



esp-dev-kits

2016 - 2022 乐鑫信息科技（上海）股份有限公司

2023 年 06 月 13 日

ESP32-C3 系列开发板

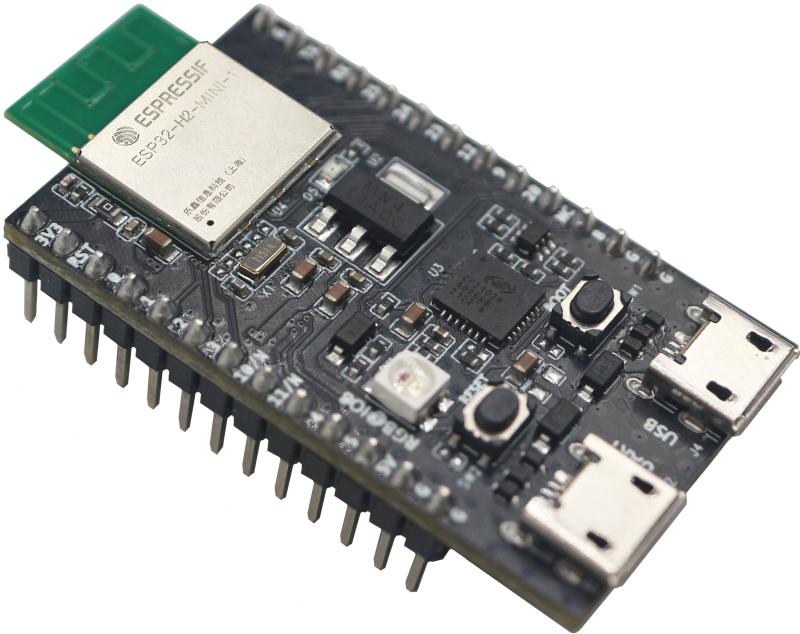
| | |
|---------------------|------------|
| 1 开发板 | 3 |
| 1.1 相关文档 | 8 |
| Bibliography | 189 |

[English]

该仓库为 [esp-dev-kits](#) 的文档。

CHAPTER 1

开发板

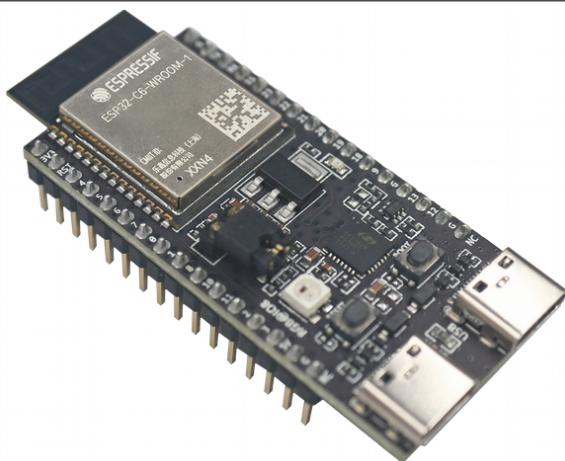
| ESP32-H2 系列开发板 | |
|--|--|
|  The image shows the ESP32-H2-DevKitM-1 development board. It is a black PCB with a central ESP32-H2 chip, a green WiFi/BT module labeled "ESPRESSIF ESP32-H2-MINI-1", and various connectors and components. A USB port is visible on the right side. | |

ESP32-C3 系列开发板

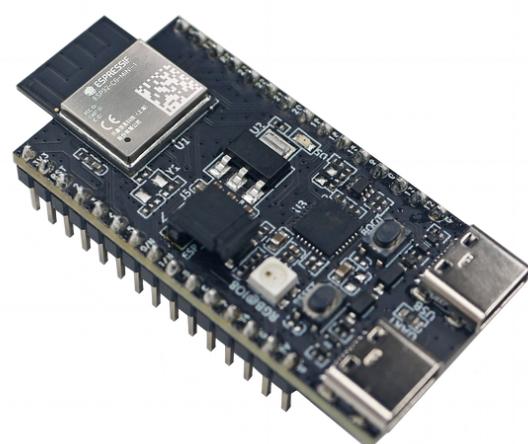


ESP32-C3-LCDkit

ESP32-C6 系列开发板

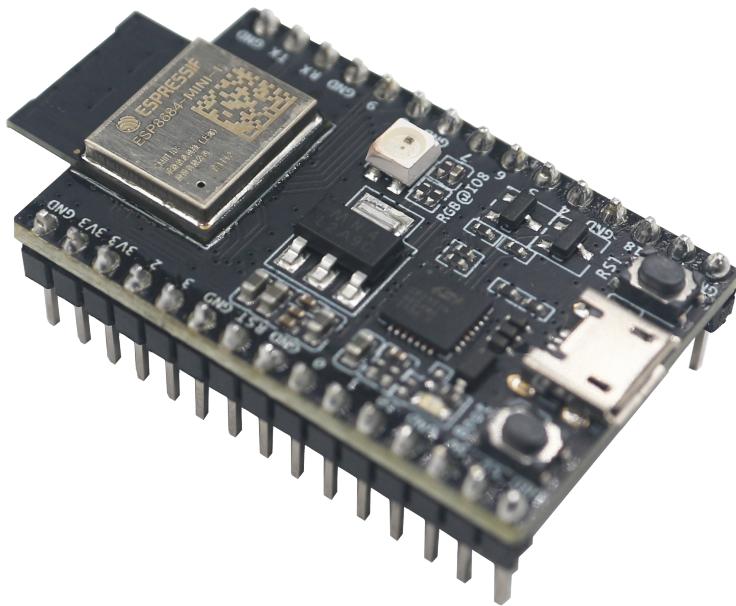


ESP32-C6-DevKitC-1

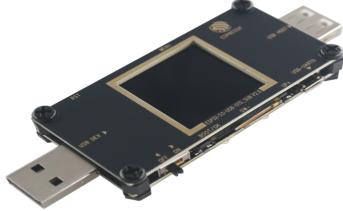


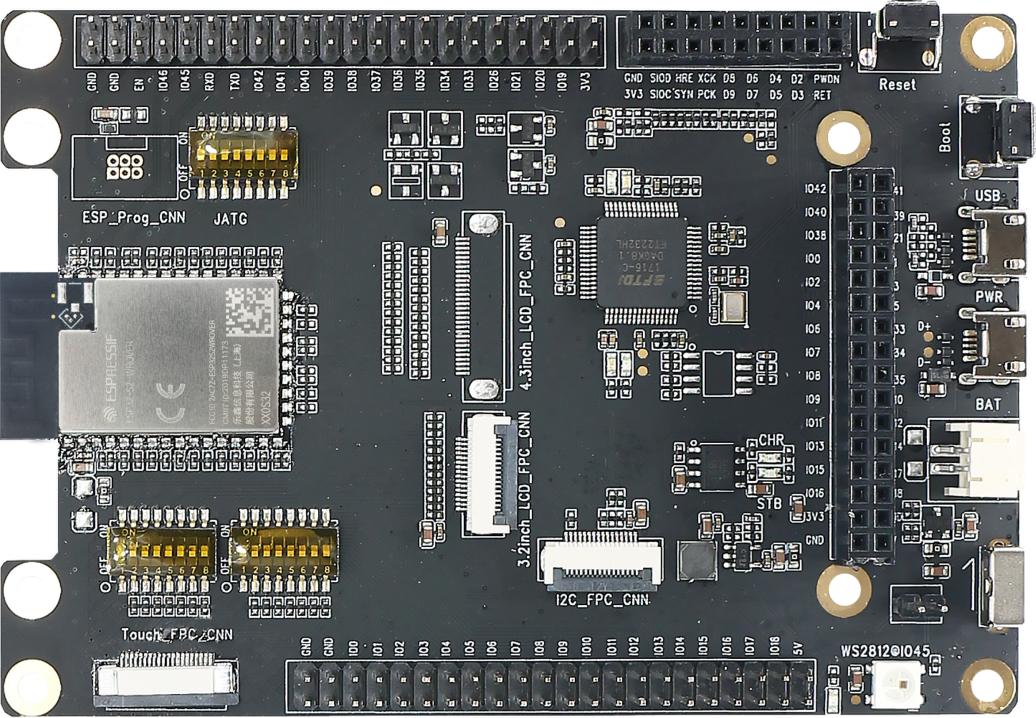
ESP32-C6-DevKitM-1

ESP8684 系列开发板

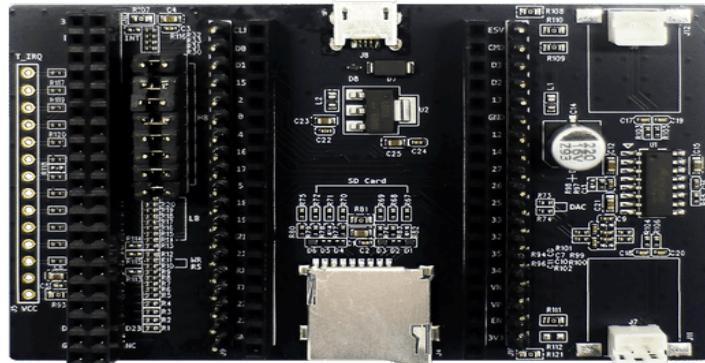


ESP8684-DevKitM-1

| ESP32-S3 系列开发板 | |
|---|--|
|  |  |

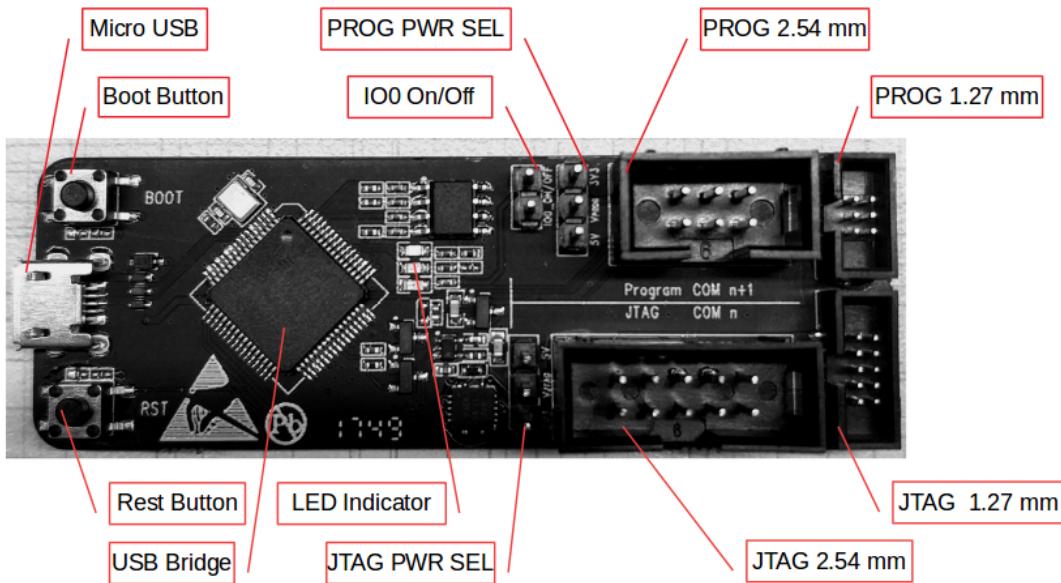
| ESP32-S2 系列开发板 | |
|--|--|
|  | |

ESP32 系列开发板



ESP32-LCDKit

其他开发板



ESP-Prog

| 寿命终止开发板 | |
|-----------------------|-------------------------|
| | |
| ESP32-S2-HMI-DevKit-1 | ESP32-Sense-Kit |
| | |
| ESP32-MeshKit-Sense | ESP32-S2-Touch-DevKit-1 |

1.1 相关文档

- [ESP-IDF 详细安装步骤](#)
- [ESP-IDF 编程指南](#)
- [ESP 产品选型工具](#)

1.1.1 ESP32-C3-LCDkit

[English]

ESP32-C3-LCDkit 是一款物种保护系列开发板，用于评估和验证 ESP32-C3 屏幕应用，由主板和子板构成。

ESP32-C3-LCDkit

[English]

本指南将帮助您快速上手 ESP32-C3-LCDkit，并提供该款开发板的详细信息。

本指南包括如下内容：

- **开发板概述**：简要介绍了开发板的软件和硬件。
- **应用程序开发**：介绍了应用程序开发过程中的软硬件设置。
- **硬件参考**：详细介绍了开发板的硬件。
- **硬件版本**：暂无历史版本。
- **样品获取**：如何获取样品。
- **相关文档**：列出了相关文档的链接。

开发板概述

ESP32-C3-LCDkit 是一款基于 ESP32-C3 芯片和 SPI 接口显示屏的评估开发板，不仅通过旋转编码器开关实现屏幕交互，还具有音频播放和红外无线控制功能。由于 ESP32-C3 具有成本低、功耗低、性能强的特点，能够满足用户基本的 GUI 交互需求，其在小尺寸屏幕的应用场景中占据优势。

特性列表

该开发板具有以下特性：

- **嵌入式模组**：板载 ESP32-C3-MINI-1 模组，内置 4 MB flash 以及 400 KB SRAM
- **屏幕**：可搭配不同屏幕子板使用，支持 I²C 和 SPI 接口屏幕，请查看 [LCD 子板](#) 了解更多信息
- **旋转编码器开关**：支持按键开关以及进行 360° 的旋转，用于实现对屏幕 GUI 的控制操作
- **红外模块**：支持红外发射器和红外接收器，用于实现红外无线控制
- **音频**：板载音频功放和扬声器，支持音频播放
- **USB**：支持 USB Type-C 接口下载调试



图 1: 搭载 ESP32-C3-MINI-1 模组的 ESP32-C3-LCDkit

功能框图

ESP32-C3-LCDkit 的主要组件和连接方式如下图所示。

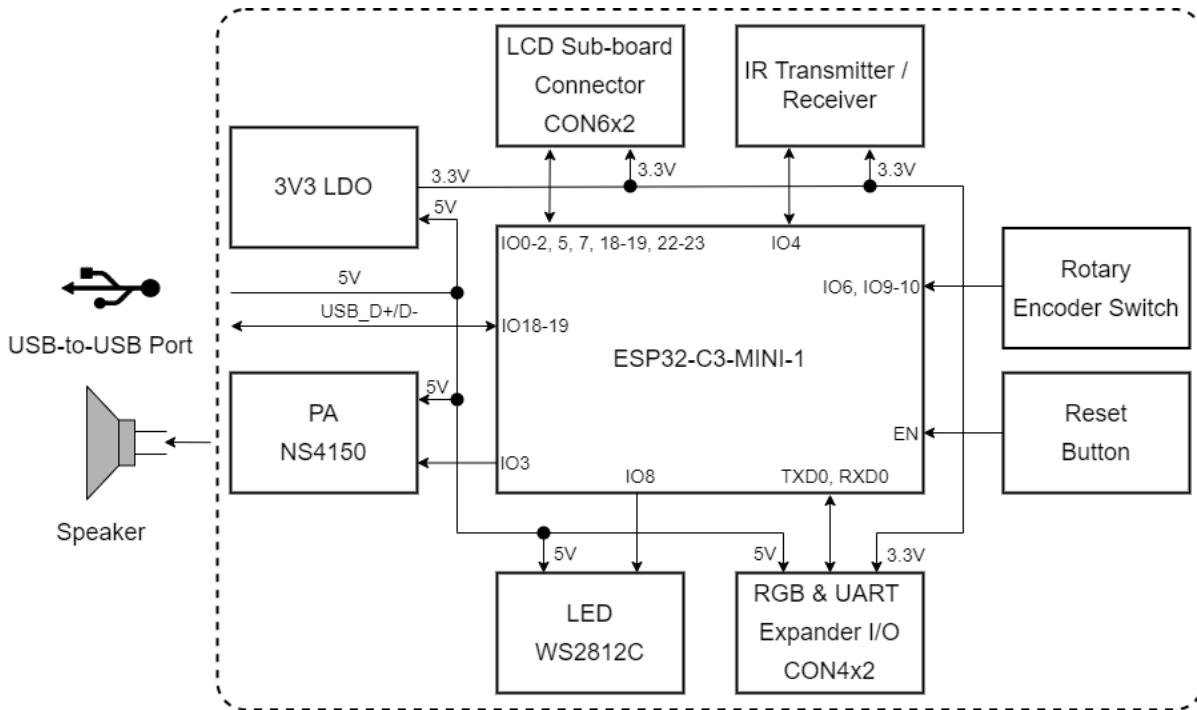


图 2: ESP32-C3-LCDkit 功能框图 (点击放大)

组件介绍

ESP32-C3-LCDkit 是一款物种保护系列开发板，由主板和子板组成。

主板

ESP32-C3-LCDkit_MB 主板是整个套件的核心，该主板集成了 **ESP32-C3-MINI-1** 模组，并提供与 LCD 子板连接的端口。

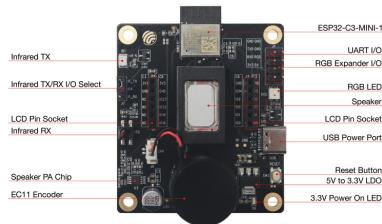


图 3: ESP32-C3-LCDkit - 正面 (点击放大)

以下按照顺时针顺序依次介绍开发板上的主要组件。

| 主要组件 | 介绍 |
|--------------------|--|
| ESP32-C3-MINI-1 模组 | ESP32-C3-MINI-1 模组是一款通用型 Wi-Fi + 低功耗蓝牙 MCU 模组，搭载 ESP32-C3 系列芯片，内置 4 MB flash 以及 400 KB SRAM。 |
| UART & RGB 扩展 I/O | 通过 2.54 mm 间距的排针连接系统电源管脚以及 UART 和 RGB 的数据引脚。 |
| RGB LED | 具有 RGB 三色显示功能的 LED，可供用户配置用来做状态行为指示。 |
| 扬声器 | 可通过音频功率放大器的支持，实现扬声器播放功能。 |
| LCD 屏幕连接器 | 通过 2.54 mm 间距的排母连接 1.28 英寸 LCD 子板。 |
| USB 电源端口 | 为整个系统提供电源。建议使用至少 5V/2A 电源适配器供电，保证供电稳定。该端口用于 PC 端与 ESP32-C3-MINI-1 模组的 USB 通信。 |
| Reset 按键 | 单独按下此按键会重置系统。 |
| 5V-to-3.3V LDO | 低压差线型稳压器 (LDO)。 |
| 3.3 V 电源指示灯 | 用于指示系统供电的状态。 |
| EC11 旋转编码器开关 | 同时具有 360° 旋转编码器和按键开关的功能，用于实现对屏幕 GUI 的控制操作。 |
| 音频功率放大芯片 | 实现扬声器播放功能。 |
| 红外接收器 | 用于接收外界发送的红外信号。 |
| 红外功能选择端口 | 通过 2.54 mm 间距的排针和跳线帽选择使用红外发射或者接收的功能。 |
| 红外发射器 | 用于向外界发送红外信号。 |

LCD 子板

ESP32-C3-LCDkit_DB 子板支持 1.28 英寸、SPI 接口、分辨率为 240x240 的 LCD 屏，该屏使用的驱动芯片型号为 GC9A01。

软件支持

ESP32-C3-LCDkit 的开发框架为 **ESP-IDF**。ESP-IDF 是基于 FreeRTOS 的乐鑫 SoC 开发框架，具有众多组件，包括 LCD、ADC、RMT、SPI 等。开发板应用示例存放在 `Examples` 中，在示例目录下输入 `idf.py menuconfig` 可以配置工程选项。

应用程序开发

本节介绍硬件和软件的设置方法，以及烧录固件至开发板以开发应用程序的说明。



图 4: ESP32-C3-LCDkit_DB - 正面 (点击放大)



图 5: ESP32-C3-LCDkit_DB - 反面 (点击放大)

必备硬件

- 1 x ESP32-C3-LCDkit_MB
- 1 x LCD 子板
- 1 x USB 2.0 数据线（标准 A 型转 Type-C 型）
- 1 x 电脑（Windows、Linux 或 macOS）

注解：请确保使用适当的 USB 数据线。部分数据线仅可用于充电，无法用于数据传输和程序烧录。

硬件设置

准备开发板，加载第一个示例应用程序：

1. 连接 LCD 子板至 **LCD 屏幕连接器**。
2. 插入 USB 数据线，分别连接 PC 与开发板的 USB 端口。
3. LCD 屏幕亮起，可以使用旋转编码器开关控制 GUI。

