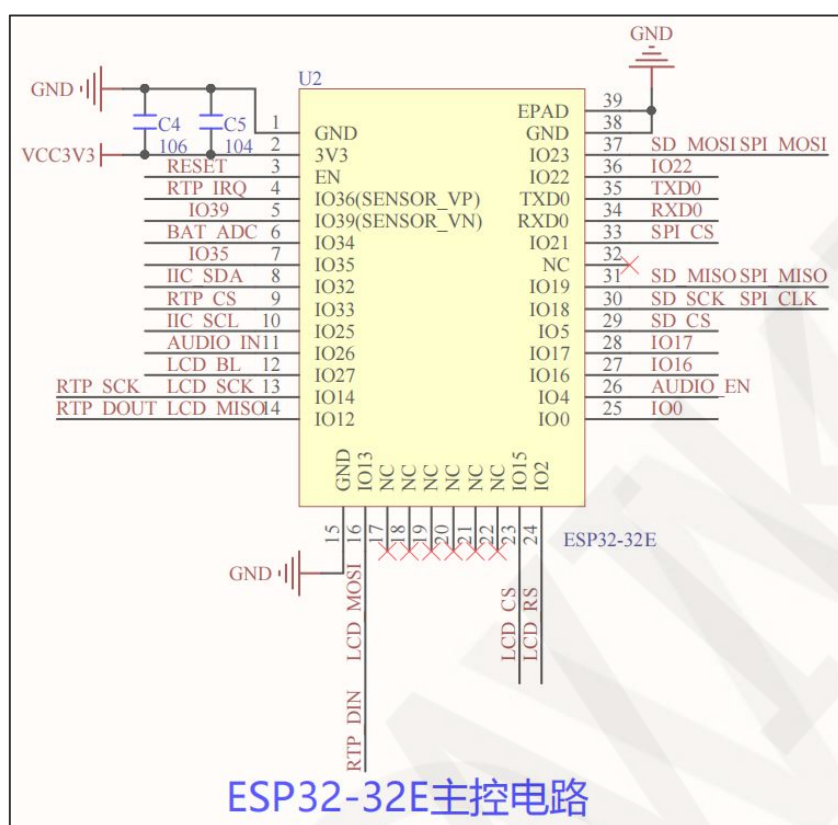


图 3.10 ESP32-32E 主控电路

此电路中，C4、C5 为旁路滤波电容，U2 为 ESP32-32E 模组，关于此模组内部电路详细介绍，请参考官方说明文档。

7) 按键复位电路

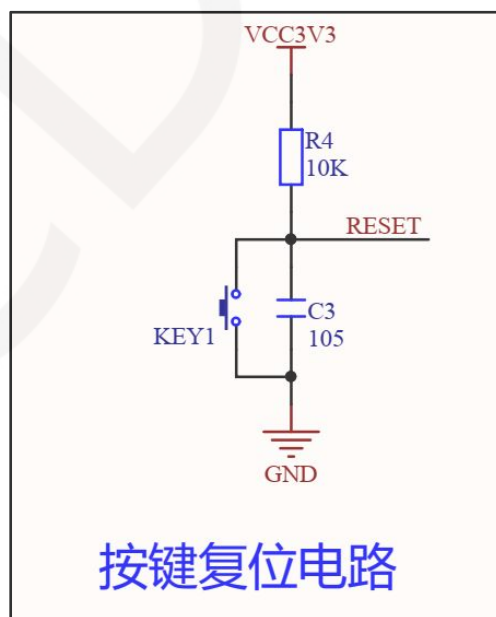


图 3.11 按键复位电路

此电路中，KEY1 为按键，R4 为上拉电阻，C3 为延时电容。复位原理：

A、上电后，C3 充电，此时 C3 相当于短路，RESET 引脚接地，ESP32 进入复位状态；

B、C3 充电完成，此时 C3 相当于开路，RESET 引脚被拉高，ESP32 复位结束，进入正常工作状态；

C、按下 KEY1，RESET 引脚接地，ESP32 进入复位状态，C3 通过 KEY1 放电；

D、松开 KEY1，C3 充电，此时 C3 相当于短路，RESET 引脚接地，ESP32 依然在复位状态，待 C3 充电完毕，RESET 引脚被拉高，ESP32 复位结束，进入正常工作状态；