硬件设置完成，接下来可以进行软件设置。

软件设置

了解如何快速设置开发环境，请前往 [快速入门 > 安装](#)。

了解开发应用程序的更多软件信息，请查看[软件支持](#)。

硬件参考

本节提供关于开发板硬件的更多信息。

GPIO 分配列表

下表为 ESP32-C3-MINI-1 模组管脚的 GPIO 分配列表，用于控制开发板的特定组件或功能。

表 1: ESP32-C3-MINI-1 GPIO 分配

| 管脚 | 管脚名称 | 功能 |
|----|------|----|
| 1 | GND | 接地 |
| 2 | GND | 接地 |

下页继续

表 1 - 续上页

| 管脚 | 管脚名称 | 功能 |
|-------|------|-------------|
| 3 | 3V3 | 供电 |
| 4 | NC | 未连接 |
| 5 | IO2 | LCD_D/C |
| 6 | IO3 | AUDIO_PA |
| 7 | NC | 未连接 |
| 8 | EN | Reset |
| 9 | NC | 未连接 |
| 10 | NC | 未连接 |
| 11 | GND | 接地 |
| 12 | IO0 | LCD_SDA |
| 13 | IO1 | LCD_SCL |
| 14 | GND | 接地 |
| 15 | NC | 未连接 |
| 16 | IO10 | ENCODER_A |
| 17 | NC | 未连接 |
| 18 | IO4 | IR_RX/IR_TX |
| 19 | IO5 | LCD_BL_CTRL |
| 20 | IO6 | ENCODER_A |
| 21 | IO7 | LCD_CS |
| 22 | IO8 | RGB_LED |
| 23 | IO9 | ENCODER_SW |
| 24 | NC | 未连接 |
| 25 | NC | 未连接 |
| 26 | IO18 | USB_DN |
| 27 | IO19 | USB_DP |
| 28 | NC | 未连接 |
| 29 | NC | 未连接 |
| 30 | RXD0 | 预留 |
| 31 | TXD0 | 预留 |
| 32-35 | NC | 未连接 |
| 36-53 | GND | 接地 |

供电说明

开发板通过 USB 端口供电：

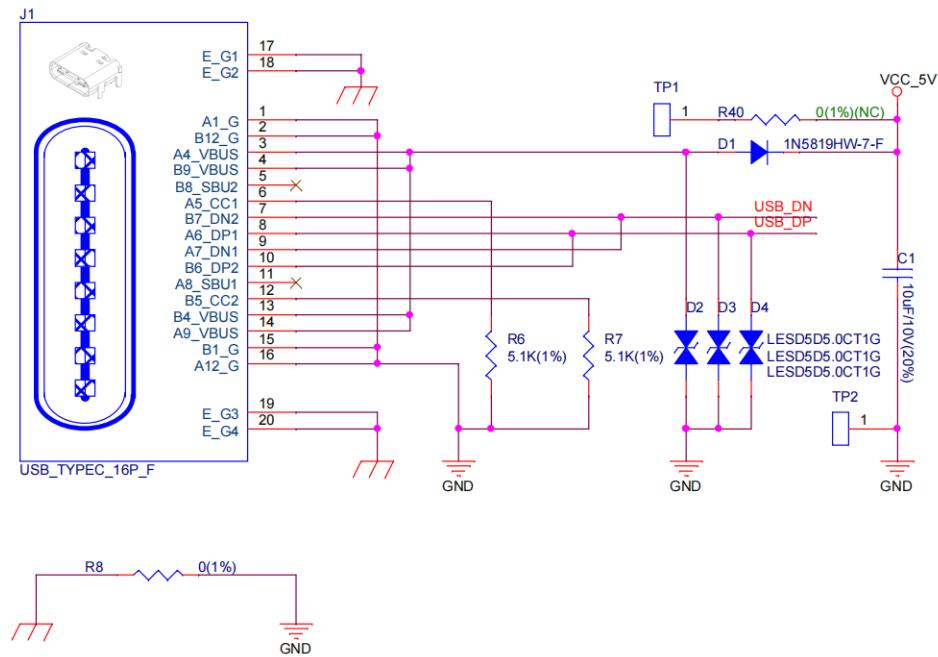


图 6: ESP32-C3-LCDkit - USB 电源供电

输出系统电源供电：

红外发射/接收选择

由于红外发射模块和红外接收模块共用芯片上同一根信号线，需要通过跳线帽短接 红外功能选择端口中特定的引脚，来选择使用红外发射或红外接收的功能：

C6 模组兼容性设计

ESP32-C3-LCDkit 开发板默认使用 ESP32-C3-MINI-1 模组，此时主板上连接了电阻 R2、R4、R5 和 R35：

开发板在设计上还兼容了 ESP32-C6-MINI-1 模组，此时需要移除电阻 R2、R4、R5 和 R35，并连接上电阻 R13 和 R14。

3V3 LDO

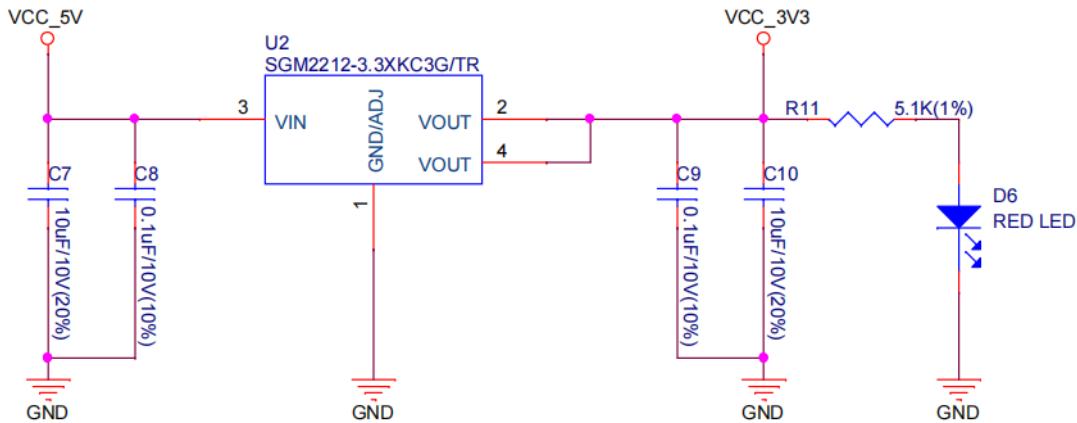


图 7: ESP32-C3-LCDkit - 系统电源供电

IR Receiver

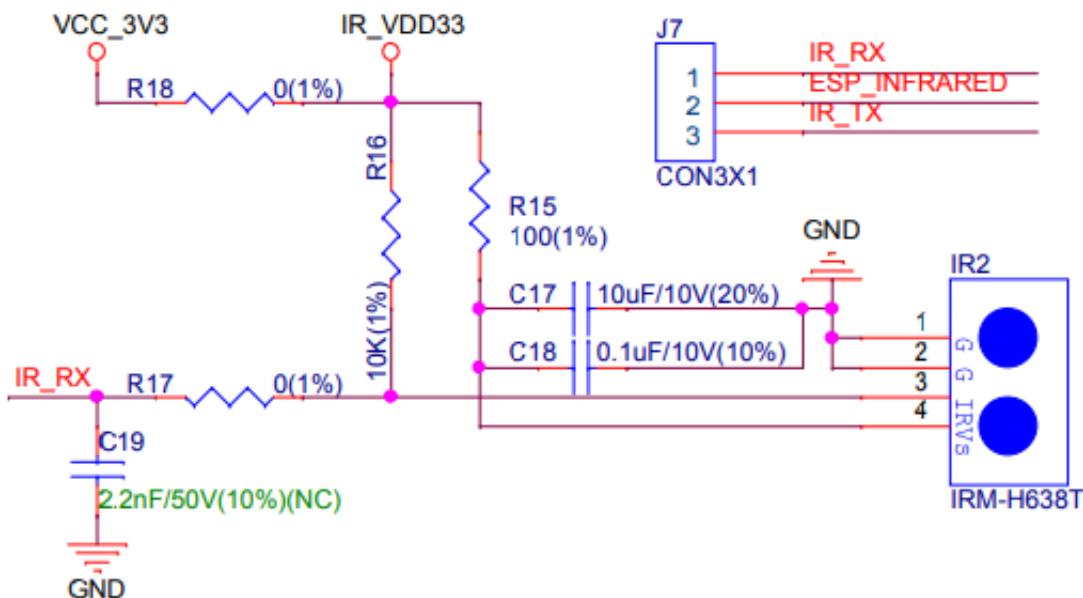


图 8: ESP32-C3-LCDkit - 红外接收模块

IR Transmitter

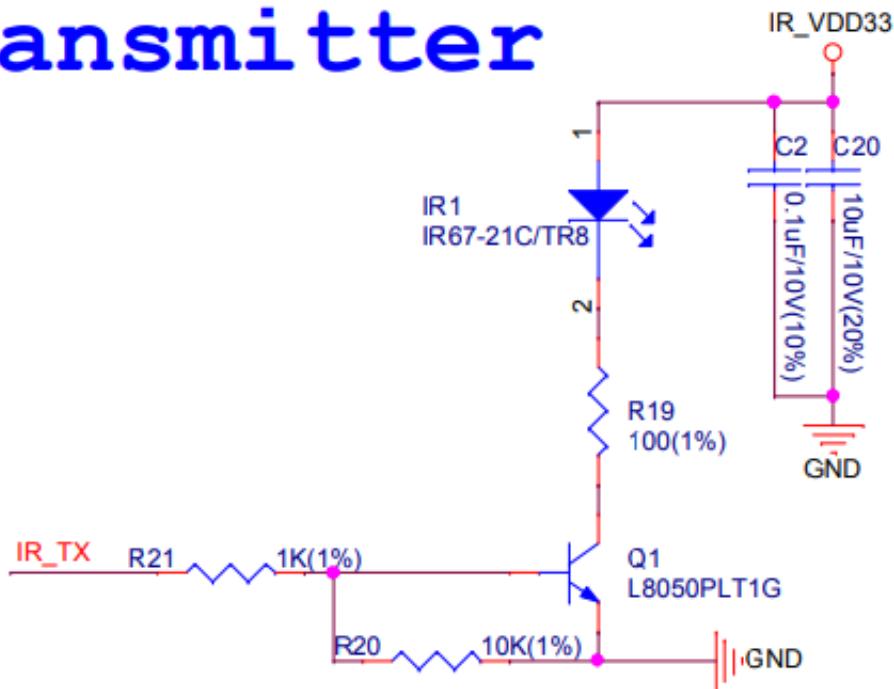


图 9: ESP32-C3-LCDkit - 红外发射模块

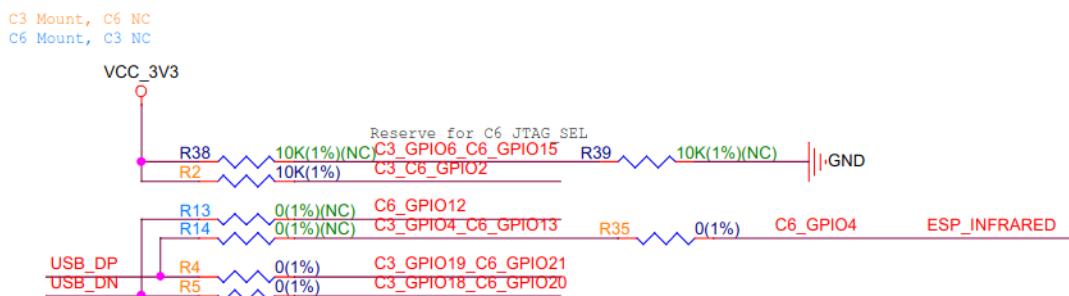


图 10: ESP32-C3-LCDkit - C3/C6 电阻设置

硬件设置选项

自动下载

开发板上电后，手动按下旋转编码器开关和 Reset 键，然后先松开 Reset，再松开编码器，即可使 ESP 开发板进入下载模式。

硬件版本

- ESP32-C3-LCD-Ev-Board：该版本为工程样品，乐鑫有限维护。有关此版本开发板的历史文档，请参见[ESP32-C3-LCD-Ev-Board 用户指南](#)。

样品获取

ESP32-C3 具有成本优势和行业领先的低功耗性能，适用于构建由 SPI 接口驱动的旋钮屏或小尺寸显示屏。有采购需求请邮件联系 sales@espressif.com。

相关文档

- [ESP32-C3 技术规格书](#)
- [ESP32-C3-MINI-1 技术规格书](#)
- [乐鑫产品选型工具](#)
- [ESP32-C3-LCDkit_MB 原理图](#)
- [ESP32-C3-LCDkit_MB PCB 布局图](#)
- [ESP32-C3-LCDkit_DB 原理图](#)
- [ESP32-C3-LCDkit_DB PCB 布局图](#)
- [1.28_TFT_240x240_SPI 屏](#)
- [红外发射器 \(IR67-21CTR8\)](#)
- [红外接收器 \(IRM-H638TTR2\)](#)
- [音频功放 \(NS4150\)](#)
- [RGB LED \(WS2812B\)](#)
- [2415 音腔喇叭](#)

有关本开发板的更多设计文档，请联系我们的商务部门 sales@espressif.com。

ESP32-C3-LCD-Ev-Board

[English]

本指南将帮助您快速上手 ESP32-C3-LCD-Ev-Board，并提供该款开发板的详细信息。

本指南包括如下内容：

- **开发板概述**：简要介绍了开发板的软件和硬件。
- **应用程序开发**：介绍了应用程序开发过程中的软硬件设置。
- **硬件参考**：详细介绍了开发板的硬件。
- **硬件版本**：暂无历史版本。
- **样品获取**：如何获取样品。
- **相关文档**：列出了相关文档的链接。

开发板概述

ESP32-C3-LCD-Ev-Board 是一款基于 ESP32-C3 芯片和 SPI 接口显示屏的评估开发板，同时它还使用了旋转编码器开关，实现屏幕交互的功能。由于 ESP32-C3 具有成本低、功耗低、性能强的特点，能够满足用户基本的 GUI 交互需求，其在小尺寸屏幕的应用场景中占据优势。

特性列表

该开发板具有以下特性：

- **嵌入式模组**：板载 ESP32-C3-MINI-1 模组，内置 4 MB flash 以及 400 KB SRAM
- **屏幕**：可搭配不同屏幕子板使用，支持 I²C 和 SPI 接口屏幕，请查看 [LCD 子板](#) 了解更多信息
- **旋转编码器开关**：支持按键开关以及进行 360° 的旋转，用于实现对屏幕 GUI 的控制操作
- **USB**：支持 USB Type-C 接口下载调试

功能框图

ESP32-C3-LCD-Ev-Board 的主要组件和连接方式如下图所示。



图 11: 搭载 ESP32-C3-MINI-1 模组的 ESP32-C3-LCD-Ev-Board

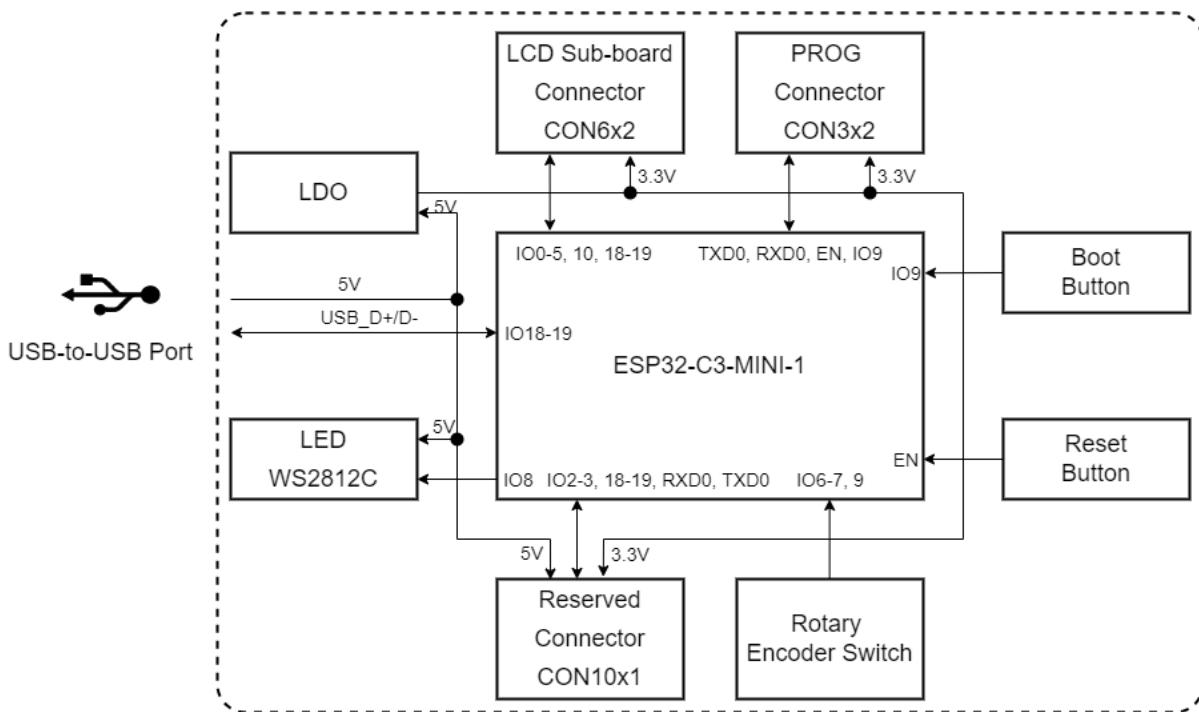


图 12: ESP32-C3-LCD-Ev-Board 功能框图 (点击放大)

组件介绍

ESP32-C3-LCD-Ev-Board 开发板由主板和子板组成。

主板

ESP32-C3-LCD-Ev-Board_MB 主板是整个套件的核心，该主板集成了 **ESP32-C3-MINI-1** 模组，并提供与 LCD 子板连接的端口。

以下按照逆时针顺序依次介绍开发板上的主要组件。

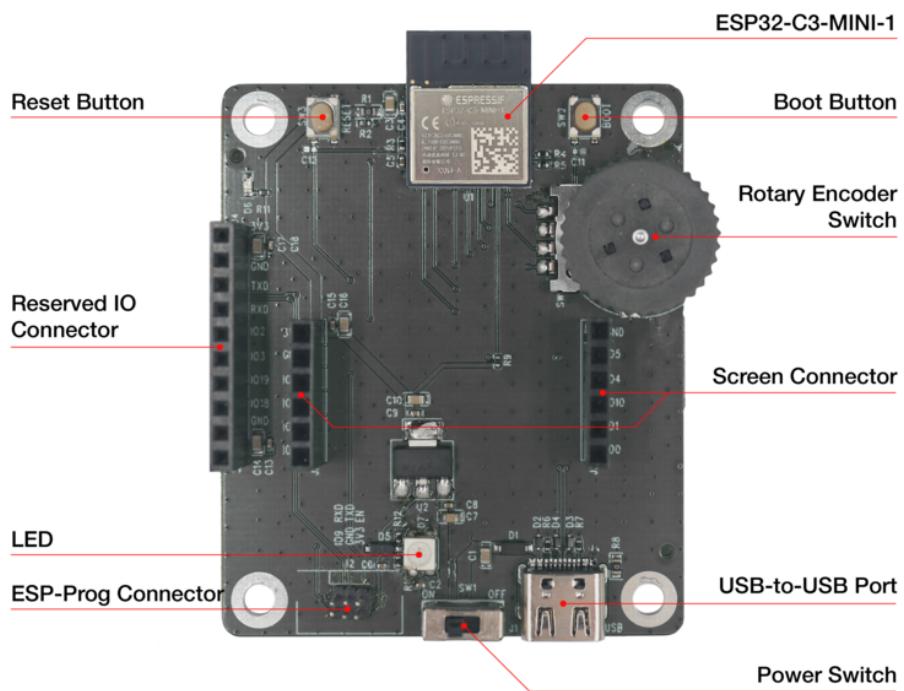


图 13: ESP32-C3-LCD-Ev-Board - 正面 (点击放大)

| 主要组件 | 介绍 |
|--------------------|--|
| ESP32-C3-MINI-1 模组 | ESP32-C3-MINI-1 模组是一款通用型 Wi-Fi + 低功耗蓝牙 MCU 模组，搭载 ESP32-C3 系列芯片，内置 4 MB flash 以及 400 KB SRAM。 |
| Reset 按键 | 单独按下此按键会重置系统。 |
| 预留 IO 连接器 | 通过 2.54 mm 间距的连接器连接系统电源管脚以及预留的部分模组管脚。 |
| LED | 具有 RGB 三色显示功能，可供用户配置用来做状态行为指示。 |
| ESP-Prog 连接器 | 通过 1.27 mm 间距的连接器可以连接 ESP-Prog 的 Program 接口，用于固件下载与调试。 |
| 电源开关 | 电源拨动开/关：向 ON 拨动开启开发板电源，向 OFF 拨动关闭开发板电源。 |
| USB-to-USB 端口 | 为整个系统提供电源。建议使用至少 5V/2A 电源适配器供电，保证供电稳定。该端口用于 PC 端与 ESP32-C3-MINI-1 模组的 USB 通信。 |
| 屏幕连接器 | 通过 2.54 mm 间距的连接器连接 1.28 英寸 LCD 子板。 |
| 旋转编码器开关 | 同时具有 360° 旋转编码器和按键开关的功能，用于实现对屏幕 GUI 的控制操作。 |
| Boot 按键 | 长按 Boot 键时，再按 Reset 键可启动固件上传模式，然后便可通过串口或 USB 上传固件。 |

LCD 子板

ESP32-C3-LCD-Ev-Board_DB 子板支持 1.28 英寸、SPI 接口、分辨率为 240x240 的 LCD 屏，该屏使用的驱动芯片型号为 GC9A01。

软件支持

ESP32-C3-LCD-Ev-Board 的开发框架为 [ESP-IDF](#)。ESP-IDF 是基于 FreeRTOS 的乐鑫 SoC 开发框架，具有众多组件，包括 LCD、ADC、RMT、SPI 等。



图 14: ESP32-C3-LCD-Ev-Board_DB - 正面 (点击放大)

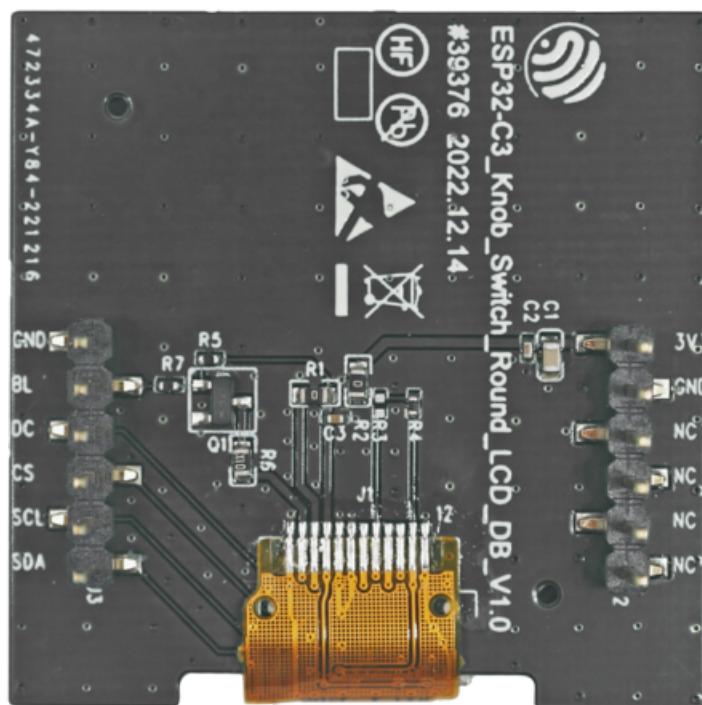


图 15: ESP32-C3-LCD-Ev-Board_DB - 反面 (点击放大)

应用程序开发

本节介绍硬件和软件的设置方法，以及烧录固件至开发板以开发应用程序的说明。

必备硬件

- 1 x ESP32-C3-LCD-Ev-Board_MB
- 1 x LCD 子板
- 1 x USB 2.0 数据线（标准 A 型转 Type-C 型）
- 1 x 电脑（Windows、Linux 或 macOS）

注解：请确保使用适当的 USB 数据线。部分数据线仅可用于充电，无法用于数据传输和程序烧录。

硬件设置

准备开发板，加载第一个示例应用程序：

1. 连接 LCD 子板至 **LCD 子板连接器** 端口。
2. 插入 USB 数据线，分别连接 PC 与开发板的 USB 端口。
3. LCD 屏幕亮起，可以使用拨轮编码器开关控制 GUI。

硬件设置完成，接下来可以进行软件设置。

软件设置

了解如何快速设置开发环境，请前往 [快速入门 > 安装](#)。

了解开发应用程序的更多软件信息，请查看[软件支持](#)。

硬件参考

本节提供关于开发板硬件的更多信息。

GPIO 分配列表

下表为 ESP32-C3-MINI-1 模组管脚的 GPIO 分配列表，用于控制开发板的特定组件或功能。

表 2: ESP32-C3-MINI-1 GPIO 分配

| 管脚 | 管脚名称 | 功能 |
|-------|------|------------------|
| 1 | GND | 接地 |
| 2 | GND | 接地 |
| 3 | 3V3 | 供电 |
| 4 | NC | 未连接 |
| 5 | IO2 | 预留 |
| 6 | IO3 | 预留 |
| 7 | NC | 未连接 |
| 8 | EN | Reset |
| 9 | NC | 未连接 |
| 10 | NC | 未连接 |
| 11 | GND | 接地 |
| 12 | IO0 | LCD_SDA |
| 13 | IO1 | LCD_SCL |
| 14 | GND | 接地 |
| 15 | NC | 未连接 |
| 16 | IO10 | LCD_CS |
| 17 | NC | 未连接 |
| 18 | IO4 | LCD_D/C |
| 19 | IO5 | LCD_BL_CTRL |
| 20 | IO6 | ENCODER_B |
| 21 | IO7 | ENCODER_A |
| 22 | IO8 | LED |
| 23 | IO9 | BOOT, ENCODER_SW |
| 24 | NC | 未连接 |
| 25 | NC | 未连接 |
| 26 | IO18 | 预留 |
| 27 | IO19 | 预留 |
| 28 | NC | 未连接 |
| 29 | NC | 未连接 |
| 30 | RXD0 | RXD0 |
| 31 | TXD0 | TXD0 |
| 32-35 | NC | 未连接 |
| 36-53 | GND | 接地 |

供电说明

开发板通过 USB-to-USB 端口供电：

USB

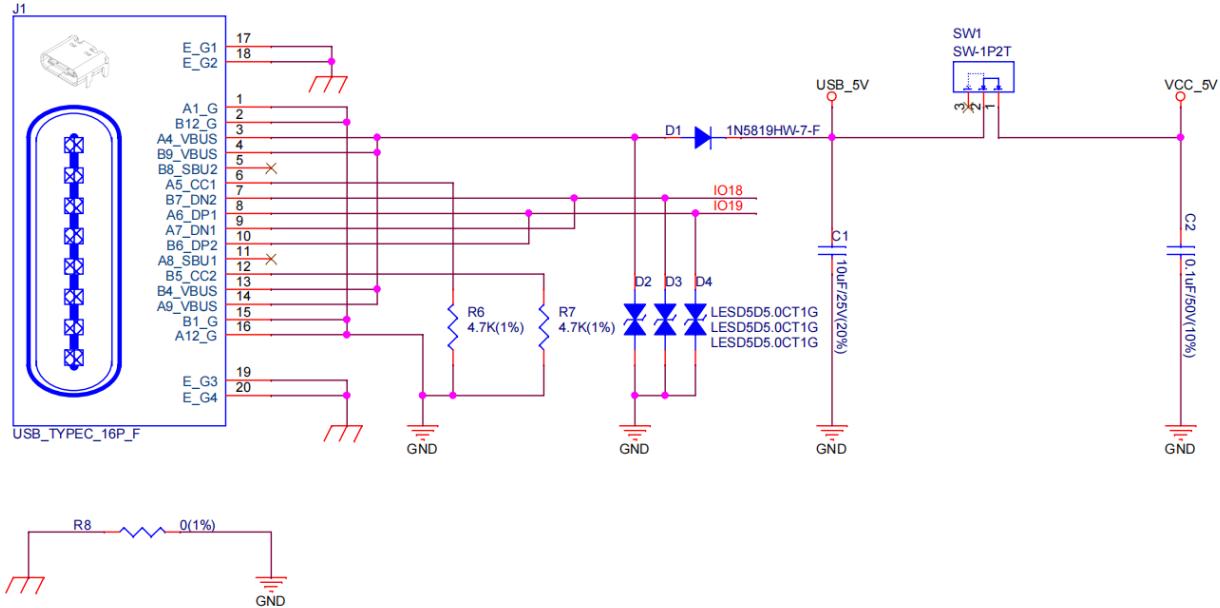


图 16: ESP32-C3-LCD-Ev-Board - USB-to-USB 电源供电

输出系统电源供电：

硬件设置选项

自动下载

可以通过两种方式使 ESP 开发板进入下载模式：

- 手动按下 Boot 和 RST 键，然后先松开 RST，再松开 Boot 键。
- 由 ESP-Prog 控制 ESP 开发板的 EN、IO9 管脚的状态来执行下载。

3V3 LDO

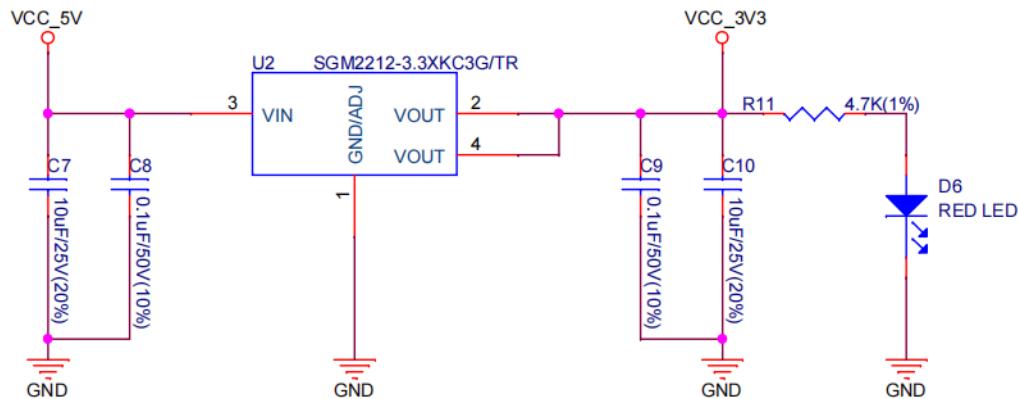


图 17: ESP32-C3-LCD-Ev-Board - 系统电源供电

硬件版本

无历史版本。

样品获取

暂无。

相关文档

- [ESP32-C3 技术规格书](#)
- [ESP32-C3-MINI-1 技术规格书](#)
- [乐鑫产品选型工具](#)
- [ESP32-C3-LCD_EV_Board-MB 原理图](#)
- [ESP32-C3-LCD_EV_Board-MB PCB 布局图](#)
- [ESP32-C3-LCD_EV_Board-DB 原理图](#)
- [ESP32-C3-LCD_EV_Board-DB PCB 布局图](#)

有关本开发板的更多设计文档, 请联系我们的商务部门 sales@espressif.com。

1.1.2 ESP32-C6-DevKitC-1

[English]

ESP32-C6-DevKitC-1 是一款入门级开发板，使用带有 8 MB SPI flash 的通用型模组 ESP32-C6-WROOM-1(U)。该款开发板具备完整的 Wi-Fi、低功耗蓝牙、Zigbee 及 Thread 功能。

ESP32-C6-DevKitC-1 v1.2

[English]

更早版本：[ESP32-C6-DevKitC-1 v1.1](#)

本指南将帮助您快速上手 ESP32-C6-DevKitC-1，并提供该款开发板的详细信息。

ESP32-C6-DevKitC-1 是一款入门级开发板，使用带有 8 MB SPI flash 的通用型模组 ESP32-C6-WROOM-1(U)。该款开发板具备完整的 Wi-Fi、低功耗蓝牙、Zigbee 及 Thread 功能。

板上模组大部分管脚均已引出至两侧排针，开发人员可根据实际需求，轻松通过跳线连接多种外围设备，同时也可将开发板插在面包板上使用。

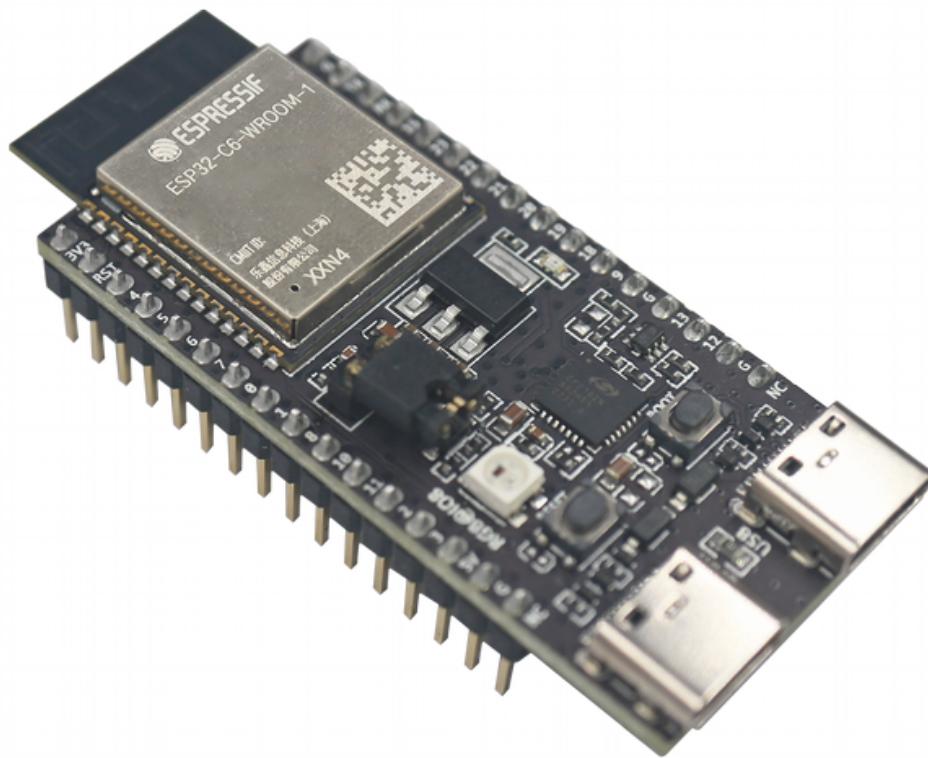


图 18: ESP32-C6-DevKitC-1

本指南包括如下内容：

- [入门指南](#): 简要介绍了 ESP32-C6-DevKitC-1 和硬件、软件设置指南。

- 硬件参考：详细介绍了 ESP32-C6-DevKitC-1 的硬件。
- 硬件版本：介绍硬件历史版本和已知问题，并提供链接至历史版本开发板的入门指南（如有）。
- 相关文档：列出了相关文档的链接。

入门指南

本小节将简要介绍 ESP32-C6-DevKitC-1，说明如何在 ESP32-C6-DevKitC-1 上烧录固件及相关准备工作。

组件介绍

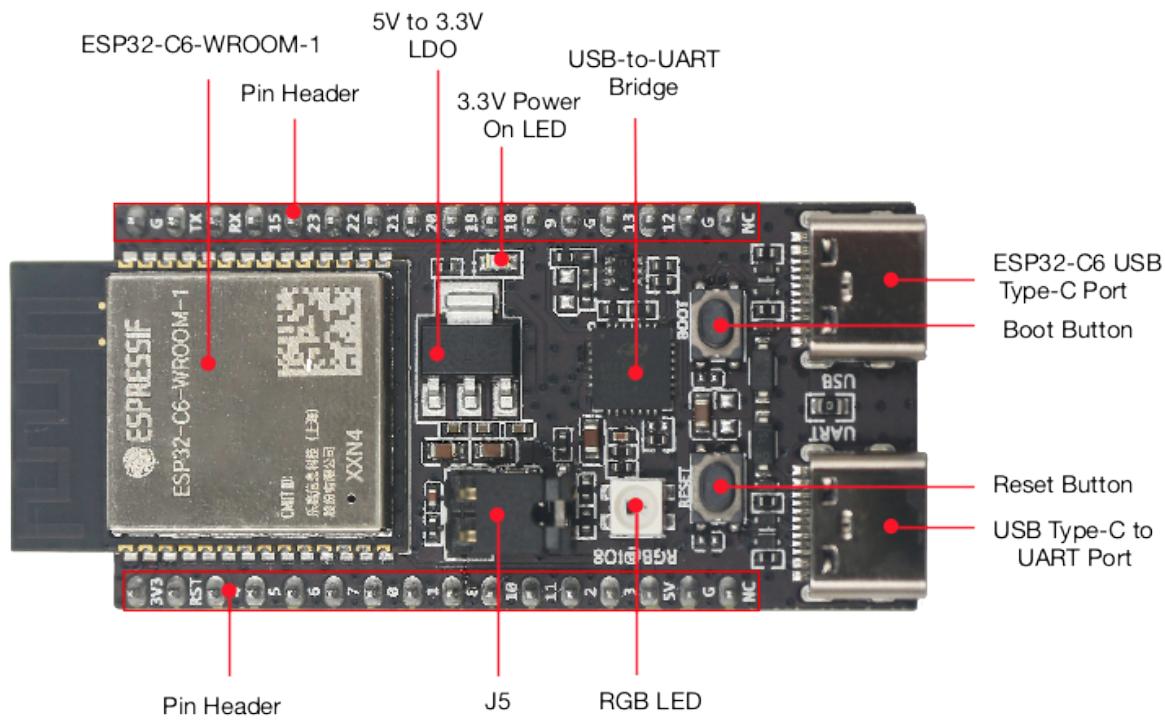


图 19: ESP32-C6-DevKitC-1 - 正面

以下按照顺时针的顺序依次介绍开发板上的主要组件。

| 主要组件 | 介绍 |
|---|---|
| ESP32-C6-WROOM-1 或 ESP32-C6-WROOM-1 | ESP32-C6-WROOM-1 和 ESP32-C6-WROOM-1U 是通用型模组，支持 2.4 GHz Wi-Fi 6、蓝牙 5 及 IEEE 802.15.4 (Zigbee 3.0 和 Thread 1.3)。模组内置 ESP32-C6 芯片，配置 8 MB SPI flash。ESP32-C6-WROOM-1 采用板载 PCB 天线，ESP32-C6-WROOM-1U 采用外部天线连接器。更多信息，请参考 ESP32-C6-WROOM-1 技术规格书 。 |
| Pin Header (排针) | 所有可用 GPIO 管脚（除 flash 的 SPI 总线）均已引出至开发板的排针。 |
| 5 V to 3.3 V LDO (5 V 转 3.3 V LDO) | 电源转换器，输入 5 V，输出 3.3 V。 |
| 3.3 V Power On LED (3.3 V 电源指示灯) | 开发板连接 USB 电源后，该指示灯亮起。 |
| USB-to-UART Bridge (USB 转 UART 桥接器) | 单芯片 USB 转 UART 桥接器，可提供高达 3 Mbps 的传输速率。 |
| ESP32-C6 USB Type-C Port (ESP32-C6 USB Type-C 接口) | ESP32-C6 芯片的 USB Type-C 接口，支持 USB 2.0 全速模式，数据传输速率最高为 12 Mbps（注意，该接口不支持 480 Mbps 的高速传输模式）。该接口可用作开发板的供电接口，可烧录固件至芯片，可通过 USB 协议与芯片通信，也可用于 JTAG 调试。 |
| Boot Button (Boot 键) | 下载按键。按住 Boot 键的同时按一下 Reset 键进入“固件下载”模式，通过串口下载固件。 |
| Reset Button (Reset 键) | 复位按键。 |
| USB Type-C to UART Port (USB Type-C 转 UART 接口) | 可用作开发板的供电接口，可烧录固件至芯片，也可作为通信接口，通过板载 USB 转 UART 桥接器与 ESP32-C6 芯片通信。 |
| RGB LED | 可寻址 RGB 发光二极管，由 GPIO8 驱动。 |
| J5 | 用于测量电流。详见章节 测量电流 。 |

开始开发应用

通电前，请确保 ESP32-C6-DevKitC-1 完好无损。

必备硬件

- ESP32-C6-DevKitC-1
- USB-A 转 USB-C 数据线
- 电脑 (Windows、Linux 或 macOS)

注解：请确保使用优质 USB 数据线。部分数据线仅可用于充电，无法用于数据传输和编程。

软件设置

请前往 [ESP-IDF 快速入门](#)，查看如何快速设置开发环境，将应用程序烧录至您的开发板。

内含组件和包装

零售订单

如购买样品，每个 ESP32-C6-DevKitC-1 将以防静电袋或零售商选择的其他方式包装。

零售订单请前往 <https://www.espressif.com/zh-hans/company/contact/buy-a-sample>。

批量订单

如批量购买，ESP32-C6-DevKitC-1 将以大纸板箱包装。

批量订单请参考 [乐鑫产品订购信息 \(PDF\)](#)。

硬件参考

功能框图

ESP32-C6-DevKitC-1 的主要组件和连接方式如下图所示。

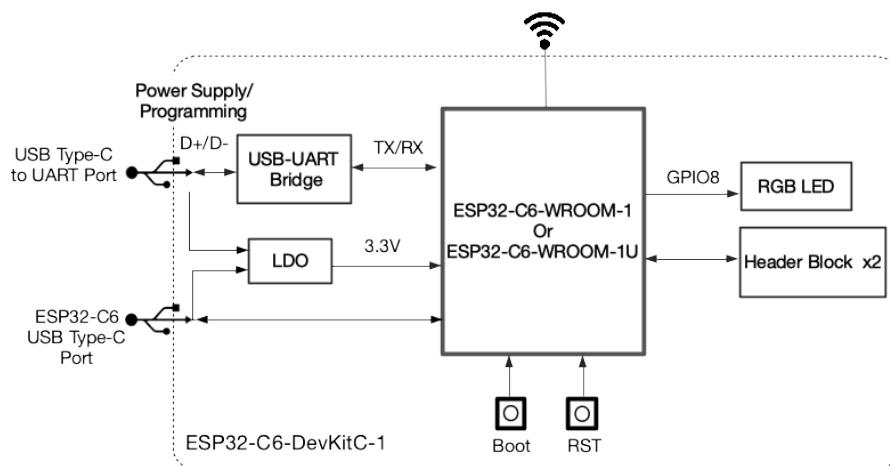


图 20: ESP32-C6-DevKitC-1 (点击放大)

电源选项

您可从以下三种供电方式中任选其一给 ESP32-C6-DevKitC-1 供电：

- USB Type-C 转 UART 接口或 ESP32-C6 USB Type-C 接口供电（选择其一或同时供电），默认供电方式（推荐）
- 5V 和 GND 排针供电
- 3V3 和 GND 排针供电