如果复位不成功，可以适当增大 C3 的容值，延迟 RESET 引脚低电平时间。

8) 串口模块接口电路

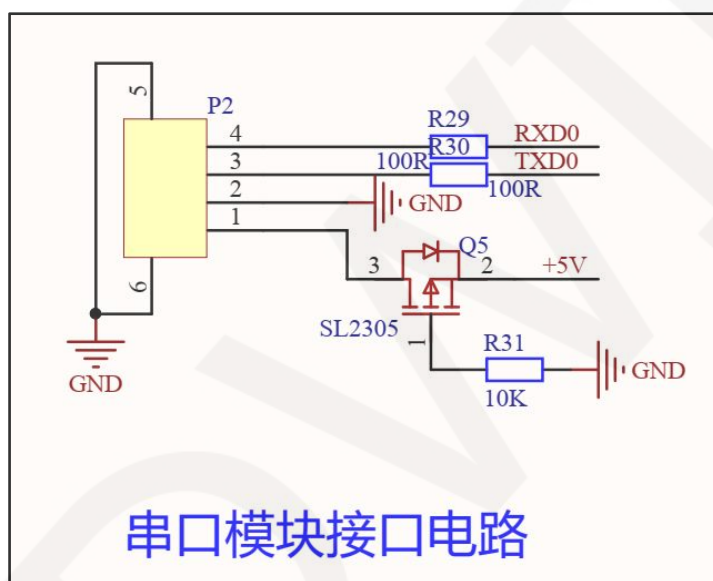


图 3.12 串口模块接口电路

此电路中，P2 为 4P 1.25mm 间距座子，R29、R30 为阻抗平衡电阻，Q5 为控制 5V 输入电源的场效应管。R31 为下拉电阻。RXD0 和 TXD0 接串口引脚，其他两个引脚供电。此接口和板载 USB 转串口模块接在同一个串口上。

9) 扩展 I/O 及外设接口电路

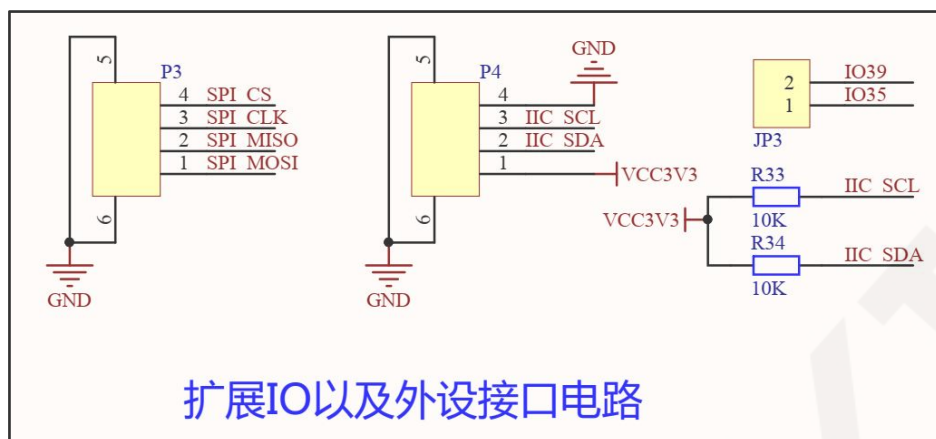


图 3.13 扩展 IO 及外设接口电路

此电路中，P3、P4 为 4P 1.25mm 间距的座子，JP3 为 2P 1.25mm 间距座子。R33、R34 为 I2C 引脚上拉电阻。SPI_CLK、SPI_MISO、SPI_MOSI 引脚和 MicroSD 卡 SPI 引脚共用。SPI_CS、IIC_SCL、IIC_SDA、IO35、IO39 引脚没有被板载设备使用，所以引出来接 SPI 和 IIC 设备，也可做普通 IO 使用。需要注意的地方：