测量电流

开发板上的 J5 排针（见图[ESP32-C6-DevKitC-1 - 正面](#) 中的 J5）可用于测量 ESP32-C6-WROOM-1(U) 模组的电流：

- 移除 J5 跳帽：此时开发板上外设和模组电源断开，J5 排针接入电流表后可测量模组电流。
- 安装 J5 跳帽（出厂时默认）：开发板恢复正常功能。

注解：使用 3V3 和 GND 排针给开发板供电时，需移除 J5 跳帽，在外部电路上串联接入电流表，才可测量模组的电流。

排针

下表列出了开发板两侧排针（J1 和 J3）的 **名称** 和 **功能**，排针的名称如图[ESP32-C6-DevKitC-1 - 正面](#) 所示，排针的序号与 [ESP32-C6-DevKitC-1 原理图 \(PDF\)](#) 一致。

J1

| 序号 | 名称 | 类型 ¹ | 功能 |
|----|-----|-----------------|--|
| 1 | 3V3 | P | 3.3 V 电源 |
| 2 | RST | I | 高电平：芯片使能；低电平：芯片关闭。 |
| 3 | 4 | I/O/T | MTMS ³ , GPIO4, LP_GPIO4, LP_UART_RXD, ADC1_CH4, FSPIHD |
| 4 | 5 | I/O/T | MTDI ³ , GPIO5, LP_GPIO5, LP_UART_TXD, ADC1_CH5, FSPIWP |
| 5 | 6 | I/O/T | MTCK, GPIO6, LP_GPIO6, LP_I2C_SDA, ADC1_CH6, FSPICLK |
| 6 | 7 | I/O/T | MTDO, GPIO7, LP_GPIO7, LP_I2C_SCL, FSPID |
| 7 | 0 | I/O/T | GPIO0, XTAL_32K_P, LP_GPIO0, LP_UART_DTRN, ADC1_CH0 |
| 8 | 1 | I/O/T | GPIO1, XTAL_32K_N, LP_GPIO1, LP_UART_DSRN, ADC1_CH1 |
| 9 | 8 | I/O/T | GPIO8 ²³ |
| 10 | 10 | I/O/T | GPIO10 |
| 11 | 11 | I/O/T | GPIO11 |
| 12 | 2 | I/O/T | GPIO2, LP_GPIO2, LP_UART_RTSN, ADC1_CH2, FSPIQ |
| 13 | 3 | I/O/T | GPIO3, LP_GPIO3, LP_UART_CTSN, ADC1_CH3 |
| 14 | 5V | P | 5 V 电源 |
| 15 | G | G | 接地 |
| 16 | NC | - | 空管脚 |

¹ P: 电源; I: 输入; O: 输出; T: 可设置为高阻。³ MTMS、MTDI、GPIO8、GPIO9 和 GPIO15 为 ESP32-C6 芯片的 Strapping 管脚。在芯片上电和系统复位过程中，Strapping 管脚根据管脚的二进制电压值控制芯片功能。Strapping 管脚的具体描述和应用，请参考 [ESP32-C6 技术规格书 > Strapping 管脚章节](#)。² 用于驱动 RGB LED。

J3

| 序号 | 名称 | 类型 | 功能 |
|----|----|-------|-----------------------------|
| 1 | G | G | 接地 |
| 2 | TX | I/O/T | U0TXD, GPIO16, FSPICS0 |
| 3 | RX | I/O/T | U0RXD, GPIO17, FSPICS1 |
| 4 | 15 | I/O/T | GPIO15 ³ |
| 5 | 23 | I/O/T | GPIO23, SDIO_DATA3 |
| 6 | 22 | I/O/T | GPIO22, SDIO_DATA2 |
| 7 | 21 | I/O/T | GPIO21, SDIO_DATA1, FSPICS5 |
| 8 | 20 | I/O/T | GPIO20, SDIO_DATA0, FSPICS4 |
| 9 | 19 | I/O/T | GPIO19, SDIO_CLK, FSPICS3 |
| 10 | 18 | I/O/T | GPIO18, SDIO_CMD, FSPICS2 |
| 11 | 9 | I/O/T | GPIO9 ³ |
| 12 | G | G | 接地 |
| 13 | 13 | I/O/T | GPIO13, USB_D+ |
| 14 | 12 | I/O/T | GPIO12, USB_D- |
| 15 | G | G | 接地 |
| 16 | NC | - | 空管脚 |

管脚布局

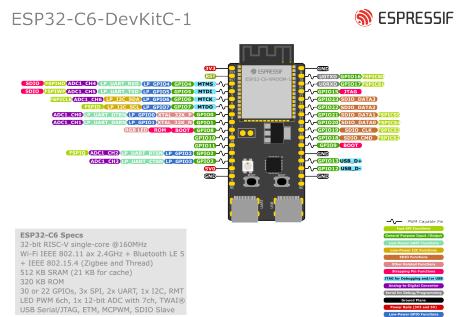


图 21: ESP32-C6-DevKitC-1 管脚布局 (点击放大)

硬件版本

ESP32-C6-DevKitC-1 v1.2

- 2023 年 2 月当月或之后生产的开发板（生产工单号为 PW-2023-02-0139），J5 排针由直针改为弯针。

注解：批量订单的生产工单号可在大纸板箱包装的物料标签中找到。

ESP32-C6-DevKitC-1 v1.1

首次发布

相关文档

- [ESP32-C6 技术规格书 \(PDF\)](#)
- [ESP32-C6-WROOM-1 技术规格书 \(PDF\)](#)
- [ESP32-C6-DevKitC-1 原理图 \(PDF\)](#)
- [ESP32-C6-DevKitC-1 PCB 布局图 \(PDF\)](#)
- [ESP32-C6-DevKitC-1 尺寸图 \(PDF\)](#)
- [ESP32-C6-DevKitC-1 尺寸图源文件 \(DXF\)](#) - 可使用 Autodesk Viewer 查看

有关本开发板的更多设计文档，请联系我们的商务部门 sales@espressif.com。

ESP32-C6-DevKitC-1 v1.1

[English]

最新版本：[ESP32-C6-DevKitC-1 v1.2](#)

本指南将帮助您快速上手 ESP32-C6-DevKitC-1，并提供该款开发板的详细信息。

ESP32-C6-DevKitC-1 是一款入门级开发板，使用带有 8 MB SPI flash 的通用型模组 [ESP32-C6-WROOM-1](#)。该款开发板具备完整的 Wi-Fi、低功耗蓝牙、Zigbee 及 Thread 功能。

板上模组大部分管脚均已引出至两侧排针，开发人员可根据实际需求，轻松通过跳线连接多种外围设备，同时也可将开发板插在面包板上使用。

本指南包括如下内容：

- [入门指南](#)：简要介绍了 ESP32-C6-DevKitC-1 和硬件、软件设置指南。
- [硬件参考](#)：详细介绍了 ESP32-C6-DevKitC-1 的硬件。

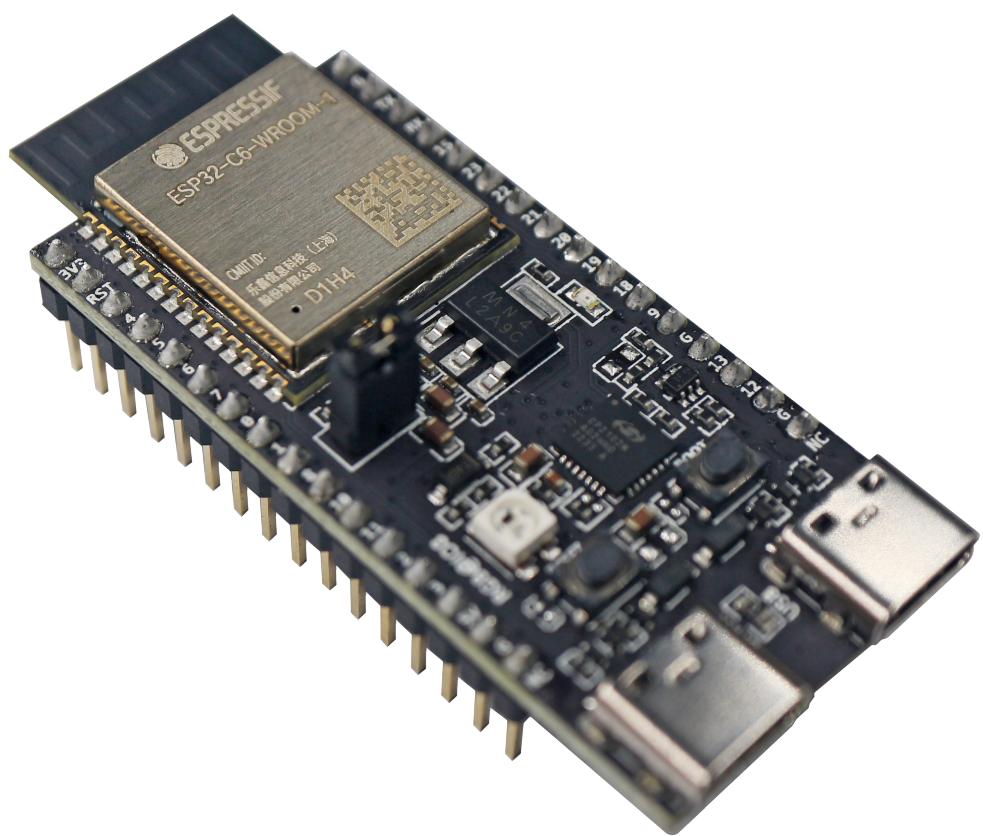


图 22: ESP32-C6-DevKitC-1

- 硬件版本：介绍硬件历史版本和已知问题，并提供链接至历史版本开发板的入门指南（如有）。
- 相关文档：列出了相关文档的链接。

入门指南

本小节将简要介绍 ESP32-C6-DevKitC-1，说明如何在 ESP32-C6-DevKitC-1 上烧录固件及相关准备工作。

组件介绍

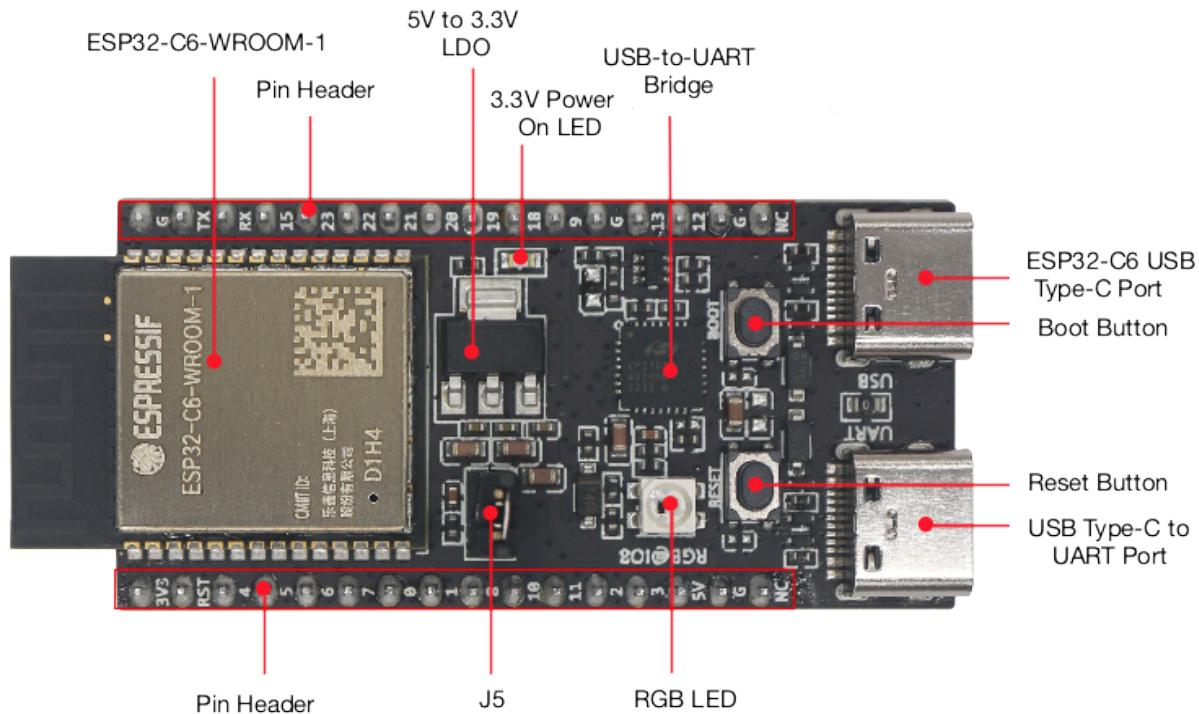


图 23: ESP32-C6-DevKitC-1 - 正面

以下按照顺时针的顺序依次介绍开发板上的主要组件。

| 主要组件 | 介绍 |
|---|--|
| ESP32-C6-WROOM-1 | ESP32-C6-WROOM-1 是一款通用型模组，支持 2.4 GHz Wi-Fi 6、蓝牙 5 及 IEEE 802.15.4 (Zigbee 3.0 和 Thread 1.3)。该款模组内置 ESP32-C6 芯片，采用 PCB 板载天线，配置 8 MB SPI flash。更多信息，请参考 ESP32-C6-WROOM-1 技术规格书 。 |
| Pin Header (排针) | 所有可用 GPIO 管脚（除 flash 的 SPI 总线）均已引出至开发板的排针。 |
| 5 V to 3.3 V LDO (5 V 转 3.3 V LDO) | 电源转换器，输入 5 V，输出 3.3 V。 |
| 3.3 V Power On LED (3.3 V 电源指示灯) | 开发板连接 USB 电源后，该指示灯亮起。 |
| USB-to-UART Bridge (USB 转 UART 桥接器) | 单芯片 USB 转 UART 桥接器，可提供高达 3 Mbps 的传输速率。 |
| ESP32-C6 USB Type-C Port (ESP32-C6 USB Type-C 接口) | ESP32-C6 芯片的 USB Type-C 接口，支持 USB 2.0 全速模式，数据传输速率最高为 12 Mbps（注意，该接口不支持 480 Mbps 的高速传输模式）。该接口可用作开发板的供电接口，可烧录固件至芯片，可通过 USB 协议与芯片通信，也可用于 JTAG 调试。 |
| Boot Button (Boot 键) | 下载按键。按住 Boot 键的同时按一下 Reset 键进入“固件下载”模式，通过串口下载固件。 |
| Reset Button (Reset 键) | 复位按键。 |
| USB Type-C to UART Port (USB Type-C 转 UART 接口) | 可用作开发板的供电接口，可烧录固件至芯片，也可作为通信接口，通过板载 USB 转 UART 桥接器与 ESP32-C6 芯片通信。 |
| RGB LED | 可寻址 RGB 发光二极管，由 GPIO8 驱动。 |
| J5 | 用于测量电流。详见章节 测量电流 。 |

开始开发应用

通电前，请确保 ESP32-C6-DevKitC-1 完好无损。

必备硬件

- ESP32-C6-DevKitC-1
- USB-A 转 USB-C 数据线
- 电脑 (Windows、Linux 或 macOS)

注解：请确保使用优质 USB 数据线。部分数据线仅可用于充电，无法用于数据传输和编程。

软件设置

请前往 [ESP-IDF 快速入门](#)，查看如何快速设置开发环境，将应用程序烧录至您的开发板。

内含组件和包装

零售订单

如购买样品，每个 ESP32-C6-DevKitC-1 将以防静电袋或零售商选择的其他方式包装。

零售订单请前往 <https://www.espressif.com/zh-hans/company/contact/buy-a-sample>。

批量订单

如批量购买，ESP32-C6-DevKitC-1 将以大纸板箱包装。

批量订单请参考 [乐鑫产品订购信息 \(PDF\)](#)。

硬件参考

功能框图

ESP32-C6-DevKitC-1 的主要组件和连接方式如下图所示。

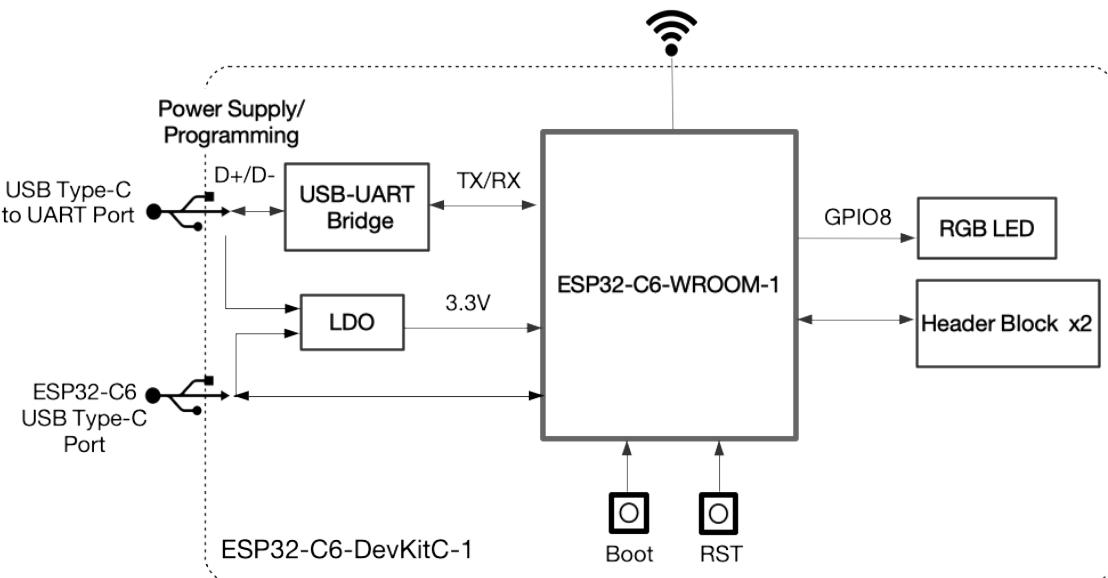


图 24: ESP32-C6-DevKitC-1 (点击放大)

电源选项

您可从以下三种供电方式中任选其一给 ESP32-C6-DevKitC-1 供电：

- USB Type-C 转 UART 接口或 ESP32-C6 USB Type-C 接口供电（选择其一或同时供电），默认供电方式（推荐）
- 5V 和 GND 排针供电
- 3V3 和 GND 排针供电

测量电流

开发板上的 J5 排针（见图[ESP32-C6-DevKitC-1 - 正面](#) 中的 J5）可用于测量 ESP32-C6-WROOM-1 模组的电流：

- 移除 J5 跳帽：此时开发板上外设和模组电源断开，J5 排针接入电流表后可测量模组电流。
- 安装 J5 跳帽（出厂时默认）：开发板恢复正常功能。

注解：使用 3V3 和 GND 排针给开发板供电时，需移除 J5 跳帽，在外部电路上串联接入电流表，才可测量模组的电流。

排针

下表列出了开发板两侧排针（J1 和 J3）的 **名称** 和 **功能**，排针的名称如图[ESP32-C6-DevKitC-1 - 正面](#) 所示，排针的序号与 [ESP32-C6-DevKitC-1 原理图 \(PDF\)](#) 一致。

J1

| 序号 | 名称 | 类型 ¹ | 功能 |
|----|-----|-----------------|--|
| 1 | 3V3 | P | 3.3 V 电源 |
| 2 | RST | I | 高电平：芯片使能；低电平：芯片关闭。 |
| 3 | 4 | I/O/T | MTMS ³ , GPIO4, LP_GPIO4, LP_UART_RXD, ADC1_CH4, FSPIHD |
| 4 | 5 | I/O/T | MTDI ³ , GPIO5, LP_GPIO5, LP_UART_TXD, ADC1_CH5, FSPIWP |
| 5 | 6 | I/O/T | MTCK, GPIO6, LP_GPIO6, LP_I2C_SDA, ADC1_CH6, FSPICLK |
| 6 | 7 | I/O/T | MTDO, GPIO7, LP_GPIO7, LP_I2C_SCL, FSPID |
| 7 | 0 | I/O/T | GPIO0, XTAL_32K_P, LP_GPIO0, LP_UART_DTRN, ADC1_CH0 |
| 8 | 1 | I/O/T | GPIO1, XTAL_32K_N, LP_GPIO1, LP_UART_DSRN, ADC1_CH1 |
| 9 | 8 | I/O/T | GPIO8 ²³ |
| 10 | 10 | I/O/T | GPIO10 |
| 11 | 11 | I/O/T | GPIO11 |
| 12 | 2 | I/O/T | GPIO2, LP_GPIO2, LP_UART_RTSN, ADC1_CH2, FSPIQ |
| 13 | 3 | I/O/T | GPIO3, LP_GPIO3, LP_UART_CTSN, ADC1_CH3 |
| 14 | 5V | P | 5 V 电源 |
| 15 | G | G | 接地 |
| 16 | NC | - | 空管脚 |

¹ P: 电源; I: 输入; O: 输出; T: 可设置为高阻。

³ MTMS、MTDI、GPIO8、GPIO9 和 GPIO15 为 ESP32-C6 芯片的 Strapping 管脚。在芯片上电和系统复位过程中，Strapping 管脚根据管脚的二进制电压值控制芯片功能。Strapping 管脚的具体描述和应用，请参考 [ESP32-C6 技术规格书 > Strapping 管脚章节](#)。

² 用于驱动 RGB LED。

J3

| 序号 | 名称 | 类型 | 功能 |
|----|----|-------|-----------------------------|
| 1 | G | G | 接地 |
| 2 | TX | I/O/T | U0TXD, GPIO16, FSPICS0 |
| 3 | RX | I/O/T | U0RXD, GPIO17, FSPICS1 |
| 4 | 15 | I/O/T | GPIO15 ³ |
| 5 | 23 | I/O/T | GPIO23, SDIO_DATA3 |
| 6 | 22 | I/O/T | GPIO22, SDIO_DATA2 |
| 7 | 21 | I/O/T | GPIO21, SDIO_DATA1, FSPICS5 |
| 8 | 20 | I/O/T | GPIO20, SDIO_DATA0, FSPICS4 |
| 9 | 19 | I/O/T | GPIO19, SDIO_CLK, FSPICS3 |
| 10 | 18 | I/O/T | GPIO18, SDIO_CMD, FSPICS2 |
| 11 | 9 | I/O/T | GPIO9 ³ |
| 12 | G | G | 接地 |
| 13 | 13 | I/O/T | GPIO13, USB_D+ |
| 14 | 12 | I/O/T | GPIO12, USB_D- |
| 15 | G | G | 接地 |
| 16 | NC | - | 空管脚 |

管脚布局

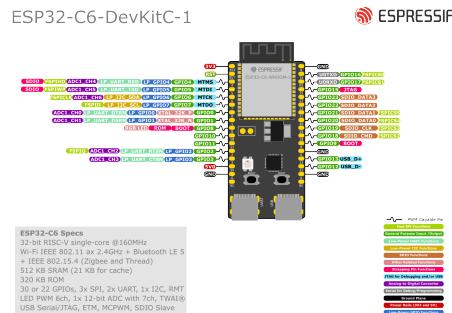


图 25: ESP32-C6-DevKitC-1 管脚布局 (点击放大)

硬件版本

该开发板为最新硬件，尚未有历史版本。

相关文档

- [ESP32-C6 技术规格书 \(PDF\)](#)
- [ESP32-C6-WROOM-1 技术规格书 \(PDF\)](#)
- [ESP32-C6-DevKitC-1 原理图 \(PDF\)](#)
- [ESP32-C6-DevKitC-1 PCB 布局图 \(PDF\)](#)
- [ESP32-C6-DevKitC-1 尺寸图 \(PDF\)](#)
- [ESP32-C6-DevKitC-1 尺寸图源文件 \(DXF\) - 可使用 Autodesk Viewer 查看](#)

有关本开发板的更多设计文档，请联系我们的商务部门 sales@espressif.com。

1.1.3 ESP32-C6-DevKitM-1

[English]

ESP32-C6-DevKitM-1 是一款入门级开发板，使用带有 4 MB SPI flash 的通用型模组 ESP32-C6-MINI-1(U)。该款开发板具备完整的 Wi-Fi、低功耗蓝牙、Zigbee 及 Thread 功能。

ESP32-C6-DevKitM-1

[English]

本指南将帮助您快速上手 ESP32-C6-DevKitM-1，并提供该款开发板的详细信息。

ESP32-C6-DevKitM-1 是一款入门级开发板，使用芯片封装内带有 4 MB SPI flash 的通用型模组 ESP32-C6-MINI-1(U)。该款开发板具备完整的 Wi-Fi、低功耗蓝牙、Zigbee 及 Thread 功能。

板上模组大部分管脚均已引出至两侧排针，开发人员可根据实际需求，轻松通过跳线连接多种外围设备，同时也可将开发板插在面包板上使用。

本指南包括如下内容：

- [入门指南](#): 简要介绍了 ESP32-C6-DevKitM-1 和硬件、软件设置指南。
- [硬件参考](#): 详细介绍了 ESP32-C6-DevKitM-1 的硬件。
- [硬件版本](#): 介绍硬件历史版本和已知问题，并提供链接至历史版本开发板的入门指南（如有）。
- [相关文档](#): 列出了相关文档的链接。

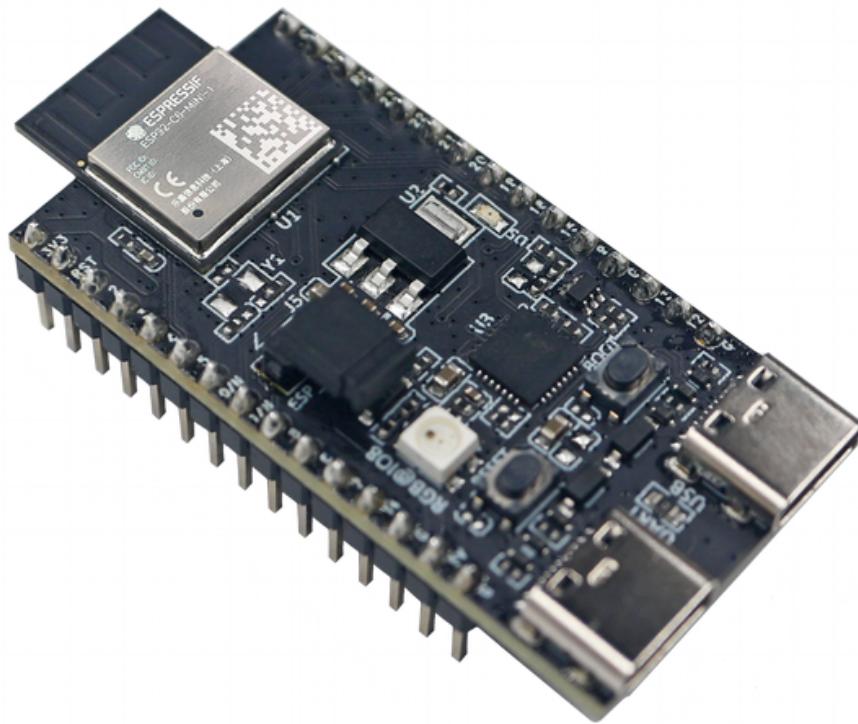


图 26: ESP32-C6-DevKitM-1

入门指南

本小节将简要介绍 ESP32-C6-DevKitM-1，说明如何在 ESP32-C6-DevKitM-1 上烧录固件及相关准备工作。

组件介绍

以下按照顺时针的顺序依次介绍开发板上的主要组件。

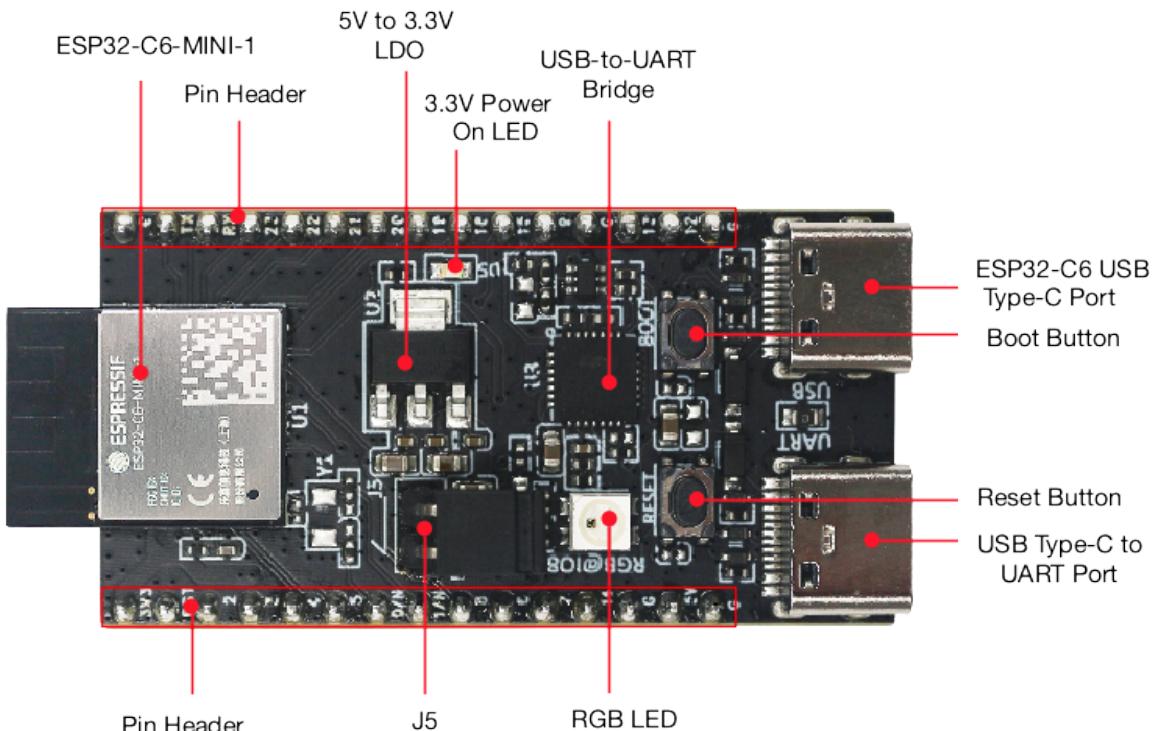


图 27: ESP32-C6-DevKitM-1 - 正面

| 主要组件 | 介绍 |
|---|--|
| ESP32-C6-MINI-1 或 ESP32-C6-MINI-1U | ESP32-C6-MINI-1 和 ESP32-C6-MINI-1U 是通用型模组，支持 2.4 GHz WiFi 6、蓝牙 5 及 IEEE 802.15.4 (Zigbee 3.0 和 Thread 1.3)。ESP32-C6-MINI-1 采用板载 PCB 天线，ESP32-C6-MINI-1U 采用外部天线连接器。模组内置封装内带有 4 MB flash 的 ESP32-C6FH4 芯片。更多信息，请参考 ESP32-C6-MINI-1 技术规格书 。 |
| Pin Header (排针) | 所有可用 GPIO 管脚（除 flash 的 SPI 总线）均已引出至开发板的排针。 |
| 5 V to 3.3 V LDO (5 V 转 3.3 V LDO) | 电源转换器，输入 5 V，输出 3.3 V。 |
| 3.3 V Power On LED (3.3 V 电源指示灯) | 开发板连接 USB 电源后，该指示灯亮起。 |
| USB-to-UART Bridge (USB 转 UART 桥接器) | 单芯片 USB 转 UART 桥接器，可提供高达 3 Mbps 的传输速率。 |
| ESP32-C6 USB Type-C Port (ESP32-C6 USB Type-C 接口) | ESP32-C6 芯片的 USB Type-C 接口，支持 USB 2.0 全速模式，数据传输速率最高为 12 Mbps（注意，该接口不支持 480 Mbps 的高速传输模式）。该接口可用作开发板的供电接口，可烧录固件至芯片，可通过 USB 协议与芯片通信，也可用于 JTAG 调试。 |
| Boot Button (Boot 键) | 下载按键。按住 Boot 键的同时按一下 Reset 键进入“固件下载”模式，通过串口下载固件。 |
| Reset Button (Reset 键) | 复位按键。 |
| USB Type-C to UART Port (USB Type-C 转 UART 接口) | 可用作开发板的供电接口，可烧录固件至芯片，也可作为通信接口，通过板载 USB 转 UART 桥接器与 ESP32-C6 芯片通信。 |
| RGB LED | 可寻址 RGB 发光二极管，由 GPIO8 驱动。 |
| J5 | 用于测量电流。详见章节 测量电流 。 |

开始开发应用

通电前，请确保 ESP32-C6-DevKitM-1 完好无损。

必备硬件

- ESP32-C6-DevKitM-1
- USB-A 转 USB-C 数据线
- 电脑（Windows、Linux 或 macOS）

注解：请确保使用优质 USB 数据线。部分数据线仅可用于充电，无法用于数据传输和编程。

软件设置

请前往 [ESP-IDF 快速入门](#)，查看如何快速设置开发环境，将应用程序烧录至您的开发板。

内含组件和包装

零售订单

如购买样品，每个 ESP32-C6-DevKitM-1 将以防静电袋或零售商选择的其他方式包装。

零售订单请前往 <https://www.espressif.com/zh-hans/company/contact/buy-a-sample>。

批量订单

如批量购买，ESP32-C6-DevKitM-1 将以大纸板箱包装。

批量订单请参考 [乐鑫产品订购信息 \(PDF\)](#)。

硬件参考

功能框图

ESP32-C6-DevKitM-1 的主要组件和连接方式如下图所示。

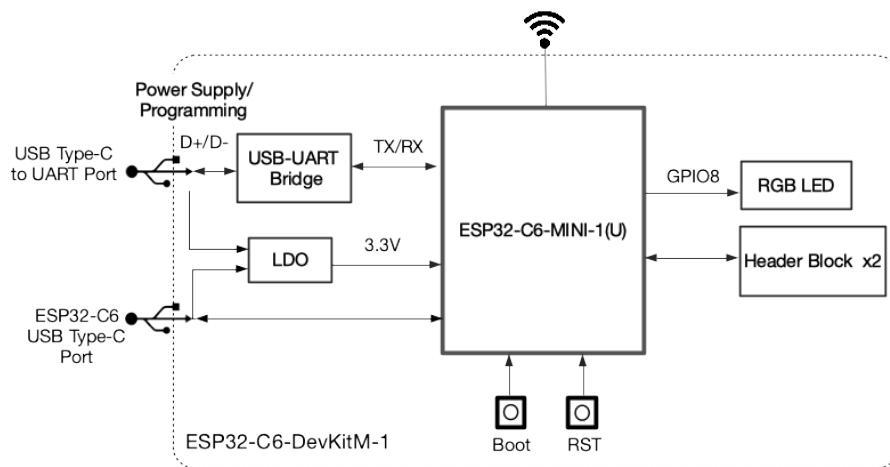


图 28: ESP32-C6-DevKitM-1 (点击放大)

电源选项

您可从以下三种供电方式中任选其一给 ESP32-C6-DevKitM-1 供电：

- USB Type-C 转 UART 接口或 ESP32-C6 USB Type-C 接口供电（选择其一或同时供电），默认供电方式（推荐）
- 5V 和 GND 排针供电
- 3V3 和 GND 排针供电

测量电流

开发板上的 J5 排针（见图`ESP32-C6-DevKitM-1 - 正面`中的 J5）可用于测量 ESP32-C6-MINI-1(U) 模组的电流：

- 移除 J5 跳帽：此时开发板上外设和模组电源断开，J5 排针接入电流表后可测量模组电流。
- 安装 J5 跳帽（出厂时默认）：开发板恢复正常功能。

注解：使用 3V3 和 GND 排针给开发板供电时，需移除 J5 跳帽，在外部电路上串联接入电流表，才可测量模组的电流。

排针

下表列出了开发板两侧排针（J1 和 J3）的 **名称和功能**，排针的名称如图[ESP32-C6-DevKitM-1 - 正面](#) 所示，排针的序号与 [ESP32-C6-DevKitM-1 原理图 \(PDF\)](#) 一致。

J1

| 序号 | 名称 | 类型 ¹ | 功能 |
|----|-----|-----------------|--|
| 1 | 3V3 | P | 3.3 V 电源 |
| 2 | RST | I | 高电平：芯片使能；低电平：芯片关闭。 |
| 3 | 2 | I/O/T | GPIO2, LP_GPIO2, LP_UART_RTSN, ADC1_CH2, FSPIQ |
| 4 | 3 | I/O/T | GPIO3, LP_GPIO3, LP_UART_CTSN, ADC1_CH3 |
| 5 | 4 | I/O/T | MTMS ³ , GPIO4, LP_GPIO4, LP_UART_RXD, ADC1_CH4, FSPIHD |
| 6 | 5 | I/O/T | MTDI ³ , GPIO5, LP_GPIO5, LP_UART_TXD, ADC1_CH5, FSPIWP |
| 7 | 0/N | I/O/T | GPIO0, XTAL_32K_P, LP_GPIO0, LP_UART_DTRN, ADC1_CH0 |
| 8 | 1/N | I/O/T | GPIO1, XTAL_32K_N, LP_GPIO1, LP_UART_DSRN, ADC1_CH1 |
| 9 | 8 | I/O/T | GPIO8 ²³ |
| 10 | 6 | I/O/T | MTCK, GPIO6, LP_GPIO6, LP_I2C_SDA, ADC1_CH6, FSPICLK |
| 11 | 7 | I/O/T | MTDO, GPIO7, LP_GPIO7, LP_I2C_SCL, FSPID |
| 12 | 14 | I/O/T | GPIO14 |
| 13 | G | G | 接地 |
| 14 | 5V | P | 5 V 电源 |
| 15 | G | G | 接地 |

¹ P: 电源；I: 输入；O: 输出；T: 可设置为高阻。

³ MTMS、MTDI、GPIO8、GPIO9 和 GPIO15 为 ESP32-C6 芯片的 Strapping 管脚。在芯片上电和系统复位过程中，Strapping 管脚根据管脚的二进制电压值控制芯片功能。Strapping 管脚的具体描述和应用，请参考 [ESP32-C6 技术规格书 > Strapping 管脚章节](#)。

² 用于驱动 RGB LED。

J3

| 序号 | 名称 | 类型 | 功能 |
|----|----|-------|------------------------------|
| 1 | G | G | 接地 |
| 2 | TX | I/O/T | U0TXD, GPIO16, FSPICS0 |
| 3 | RX | I/O/T | U0RXD, GPIO17, FSPICS1 |
| 4 | 23 | I/O/T | GPIO23, SDIO_DATA3 |
| 5 | 22 | I/O/T | GPIO22, SDIO_DATA2 |
| 6 | 21 | I/O/T | GPIO21, SDIO_DATA1, FSPICSS5 |
| 7 | 20 | I/O/T | GPIO20, SDIO_DATA0, FSPICS4 |
| 8 | 19 | I/O/T | GPIO19, SDIO_CLK, FSPICS3 |
| 9 | 18 | I/O/T | GPIO18, SDIO_CMD, FSPICS2 |
| 10 | 15 | I/O/T | GPIO15 ³ |
| 11 | 9 | I/O/T | GPIO9 ³ |
| 12 | G | G | 接地 |
| 13 | 13 | I/O/T | GPIO13, USB_D+ |
| 14 | 12 | I/O/T | GPIO12, USB_D- |
| 15 | G | G | 接地 |

管脚布局

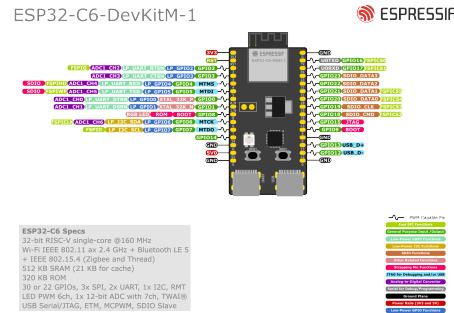


图 29: ESP32-C6-DevKitM-1 管脚布局 (点击放大)

硬件版本

该开发板为最新硬件，尚未有历史版本。

相关文档

- [ESP32-C6 技术规格书 \(PDF\)](#)
- [ESP32-C6-MINI-1 技术规格书 \(PDF\)](#)
- [ESP32-C6-DevKitM-1 原理图 \(PDF\)](#)
- [ESP32-C6-DevKitM-1 PCB 布局图 \(PDF\)](#)
- [ESP32-C6-DevKitM-1 尺寸图 \(PDF\)](#)
- [ESP32-C6-DevKitM-1 尺寸图源文件 \(DXF\) - 可使用 Autodesk Viewer 查看](#)

有关本开发板的更多设计文档，请联系我们的商务部门 sales@espressif.com。

1.1.4 ESP32-H2-DevKitM-1

[English]

ESP32-H2-DevKitM-1 是一款入门级开发板，搭载低功耗蓝牙® 和 IEEE 802.15.4 双模模组 ESP32-H2-MINI-1 或 ESP32-H2-MINI-1U。本指南将帮助您快速上手 ESP32-H2-DevKitM-1，并提供该款开发板的详细信息。

ESP32-H2-DevKitM-1

[English]

本指南将帮助您快速上手 ESP32-H2-DevKitM-1，并提供该款开发板的详细信息。

ESP32-H2-DevKitM-1 是一款入门级开发板，搭载低功耗蓝牙® 和 IEEE 802.15.4 双模模组 ESP32-H2-MINI-1 或 ESP32-H2-MINI-1U。

板上模组的大部分管脚均已引出至开发板两侧排针，开发人员可根据实际需求，轻松通过跳线连接多种外围设备，同时也可将开发板插在面包板上使用。

本指南包括如下内容：

- [入门指南](#): 简要介绍了开发板和硬件、软件设置指南。
- [硬件参考](#): 详细介绍了开发板的硬件。
- [硬件版本](#): 介绍硬件历史版本和已知问题，并提供链接至历史版本开发板的入门指南（如有）。
- [相关文档](#): 列出了相关文档的链接。