A、IO35 和 IO39 只能做输入引脚；

B、IIC 引脚做普通 IO 使用时，最好将 R33 和 R34 上拉电阻去掉；

10) 电池充放电管理电路

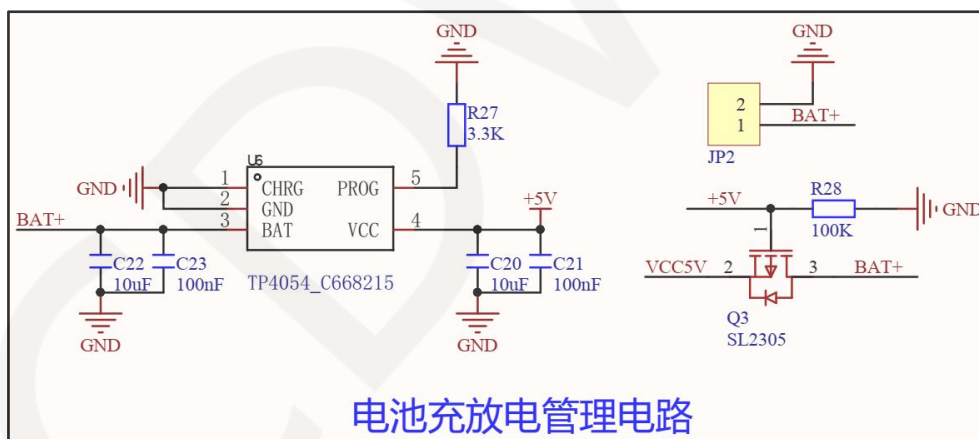


图 3.13 电池充放电管理电路

此电路中，C20、C21、C22、C23 为旁路滤波电容。U6 为 TP4054 电池充电管理 IC。R27 为电池充电电流调节电阻。JP2 为 2P 1.25mm 间距座子，接电池。Q3 为 P 沟道场效应管。R28 为 Q3 栅极下拉电阻。TP4054 通过 BAT 引脚给电池充电，R27 阻值越小，充电电流越大，最大为 500mA。Q3 和 R28 共同构成了电池放电电路，当没有通过 Type-C 接口供电时，+5V 电压为 0，此时 Q3 栅极被下拉到低电平，漏极和源极导通，电池给整个显示模块供电。当通过 Type-C 接口供电时，+5V 电压为 5V，此时 Q3 栅极为 5V

高电平，漏极和源极截止，电池供电中断。

11) 48P 液晶屏排线焊接接口

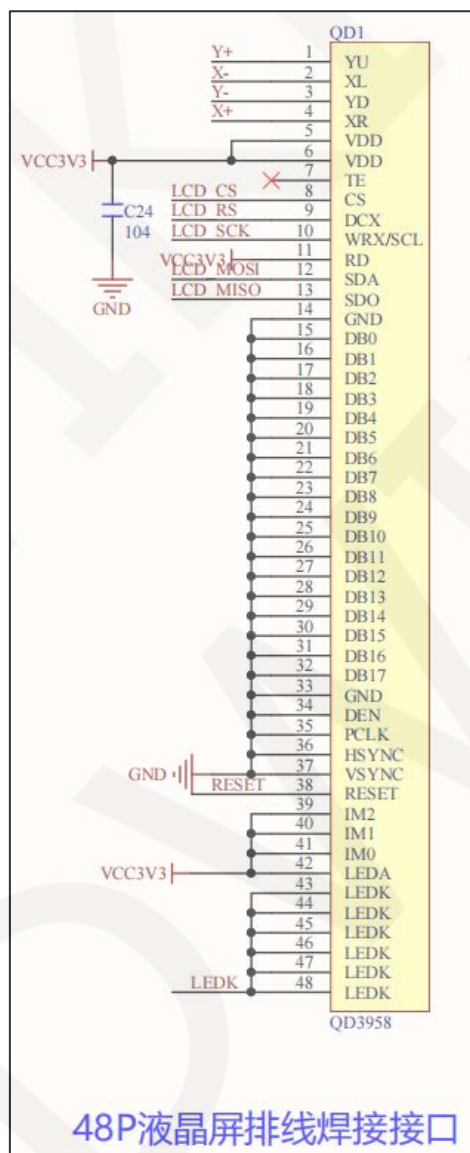


图 3.14 48P 液晶屏排线焊接接口

此电路中，C24 为旁路滤波电容，QD1 为 48P 0.8mm 间距液晶屏焊接接口。QD1 上有电阻触摸屏信号引脚，液晶屏电压引脚、SPI 通信引脚、控制引脚以及背光电路引脚。ESP32 就是通过这些引脚来控制液晶屏和触摸屏。