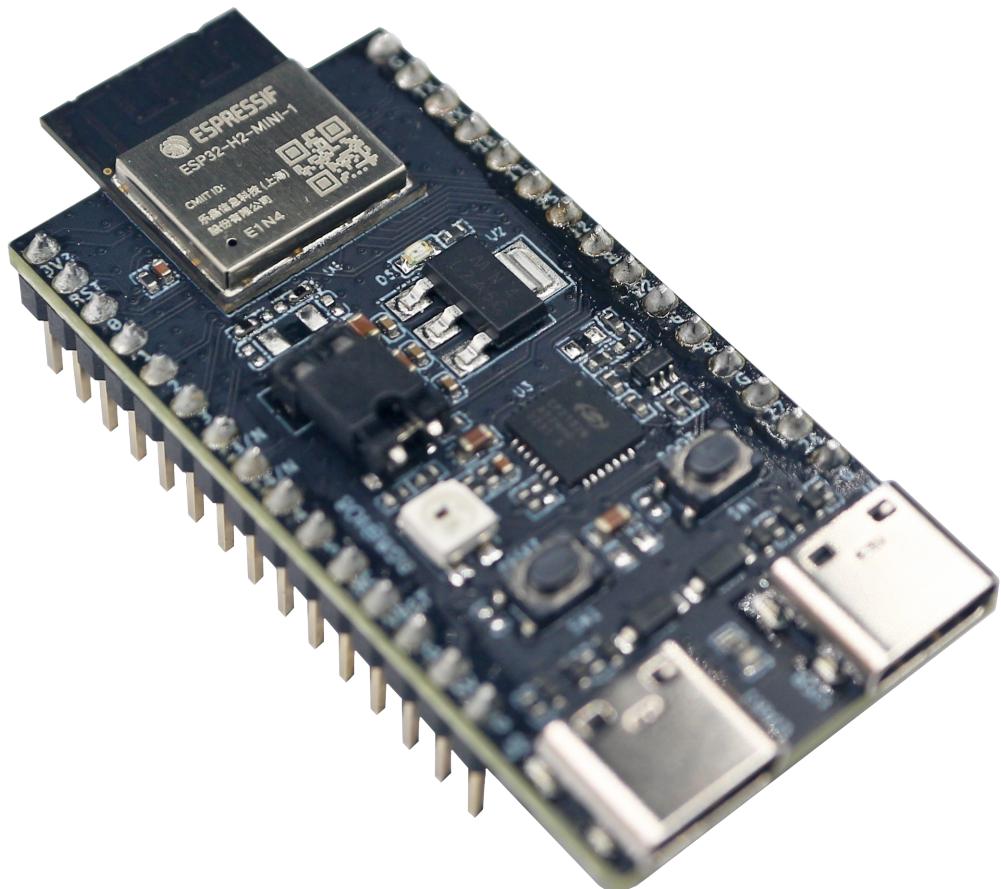


图 30: ESP32-H2-DevKitM-1 (板载 ESP32-H2-MINI-1 模组)

入门指南

本小节将简要介绍 ESP32-H2-DevKitM-1，说明如何在 ESP32-H2-DevKitM-1 上烧录固件及相关准备工作。

组件介绍

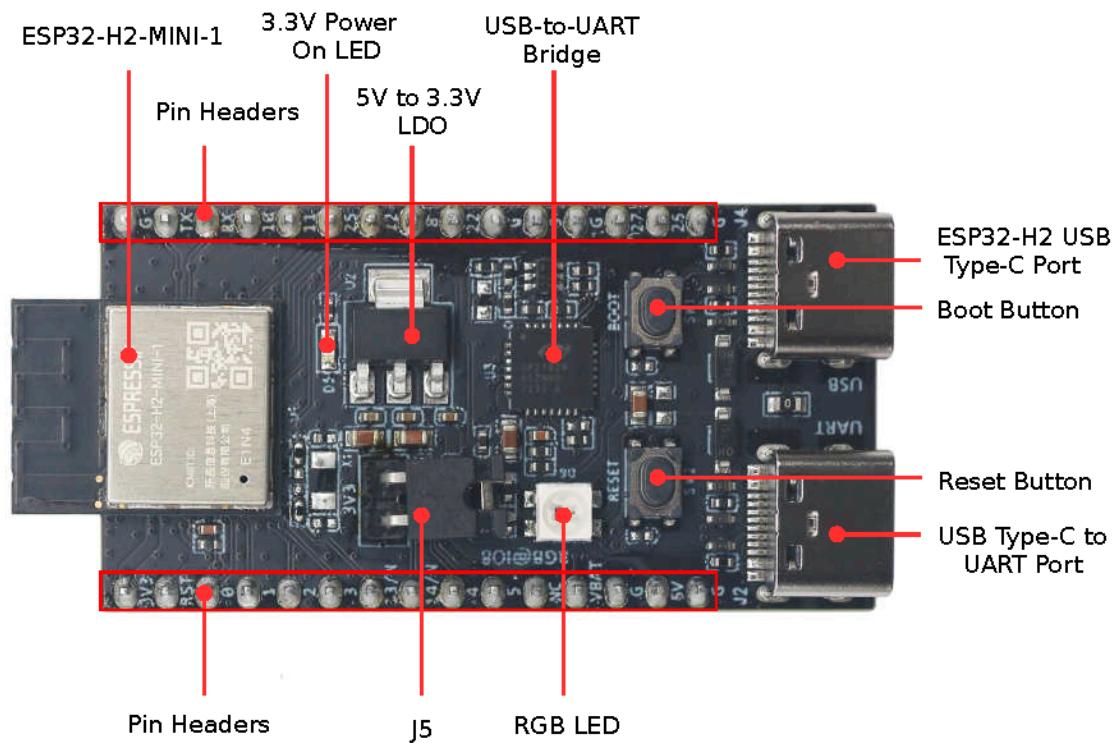


图 31: ESP32-H2-DevKitM-1 - 正面

以下按照顺时针的顺序依次介绍开发板上的主要组件。

| 主要组件 | 介绍 |
|---|--|
| ESP32-H2-MINI-1 或 ESP32-H2-MINI-1U | ESP32-H2-MINI-1/1U 搭载 ESP32-H2 芯片，集成 2.4 GHz 接收器兼容低功耗蓝牙和 IEEE 802.15.4 技术，支持 Bluetooth 5 (LE) , Bluetooth mesh, Zigbee, Thread 和 Matter 等协议，适用于多种低功耗物联网应用场景。 |
| Pin Headers (排针) | 所有可用 GPIO 管脚（除 flash 的 SPI 总线）均已引出至开发板的排针。请查看 排针 获取更多信息 |
| 3.3 V Power On LED (3.3 V 电源指示灯) | 开发板连接 USB 电源后，该指示灯亮起。 |
| 5 V to 3.3 V LDO (5 V 转 3.3 V LDO) | 电源转换器，输入 5 V，输出 3.3 V |
| USB-to-UART Bridge (USB 至 UART 桥接器) | 单芯片 USB 至 UART 桥接器，可提供高达 3 Mbps 的传输速率。 |
| ESP32-H2 USB Type-C Port (ESP32-H2 USB Type-C 接口) | ESP32-H2 芯片的 USB Type-C 接口，支持 USB 2.0 全速模式，数据传输速率最高为 12 Mbps（注意，该接口不支持 480 Mbps 的高速传输模式）。 |
| Boot Button (Boot 键) | 下载按键。按住 Boot 键的同时按一下 Reset 键进入“固件下载”模式，通过串口下载固件。 |
| Reset Button (Reset 键) | 复位按键。 |
| USB Type-C to UART Port (USB Type-C 转 UART 接口) | 可用作开发板的供电接口，也可作为通信接口，通过板载 USB 转 UART 桥接器与 ESP32-H2 芯片通信。 |
| RGB LED | 可寻址 RGB 发光二极管，由 GPIO8 驱动。 |
| J5 | 用于测量电流，详见章节 测量电流 。 |

开始开发应用

通电前，请确保开发板完好无损。

必备硬件

- ESP32-H2-DevKitM-1
- USB-A 转 USB-C (Type C) 数据线
- 电脑（Windows、Linux 或 macOS）

注解：请确保使用适当的 USB 数据线。部分数据线仅可用于充电，无法用于数据传输和编程。

软件设置

请前往 [快速入门](#) 的 [详细安装步骤](#) 小节查看如何快速设置开发环境，将应用程序烧录至您的开发板。

内含组件和包装

订购信息

该开发板有多种型号可供选择，详见下表。

| 订购代码 | 搭载模组 | Flash [A] | PSRAM | 天线 |
|------------------------|------------------|--------------|-------|----------|
| ESP32-H2-DevKitM-1-N4 | ESP32-H2-MINI-1 | 4 MB | 0 MB | PCB 板载天线 |
| ESP32-H2-DevKitM-1U-N4 | ESP32-H2-MINI-1U | 4 MB | 0 MB | 外部天线连接器 |

零售订单

如购买样品，每个开发板将以防静电袋或零售商选择的其他方式包装。

零售订单请前往 <https://www.espressif.com/zh-hans/company/contact/buy-a-sample>。

批量订单

如批量购买，开发板将以大纸板箱包装。

批量订单请前往 <https://www.espressif.com/zh-hans/contact-us/sales-questions>。

硬件参考

功能框图

ESP32-H2-DevKitM-1 的主要组件和连接方式如下图所示。

电源选项

您可从以下三种供电方式中任选其一给开发板供电：

- USB Type-C 转 UART 接口供电（默认）
- 5V 和 G (GND) 排针供电
- 3V3 和 G (GND) 排针供电

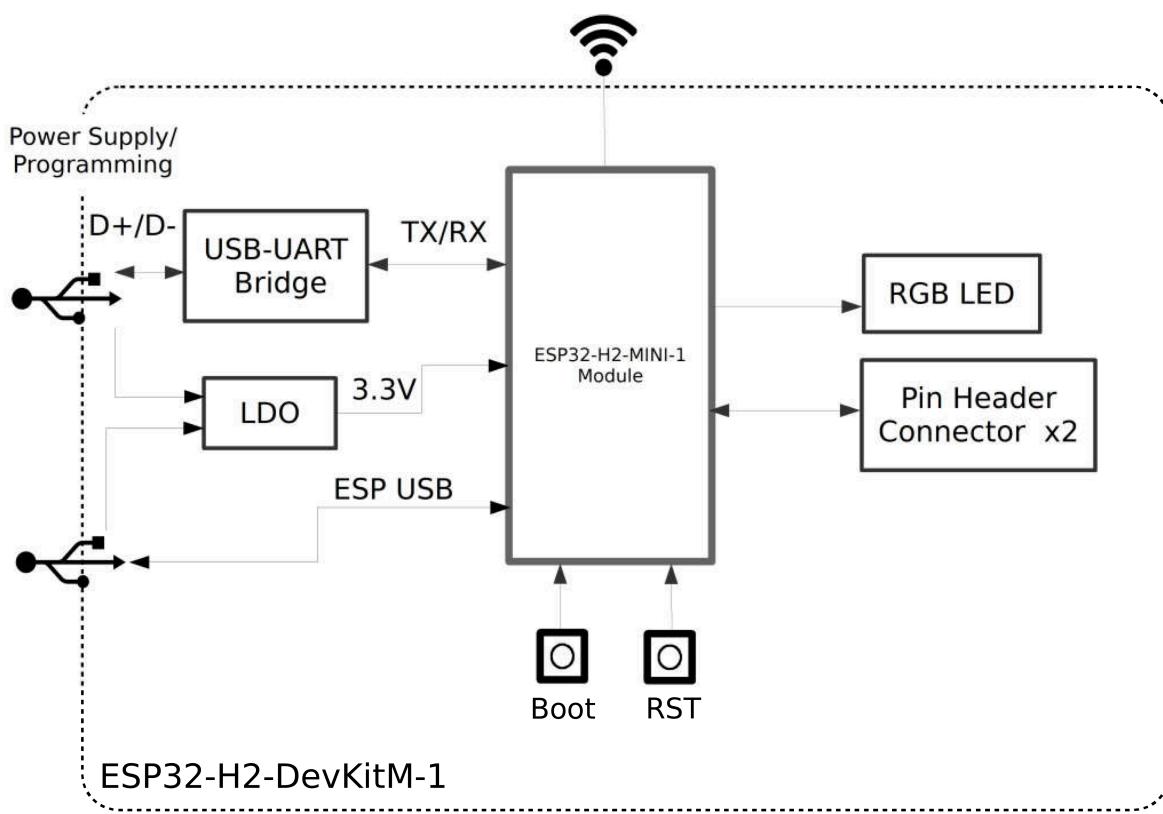


图 32: ESP32-H2-DevKitM-1

测量电流

开发板上的 J5 排针（见图`ESP32-H2-DevKitM-1 - 正面` - 正面中的 J5）可用于测量 ESP32-H2-MINI-1/U 模组的电流：

- 移除 J5 跳帽：此时开发板上外设和模组电源断开，J5 排针接入电流表后可测量模组电流。
- 安装 J5 跳帽（出厂时默认）：开发板恢复正常功能。

注解：使用 3V3 和 GND 排针给开发板供电时，需移除 J5 跳帽，在外部电路上串联接入电流表，才可测量模组的电流。

排针

下表列出了开发板两侧排针（J1 和 J3）的 **名称** 和 **功能**，排针的名称见管脚布局 所示，排针的序号与开发板原理图（随附 PDF）一致。

J1

| 序号 | 名称 | 类型 ¹ | 功能 |
|----|------|-----------------|---------------------------------|
| 1 | 3V3 | P | 3.3 V 电源 |
| 2 | RST | I | 高电平：芯片使能；低电平：芯片关闭；内部默认已上拉。 |
| 3 | 0 | I/O/T | GPIO0, FSPIQ |
| 4 | 1 | I/O/T | GPIO1, FSPICS0, ADC1_CH0 |
| 5 | 2 | I/O/T | GPIO2, FSPIWP, ADC1_CH1, MTMS |
| 6 | 3 | I/O/T | GPIO3, FSPIHD, ADC1_CH2, MTDO |
| 7 | 13/N | I/O/T | GPIO13, XTAL_32K_P ² |
| 8 | 14/N | I/O/T | GPIO14, XTAL_32K_N ³ |
| 9 | 4 | I/O/T | GPIO4, FSPICLK, ADC1_CH3, MTCK |
| 10 | 5 | I/O/T | GPIO5, FSPID, ADC1_CH4, MTDI |
| 11 | NC | — | 空管脚 |
| 12 | VBAT | P | 3.3 V 电源或电池电源 |
| 13 | G | P | 接地 |
| 14 | 5V | P | 5 V 电源 |
| 15 | G | P | 接地 |

¹ P：电源；I：输入；O：输出；T：可设置为高阻。

² 当模组内部选择连接 XTAL_32K_P，不能作他用。

³ 当模组内部选择连接 XTAL_32K_N，不能作他用。

J3

| 序号 | 名称 | 类型 ¹ | 功能 |
|----|----|-----------------|--------------------------|
| 1 | G | P | 接地 |
| 2 | TX | I/O/T | GPIO24, FSPICS2, U0TXD |
| 3 | RX | I/O/T | GPIO23, FSPICS1, U0RXD |
| 4 | 10 | I/O/T | GPIO10, ZCD0 |
| 5 | 11 | I/O/T | GPIO11, ZCD1 |
| 6 | 25 | I/O/T | GPIO25, FSPICS3 |
| 7 | 12 | I/O/T | GPIO12 |
| 8 | 8 | I/O/T | GPIO8 ⁴ , LOG |
| 9 | 22 | I/O/T | GPIO22 |
| 10 | G | P | 接地 |
| 11 | 9 | I/O/T | GPIO9, BOOT |
| 12 | G | P | 接地 |
| 13 | 27 | I/O/T | GPIO27, FSPICS5, USB_D+ |
| 14 | 26 | I/O/T | GPIO26, FSPICS4, USB_D- |
| 15 | G | P | 接地 |

有关管脚功能名称的解释，请参考 [ESP32-H2 技术规格书](#)。

管脚布局

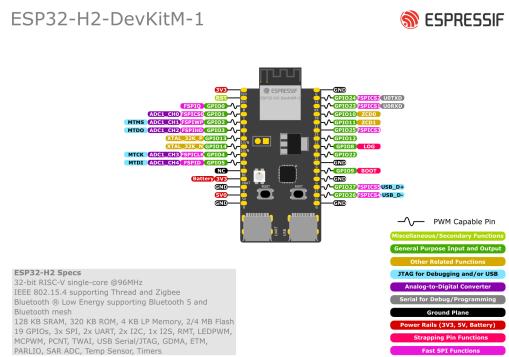


图 33: ESP32-H2-DevKitM-1 管脚布局

⁴ 模组内部已用于驱动 RGB LED。

硬件版本

无历史版本。

相关文档

- [ESP32-H2 技术规格书 \(PDF\)](#)
- [ESP32-H2-MINI-1/1U 技术规格书 \(PDF\)](#)
- [ESP32-H2-DevKitM-1 原理图 \(PDF\)](#)
- [ESP32-H2-DevKitM-1 PCB 布局图 \(PDF\)](#)
- [ESP32-H2-DevKitM-1 尺寸图 \(PDF\)](#)
- [ESP32-H2-DevKitM-1 尺寸图源文件 \(DXF\)](#)

有关本开发板的更多设计文档, 请联系我们的商务部门 sales@espressif.com。

1.1.5 ESP8684-DevKitM-1

[English]

ESP8684-DevKitM-1 是一款入门级开发板, 使用内置 1 MB/2 MB/4 MB SPI flash 的通用型模组 ESP8684-MINI-1。该款开发板具备完整的 Wi-Fi 和低功耗蓝牙功能。

ESP8684-DevKitM-1 v1.1

[English]

更早版本: [ESP8684-DevKitM-1](#)

本指南将帮助您快速上手 ESP8684-DevKitM-1, 并提供该款开发板的详细信息。

ESP8684-DevKitM-1 是一款入门级开发板, 使用内置 1 MB/2 MB/4 MB SPI flash 的通用型模组 ESP8684-MINI-1。该款开发板具备完整的 Wi-Fi 和低功耗蓝牙功能。

板上模组的大部分管脚均已引出至开发板两侧排针, 开发人员可根据实际需求, 轻松通过跳线连接多种外围设备, 同时也可将开发板插在面包板上使用。

本指南包括如下内容:

- [入门指南](#): 简要介绍了开发板和硬件、软件设置指南。
- [硬件概况](#): 详细介绍了开发板的硬件。
- [硬件版本](#): 介绍硬件历史版本和已知问题, 并提供链接至历史版本开发板的入门指南 (如有)。
- [相关文档](#): 列出了相关文档的链接。

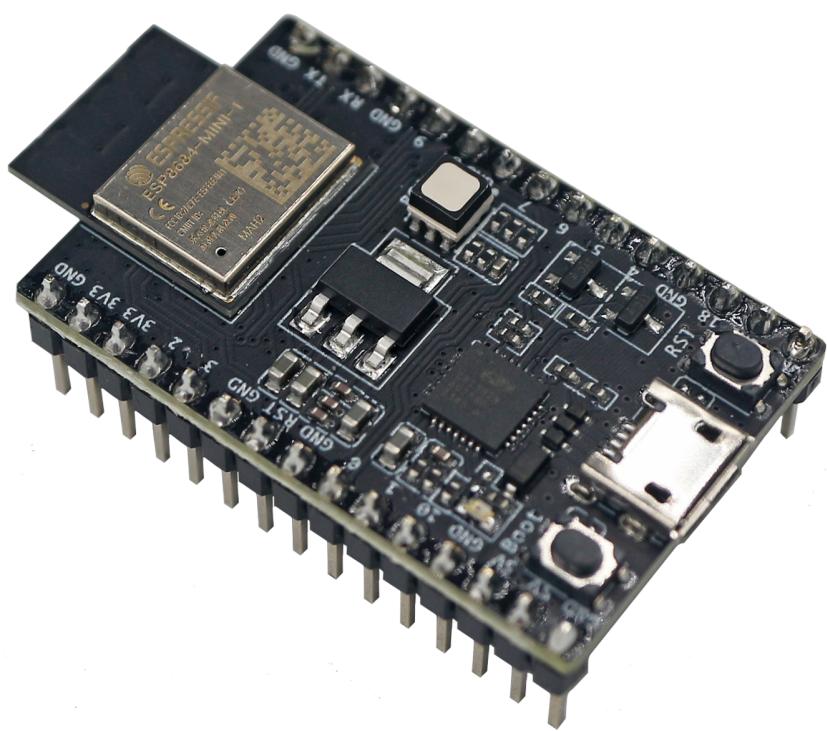


图 34: ESP8684-DevKitM-1 (板载 ESP8684-MINI-1 模组)

入门指南

本小节将简要介绍 ESP8684-DevKitM-1，说明如何在 ESP8684-DevKitM-1 上烧录固件及相关准备工作。

组件介绍

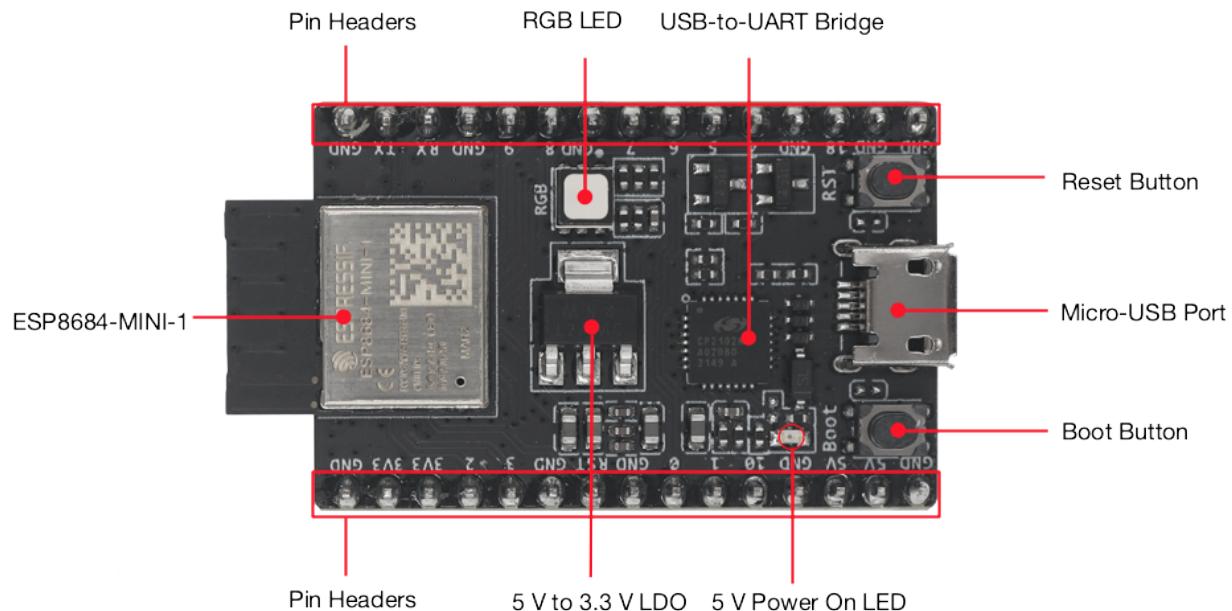


图 35: ESP8684-DevKitM-1 - 正面

以下按照逆时针的顺序依次介绍开发板上的主要组件。

| 主要组件 | 介绍 |
|-------------------------------------|---|
| ESP8684-MINI-1 | ESP8684-MINI-1 是乐鑫推出的一款通用型 Wi-Fi 和低功耗蓝牙双模模组，功能强大。该模组采用 PCB 板载天线，配置了 1 MB/2 MB/4 MB SPI flash。 |
| Pin Headers (排针) | 所有可用 GPIO 管脚均已引出至开发板的排针。请查看 排针 获取更多信息。 |
| 5 V to 3.3 V LDO (5 V 转 3.3 V LDO) | 电源转换器，输入 5 V，输出 3.3 V。 |
| 5 V Power On LED (5 V 电源指示灯) | 开发板连接 USB 电源后，该指示灯亮起。 |
| Boot Button (Boot 键) | 下载按键。按住 Boot 键的同时按一下 Reset 键进入“固件下载”模式，通过串口下载固件。 |
| Micro-USB Port (Micro-USB 接口) | USB 接口。可用作开发板的供电电源或 PC 和 ESP8684 芯片的通信接口。 |
| Reset Button (Reset 键) | 复位按键。 |
| USB-to-UART Bridge (USB 至 UART 桥接器) | 单芯片 USB 至 UART 桥接器，可提供高达 3 Mbps 的传输速率。 |
| RGB LED | RGB 发光二极管，由 GPIO0、GPIO1、和 GPIO8 驱动。 |

开始开发应用

通电前，请确保开发板完好无损。

必备硬件

- ESP8684-DevKitM-1
- USB 2.0 数据线（标准 A 型转 Micro-B 型）
- 电脑（Windows、Linux 或 macOS）

注解：请确保使用适当的 USB 数据线。部分数据线仅可用于充电，无法用于数据传输和编程。

软件设置

请前往 [快速入门](#)，在 [详细安装步骤](#) 小节查看如何快速设置开发环境，将应用程序烧录至您的开发板。

ESP-AT 固件支持

ESP8684-DevKitM-1 支持使用 ESP-AT 指令集，无需对本开发板上的模组进行软件开发即可快速实现产品的无线通讯功能。

乐鑫提供官方的 ESP-AT 固件，您可以前往 [ESP-AT 仓库](#) 选择并下载。

如果您需要自定义固件，或查询更多信息，请参考 [ESP-AT 用户指南](#)。

内含组件和包装

零售订单

如购买样品，每个开发板将以防静电袋或零售商选择的其他方式包装。

零售订单请前往 <https://www.espressif.com/zh-hans/company/contact/buy-a-sample>。

批量订单

如批量购买，开发板将以大纸板箱包装。

批量订单请前往 <https://www.espressif.com/zh-hans/contact-us/sales-questions>。

硬件概况

功能框图

ESP8684-DevKitM-1 的主要组件和连接方式如下图所示。

电源选项

您可从以下三种供电方式中任选其一给开发板供电：

- Micro-USB 接口供电，默认供电方式（推荐）
- 5V 和 G (GND) 排针供电
- 3V3 和 G (GND) 排针供电

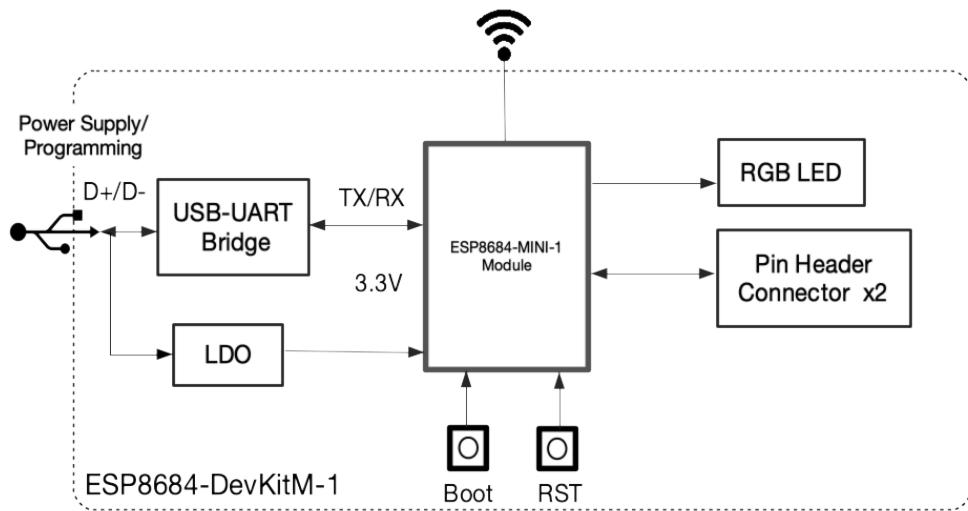


图 36: ESP8684-DevKitM-1 (点击放大)

排针

下表列出了开发板两侧排针 (J1 和 J3) 的 **名称** 和 **功能**, 排针的名称如图 [ESP8684-DevKitM-1 - 正面](#) 所示, 排针的序号与 [开发板原理图 \(PDF\)](#) 一致。

J1

| 序号 | 名称 | 类型 ¹ | 功能 |
|----|-----|-----------------|----------------------------|
| 1 | G | G | 接地 |
| 2 | 3V3 | P | 3.3 V 电源 |
| 3 | 3V3 | P | 3.3 V 电源 |
| 4 | 2 | I/O/T | GPIO2, ADC1_CH2, FSPIQ |
| 5 | 3 | I/O/T | GPIO3, ADC1_CH3 |
| 6 | G | G | 接地 |
| 7 | RST | I | 复位; 高电平: 使能; 低电平: 关闭 |
| 8 | G | G | 接地 |
| 9 | 0 | I/O/T | GPIO0, ADC1_CH0, LED Red |
| 10 | 1 | I/O/T | GPIO1, ADC1_CH1, LED Green |
| 11 | 10 | I/O/T | GPIO10, FSPICS0 |
| 12 | G | G | 接地 |
| 13 | 5V | P | 5 V 电源 |
| 14 | 5V | P | 5 V 电源 |
| 15 | G | G | 接地 |

J3

| 序号 | 名称 | 类型 ¹ | 功能 |
|----|----|-----------------|-------------------------------|
| 1 | G | G | 接地 |
| 2 | TX | I/O/T | GPIO20, U0TXD |
| 3 | RX | I/O/T | GPIO19, U0RXD |
| 4 | G | G | 接地 |
| 5 | 9 | I/O/T | GPIO9 ² |
| 6 | 8 | I/O/T | GPIO8 ² , LED Blue |
| 7 | G | G | 接地 |
| 8 | 7 | I/O/T | GPIO7, FSPID, MTDO |
| 9 | 6 | I/O/T | GPIO6, FSPICLK, MTCK |
| 10 | 5 | I/O/T | GPIO5, ADC2_CH0, FSPIWP, MTDI |
| 11 | 4 | I/O/T | GPIO4, ADC1_CH4, FSPIHD, MTMS |
| 12 | G | G | 接地 |
| 13 | 18 | I/O/T | GPIO18 |
| 14 | G | G | 接地 |
| 15 | G | G | 接地 |

管脚布局

ESP8684-DevKitM-1

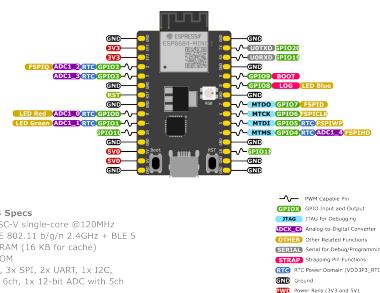


图 37: ESP8684-DevKitM-1 管脚布局 (点击放大)

¹ P: 电源; I: 输入; O: 输出; T: 可设置为高阻。

² GPIO8 和 GPIO9 为 ESP8684 芯片的 Strapping 管脚。在芯片上电和系统复位过程中，Strapping 管脚根据管脚的二进制电压值控制芯片功能。Strapping 管脚的具体描述和应用，请参考 [ESP8684 技术规格书 > Strapping 管脚章节](#)。

硬件版本

首次发布

与旧版本相比的主要区别：

- 旧版本中是可寻址 RGB 发光二极管，该版本中 RGB 发光二极管的每个颜色均由独立的管脚驱动。
- 旧版本中可寻址 RGB 发光二极管由 GPIO8 驱动，该版本中 RGB 发光二极管由 GPIO0、GPIO1、和 GPIO8 驱动。

注解：目前两个版本开发板均可订购。

相关文档

- [ESP8684 技术规格书 \(PDF\)](#)
- [ESP8684-DevKitM-1 原理图 \(PDF\)](#)
- [ESP8684-DevKitM-1 PCB 布局图 \(PDF\)](#)
- [ESP8684-DevKitM-1 尺寸图 \(PDF\)](#)
- [ESP8684-DevKitM-1 尺寸图源文件 \(DXF\) - 可使用 Autodesk Viewer 查看](#)

有关本开发板的更多设计文档，请联系我们的商务部门 sales@espressif.com。

ESP8684-DevKitM-1

[English]

最新版本：[ESP8684-DevKitM-1 v1.1](#)

本指南将帮助您快速上手 ESP8684-DevKitM-1，并提供该款开发板的详细信息。

ESP8684-DevKitM-1 是一款入门级开发板，使用内置 1 MB/2 MB/4 MB SPI flash 的通用型模组 ESP8684-MINI-1。该款开发板具备完整的 Wi-Fi 和低功耗蓝牙功能。

板上模组的大部分管脚均已引出至开发板两侧排针，开发人员可根据实际需求，轻松通过跳线连接多种外围设备，同时也可将开发板插在面包板上使用。

本指南包括如下内容：

- [入门指南](#)：简要介绍了开发板和硬件、软件设置指南。
- [硬件概况](#)：详细介绍了开发板的硬件。
- [硬件版本](#)：介绍硬件历史版本和已知问题，并提供链接至历史版本开发板的入门指南（如有）。
- [相关文档](#)：列出了相关文档的链接。

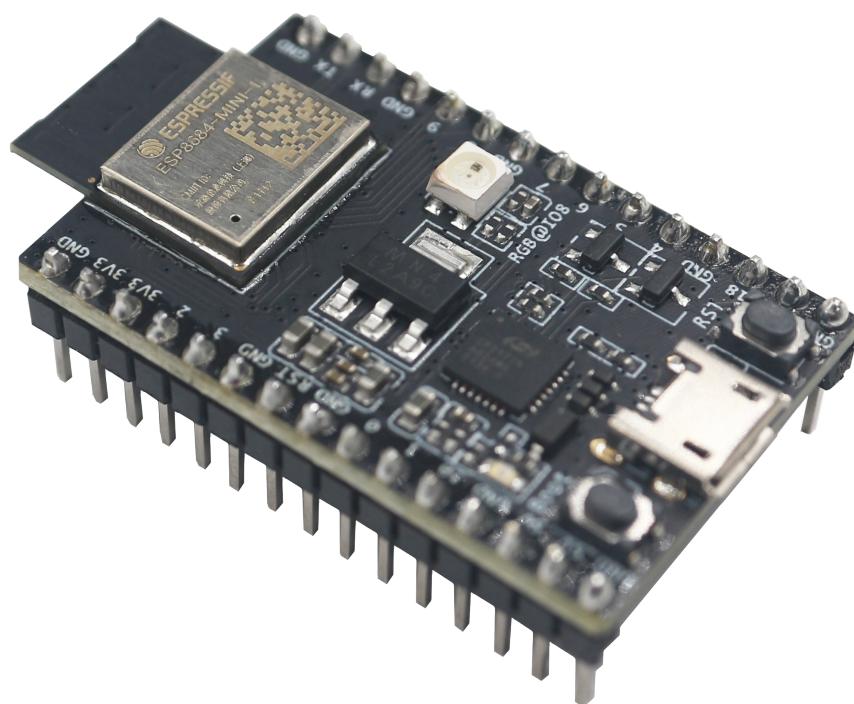


图 38: ESP8684-DevKitM-1 (板载 ESP8684-MINI-1 模组)

入门指南

本小节将简要介绍 ESP8684-DevKitM-1，说明如何在 ESP8684-DevKitM-1 上烧录固件及相关准备工作。

组件介绍

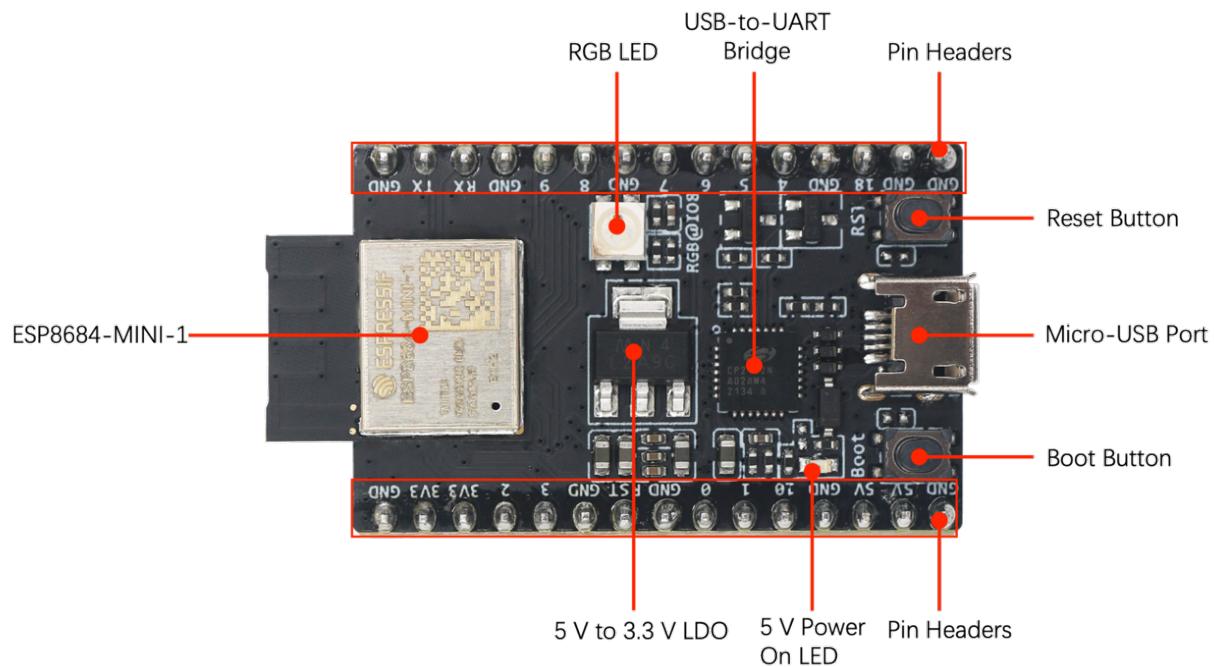


图 39: ESP8684-DevKitM-1 - 正面

以下按照逆时针的顺序依次介绍开发板上的主要组件。

| 主要组件 | 介绍 |
|-------------------------------------|---|
| ESP8684-MINI-1 | ESP8684-MINI-1 是乐鑫推出的一款通用型 Wi-Fi 和低功耗蓝牙双模模组，功能强大。该模组采用 PCB 板载天线，配置了 1 MB/2 MB/4 MB SPI flash。 |
| 5 V to 3.3 V LDO (5 V 转 3.3 V LDO) | 电源转换器，输入 5 V，输出 3.3 V。 |
| 5 V Power On LED (5 V 电源指示灯) | 开发板连接 USB 电源后，该指示灯亮起。 |
| Pin Headers (排针) | 所有可用 GPIO 管脚均已引出至开发板的排针。请查看 排针 获取更多信息。 |
| Boot Button (Boot 键) | 下载按键。按住 Boot 键的同时按一下 Reset 键进入“固件下载”模式，通过串口下载固件。 |
| Micro-USB Port (Micro-USB 接口) | USB 接口。可用作开发板的供电电源或 PC 和 ESP8684 芯片的通信接口。 |
| Reset Button (Reset 键) | 复位按键。 |
| USB-to-UART Bridge (USB 至 UART 桥接器) | 单芯片 USB 至 UART 桥接器，可提供高达 3 Mbps 的传输速率。 |
| RGB LED | 可寻址 RGB 发光二极管，由 GPIO8 驱动。 |

开始开发应用

通电前，请确保开发板完好无损。

必备硬件

- ESP8684-DevKitM-1
- USB 2.0 数据线（标准 A 型转 Micro-B 型）
- 电脑（Windows、Linux 或 macOS）

注解：请确保使用适当的 USB 数据线。部分数据线仅可用于充电，无法用于数据传输和编程。

软件设置

请前往 [快速入门](#)，在 [详细安装步骤](#) 小节查看如何快速设置开发环境，将应用程序烧录至您的开发板。

ESP-AT 固件支持

ESP8684-DevKitM-1 支持使用 ESP-AT 指令集，无需对本开发板上的模组进行软件开发即可快速实现产品的无线通讯功能。

乐鑫提供官方的 ESP-AT 固件，您可以前往 [ESP-AT 仓库](#) 选择并下载。

如果您需要自定义固件，或查询更多信息，请参考 [ESP-AT 用户指南](#)。

内含组件和包装

零售订单

如购买样品，每个开发板将以防静电袋或零售商选择的其他方式包装。

零售订单请前往 <https://www.espressif.com/zh-hans/company/contact/buy-a-sample>。

批量订单

如批量购买，开发板将以大纸板箱包装。

批量订单请前往 <https://www.espressif.com/zh-hans/contact-us/sales-questions>。

硬件概况

功能框图

ESP8684-DevKitM-1 的主要组件和连接方式如下图所示。

电源选项

您可从以下三种供电方式中任选其一给开发板供电：

- Micro-USB 接口供电，默认供电方式（推荐）
- 5V 和 G (GND) 排针供电
- 3V3 和 G (GND) 排针供电

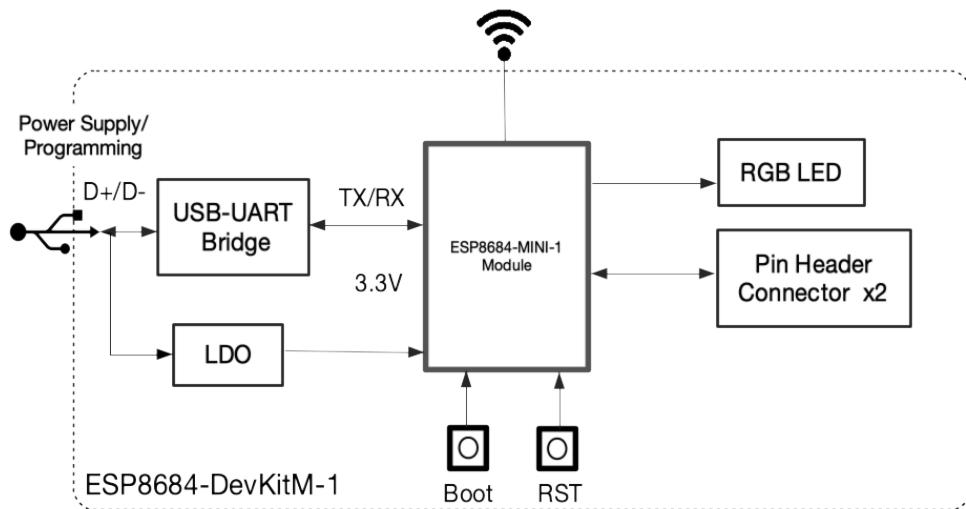


图 40: ESP8684-DevKitM-1 (点击放大)