12) 下载按键电路

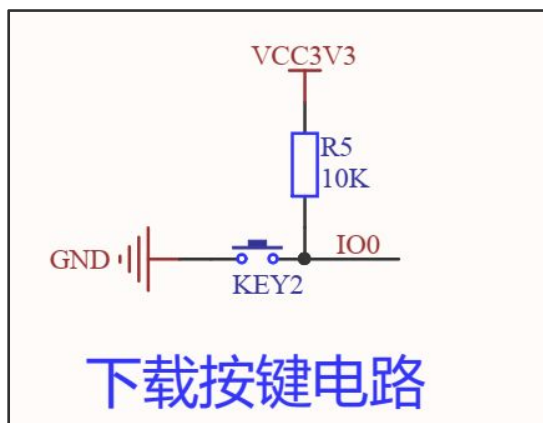


图 3.15 下载按键电路

此电路中，KEY2 为按键，R5 为上拉电阻。IO0 默认状态为高电平，KEY2 按下，则为低电平。按住 KEY2，上电或者复位，ESP32 会进入下载模式。其他情况下，KEY2 可以做普通按键使用。

13) 电池电量检测电路

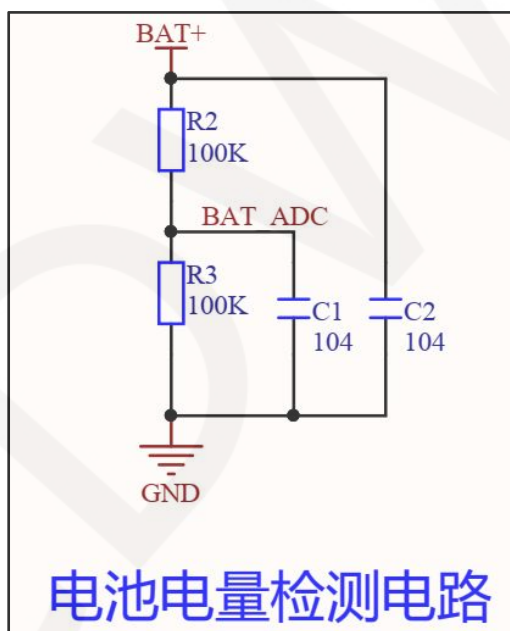


图 3.15 电池电量检测电路

此电路中，R2、R3 为分压电阻，C1、C2 为旁路滤波电容。电池电压 BAT+ 信号输入经过分压电阻。BAT_ADC 为 R3 两端电压值，通过输入引脚传输给 ESP32 主控，然后经过 ADC 转换，最终获取到电池电压值。之所以使用分压电阻，是因为 ESP32 ADC 转换的最大值为 3.3V，而电池饱和电压为 4.2V，超出范围。获取的电压值乘以 2 就是实际的电池电压值。

14) 液晶屏背光控制电路

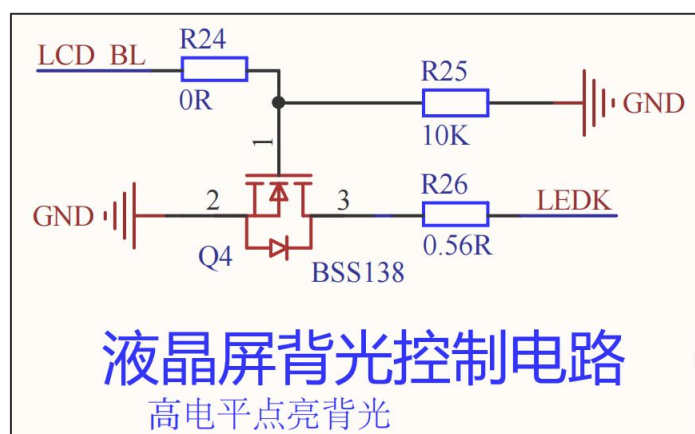


图 3.16 液晶屏背光控制电路

此电路中，R24 为调试电阻，暂时保留。Q4 为 N 沟道场效应管，R25 为 Q4 栅极下拉电阻，R26 为背光限流电阻。液晶屏背光 LED 灯为并联状态，正极接 3.3V，负极接 Q4 的漏极，当控制引脚 LCD_BL 输出高电平时，Q4 的漏极和源极导通，此时液晶屏背光负极接地，背光 LED 灯导通发光；当控制引脚 LCD_BL 输出低电平时，Q4 的漏极和源极截止，此时液晶屏背光负极悬空，背光 LED 灯不导通。默认状态下液晶屏背光灯是不亮的。减小 R26 阻值，可以增大背光灯最大亮度。此外，LCD_BL 引脚可以输入 PWM 信号调节液晶屏背光灯亮度。

15) RGB 三色灯控制电路

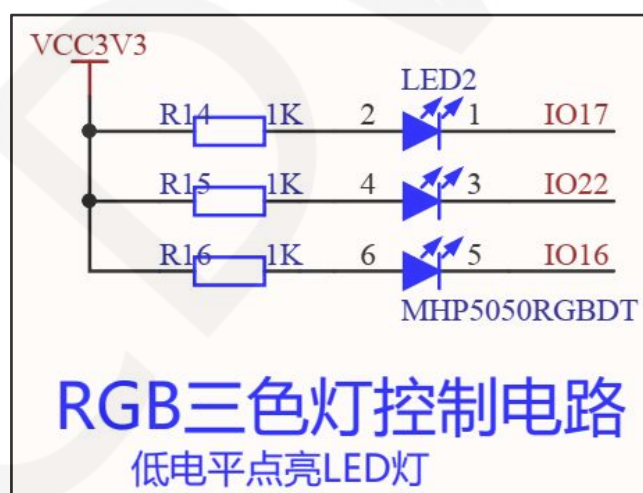


图 3.17 液晶屏背光控制电路

此电路中，LED2 为 RGB 三色灯，R14~R16 为三色灯限流电阻。LED2 包含红、绿、蓝三色 LED 灯，为共阳极连接，IO16、IO17、IO22 为三个控制引脚，低电平点亮 LED 灯，高电平熄灭 LED 灯。

16) MicroSD 卡槽接口电路

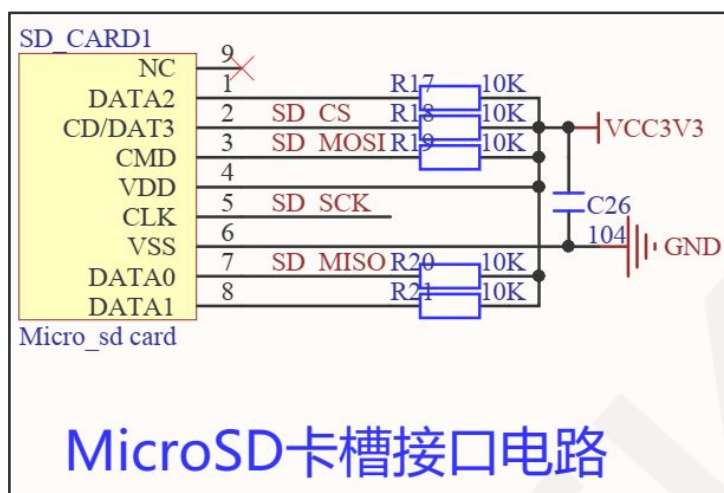


图 3.18 MicroSD 卡槽接口电路

此电路中，SD_CARD1 为 MicroSD 卡槽。R17~R21 为各个引脚上拉电阻。C26 为旁路滤波电容。此接口电路采用 SPI 通信方式。支持高速存储 MicroSD 卡。需要注意的是此接口和 SPI 外设接口共用 SPI 总线。

3.3. 显示模块使用注意事项

- 1) 显示模块接电池充电，外接喇叭播放音频，显示屏也工作，此时总电流可能超过 500mA。这种情况下需要注意 Type-C 线支持的最大电流以及供电接口支持的最大电流，避免供电不足。
- 2) 在使用过程中，不要用手接触 LD0 稳压管和电池充电管理 IC，避免被高温烫伤。
- 3) 接引出来的 IO 口时，注意 IO 使用情况，避免接错导致和程序代码定义不匹配。
- 4) 安全合理的使用该产品。