排针

下表列出了开发板两侧排针 (J1 和 J3) 的 **名称** 和 **功能**, 排针的名称如图[ESP8684-DevKitM-1 - 正面](#) 所示, 排针的序号与 [开发板原理图 \(PDF\)](#) 一致。

J1

| 序号 | 名称 | 类型 ¹ | 功能 |
|----|-----|-----------------|------------------------|
| 1 | G | G | 接地 |
| 2 | 3V3 | P | 3.3 V 电源 |
| 3 | 3V3 | P | 3.3 V 电源 |
| 4 | 2 | I/O/T | GPIO2, ADC1_CH2, FSPIQ |
| 5 | 3 | I/O/T | GPIO3, ADC1_CH3 |
| 6 | G | G | 接地 |
| 7 | RST | I | 复位; 高电平: 使能; 低电平: 关闭 |
| 8 | G | G | 接地 |
| 9 | 0 | I/O/T | GPIO0, ADC1_CH0 |
| 10 | 1 | I/O/T | GPIO1, ADC1_CH1 |
| 11 | 10 | I/O/T | GPIO10, FSPICS0 |
| 12 | G | G | 接地 |
| 13 | 5V | P | 5 V 电源 |
| 14 | 5V | P | 5 V 电源 |
| 15 | G | G | 接地 |

J3

| 序号 | 名称 | 类型 ¹ | 功能 |
|----|----|-----------------|-------------------------------|
| 1 | G | G | 接地 |
| 2 | TX | I/O/T | GPIO20, U0TXD |
| 3 | RX | I/O/T | GPIO19, U0RXD |
| 4 | G | G | 接地 |
| 5 | 9 | I/O/T | GPIO9 ² |
| 6 | 8 | I/O/T | GPIO8 ² , RGB LED |
| 7 | G | G | 接地 |
| 8 | 7 | I/O/T | GPIO7, FSPID , MTDO |
| 9 | 6 | I/O/T | GPIO6, FSPICLK , MTCK |
| 10 | 5 | I/O/T | GPIO5, ADC2_CH0, FSPIWP ,MTDI |
| 11 | 4 | I/O/T | GPIO4, ADC1_CH4, FSPIHD ,MTMS |
| 12 | G | G | 接地 |
| 13 | 18 | I/O/T | GPIO18 |
| 14 | G | G | 接地 |
| 15 | G | G | 接地 |

管脚布局

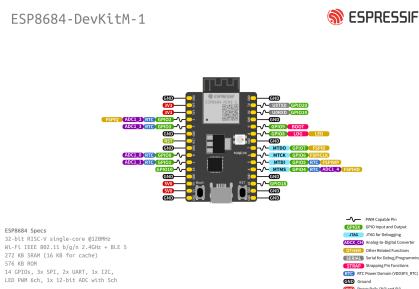


图 41: ESP8684-DevKitM-1 管脚布局 (点击放大)

¹ P: 电源; I: 输入; O: 输出; T: 可设置为高阻。

² GPIO8 和 GPIO9 为 ESP8684 芯片的 Strapping 管脚。在芯片上电和系统复位过程中, Strapping 管脚根据管脚的二进制电压值控制芯片功能。

硬件版本

无历史版本。

相关文档

- [ESP8684-DevKitM-1 原理图 \(PDF\)](#)
- [ESP8684-DevKitM-1 PCB 布局图 \(PDF\)](#)
- [ESP8684-DevKitM-1 尺寸图 \(PDF\)](#)
- [ESP8684-DevKitM-1 尺寸图源文件 \(DXF\) - 可使用 Autodesk Viewer 查看](#)

有关本开发板的更多设计文档, 请联系我们的商务部门 sales@espressif.com。

1.1.6 ESP32-S3-USB-OTG

[\[English\]](#)

ESP32-S3-USB-OTG 是一款侧重于 USB-OTG 功能验证和应用开发的开发板。

开发板的应用示例存放在 [Examples](#) 文件夹中。

ESP32-S3-USB-OTG

[\[English\]](#)

ESP32-S3-USB-OTG 是一款侧重于 USB-OTG 功能验证和应用开发的开发板, 基于 ESP32-S3 SoC, 支持 Wi-Fi 和 BLE 5.0 无线功能, 支持 USB 主机和 USB 从机功能。可用于开发无线存储设备, Wi-Fi 网卡, LTE MiFi, 多媒体设备, 虚拟键鼠等应用。该开发板具有以下特性:

- 板载 ESP32-S3-MINI-1-N8 模组, 内置 8 MB flash
- 板载 USB Type-A 主机和从机接口, 内置接口切换电路
- 板载 USB 转串口调试芯片 (micro USB 接口)
- 板载 1.3 英寸 LCD 彩屏, 支持 GUI
- 板载 SD 卡接口, 兼容 SDIO 和 SPI 接口
- 板载充电 IC, 可外接锂电池

本指南包括:

- [快速入门](#): 提供 ESP32-S3-USB-OTG 的简要概述及必须了解的硬件和软件信息。
- [硬件参考](#): 提供 ESP32-S3-USB-OTG 的详细硬件信息。
- [相关文档](#): 提供相关文档的链接。



图 42: ESP32-S3-USB-OTG (点击图片放大)

快速入门

本节介绍如何开始使用 ESP32-S3-USB-OTG。首先，介绍一些关于 ESP32-S3-USB-OTG 的基本信息，然后在应用程序开发章节介绍如何开始使用该开发板进行开发，最后介绍该开发板包装和零售信息。

组件介绍

ESP32-S3-USB-OTG 开发板包括以下两部分：

- **主板**: ESP32-S3-USB-OTG 主板是整个套件的核心，该主板集成了 ESP32-S3-MINI-1 模组，并提供一个与 1.3 英寸 LCD 屏连接的接口。

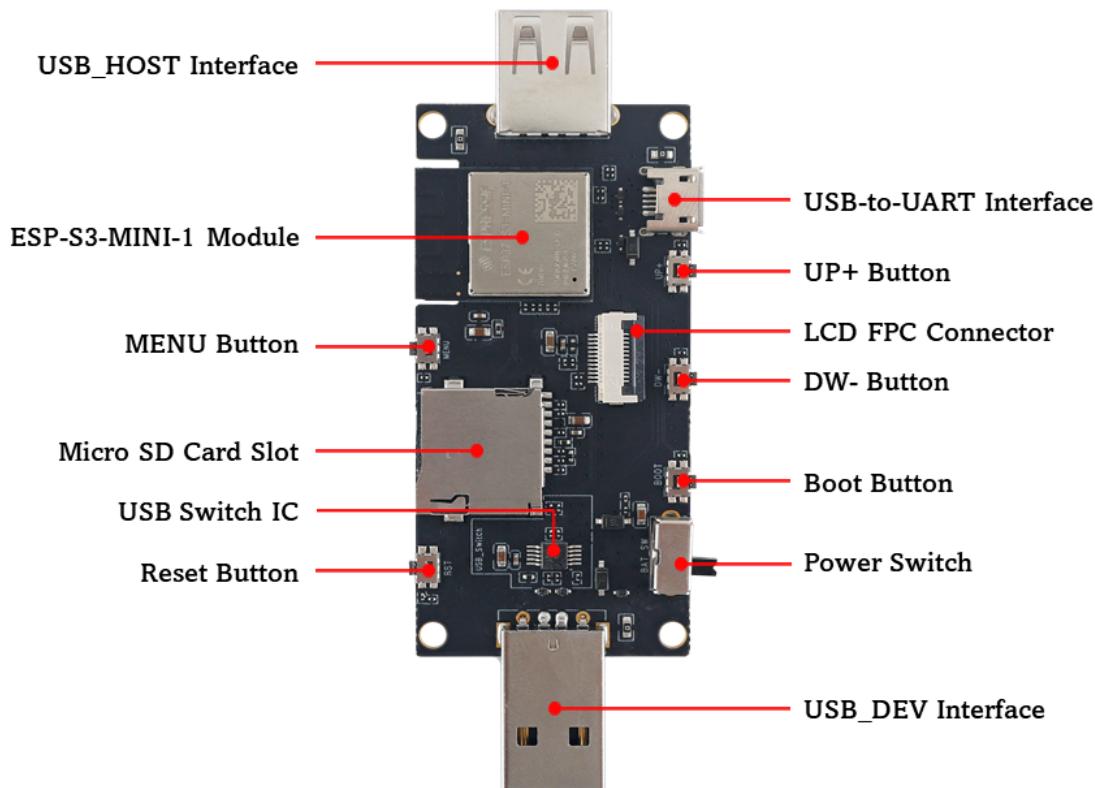


图 43: ESP32-S3-USB-OTG 正面图（点击图片放大）

下表将从左边的 USB_HOST 接口开始，以逆时针顺序介绍上图中的主要组件。

| 主要组件 | 描述 |
|--------------------|---|
| USB_HOST 接口 | USB Type-A 母口，用来连接其它 USB 设备。 |
| ESP32-S3-MINI-1 模组 | ESP32-S3-MINI-1 是通用型 Wi-Fi + 低功耗蓝牙 MCU 模组，具有丰富的外设接口、强大的神经网络运算能力和信号处理能力，专为人工智能和 AIoT 市场打造。ESP32-S3-MINI-1 采用 PCB 板载天线，与 ESP32-S2-MINI-1 pin-to-pin 兼容。 |
| MENU 按键 | 菜单按键。 |
| Micro SD 卡槽 | 可插入 Micro SD 卡。支持 4-线 SDIO 和 SPI 模式。 |
| USB Switch IC | 通过设置 USB_SEL 的电平，可以切换 USB 外设连接到 USB_DEV 或 USB_HOST 接口，默认连接到 USB_DEV。 |
| Reset 按钮 | 用于重启系统。 |
| USB_DEV 接口 | USB Type-A 公口，可连接其它 USB 主机，也作为锂电池充电。 |
| 电池供电开关 | 拨向 ON 按键一侧，使用电池供电；拨向 GND 按键一侧，通过其它方式供电。 |
| Boot 按键 | 按键 Boot 键并保持，同时按一下 Reset 键，进入“固件下载”模式，通过串口下载固件。正常使用中可以作为确认按钮。 |
| DW- 按键 | 向下按键。 |
| 屏幕排座 | 用于连接 1.3 英寸 LCD 屏。 |
| UP+ 按键 | 向上按键。 |
| USB 转 UART 接口 | Micro-USB 接口，可用作开发板的供电接口，可烧录固件至芯片，也可作为通信接口，通过板载 USB 转 UART 桥接器与芯片通信。 |

下表将从左边的 USB_HOST 接口开始，以逆时针顺序介绍上图中的主要组件。

| 主要组件 | 描述 |
|----------------|--|
| 黄色指示灯 | 设置 GPIO16 为高电平，指示灯亮。 |
| 绿色指示灯 | 设置 GPIO15 为高电平，指示灯亮。 |
| 充电指示灯 | 当为电池充电时，亮起红灯，充电完成红灯熄灭。 |
| 电池焊点 | 可焊接 3.6 V 锂电池，为主板供电。 |
| 充电电路 | 用于为锂电池充电。 |
| 空闲管脚 | 可自定义的空闲管脚。 |
| USB 转 UART 桥接器 | 单芯片 USB 至 UART 桥接器，可提供高达 3 Mbps 的传输速率。 |

- 子板：ESP32-S3-USB-OTG-SUB - 贴装 1.3 英寸 LCD 屏

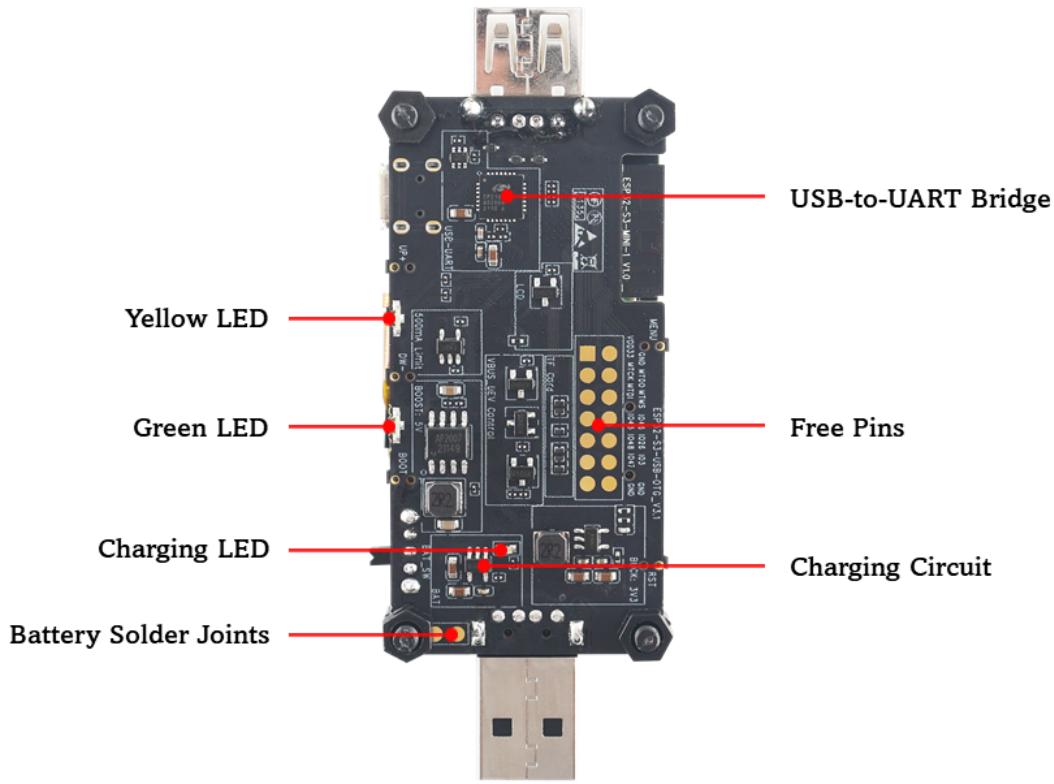


图 44: ESP32-S3-USB-OTG 背面图 (点击图片放大)

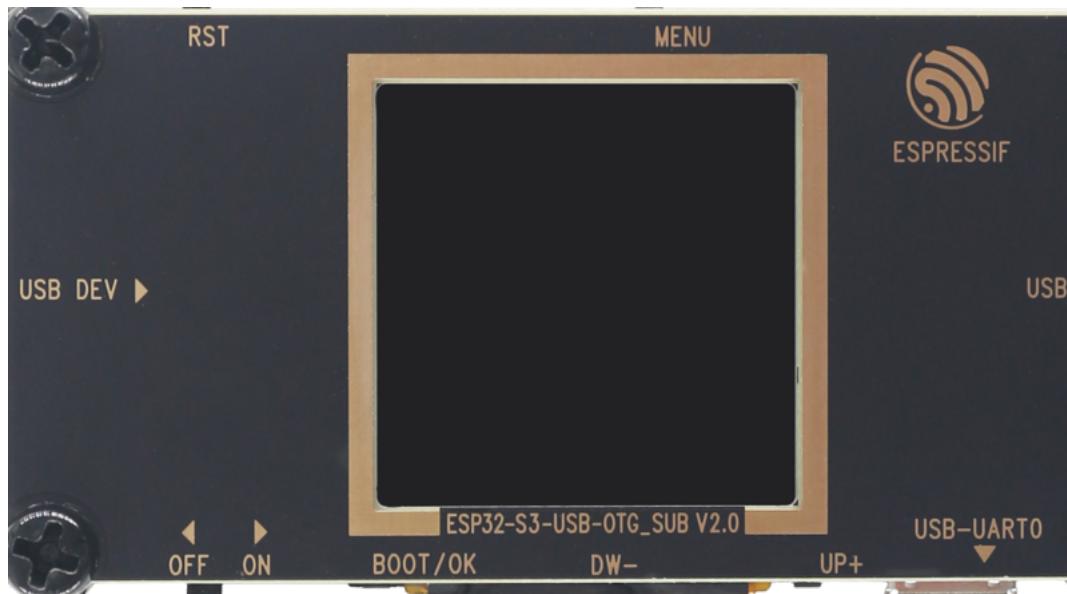


图 45: ESP32-S3-USB-OTG 子板 (点击图片放大)

应用程序开发

ESP32-S3-USB-OTG 上电前, 请首先确认开发板完好无损。

硬件准备

- ESP32-S3-USB-OTG
- 一根 USB 2.0 数据线（标准 A 转 Micro-B）
- 电脑（Windows、Linux 或 macOS）

软件设置

请前往 [快速入门](#) 中 [详细安装步骤](#) 一节查看如何快速设置开发环境。

工程选项

我们为 ESP32-S3-USB-OTG 开发板提供了应用示例, 存放在 Examples 文件夹中。

您可以在示例目录下, 输入 `idf.py menuconfig` 配置工程选项。

内容和包装

零售订单

每一个零售 ESP32-S3-USB-OTG 开发套件均有独立包装。

包含以下部分:

- 主板:
 - ESP32-S3-USB-OTG
- 子板:
 - ESP32-S3-USB-OTG_SUB
- 紧固件
 - 安装螺栓 (x4)
 - 螺丝 (x4)
 - 螺母 (x4)

零售订单请前往 <https://www.espressif.com/zh-hans/company/contact/buy-a-sample>。



图 46: ESP32-S3-USB-OTG 包装 (click to enlarge)

批量订单

如批量购买，开发板将以大纸板箱包装。

批量订单请前往 <https://www.espressif.com/zh-hans/contact-us/sales-questions>。

硬件参考

功能框图

ESP32-S3-USB-OTG 的主要组件和连接方式如下图所示。

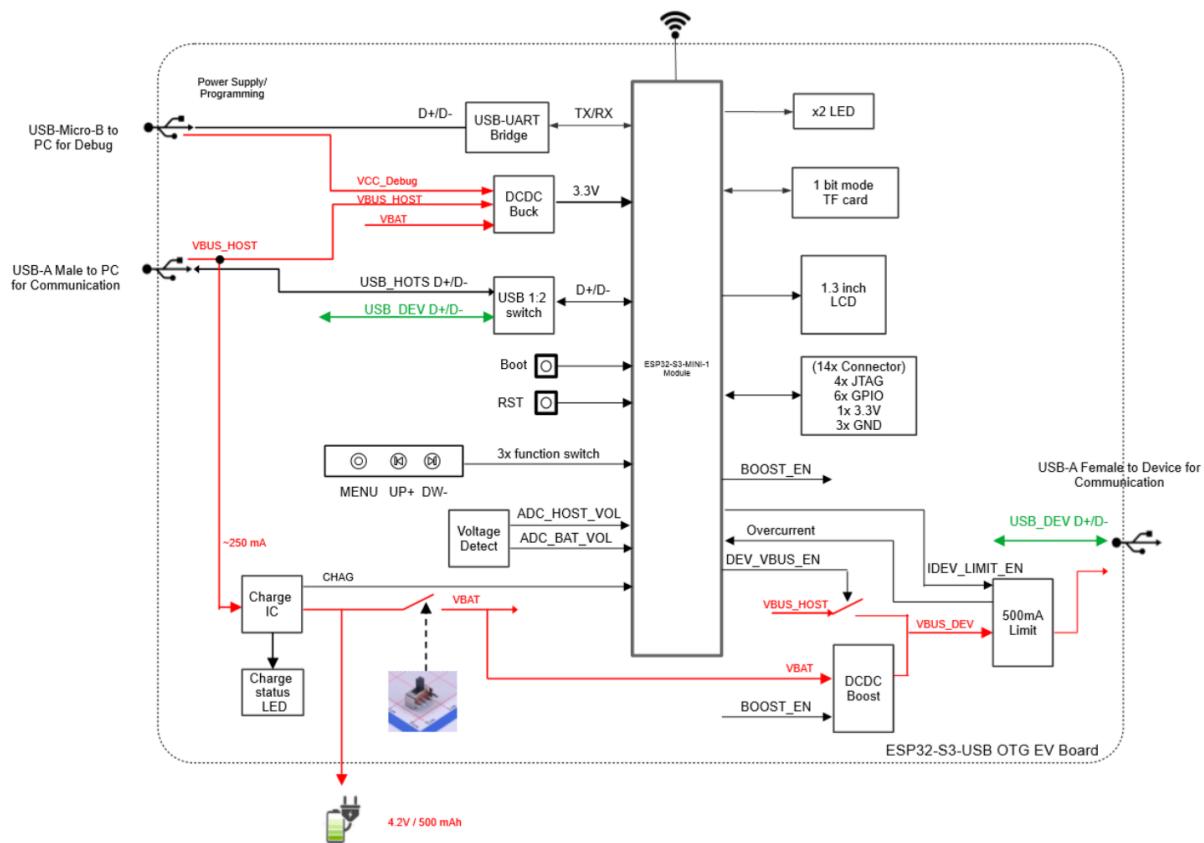


图 47: ESP32-S3-USB-OTG 功能框图 (click to enlarge)

请注意，功能框图中的 USB_HOST D+ D- 信号对应的外部接口是 USB_DEV，是指 ESP32-S3 作为设备接收其它 USB 主机的信号。USB_DEV D+ D- 信号对应的外部接口是 USB_HOST，是指 ESP32-S3 作为主机控制其它设备。

开发板供电选择

开发板有三种供电方式：

1. 通过 Micro_USB 接口供电

- 使用方法 1 供电时，应使用一根 USB 电缆（标准 A 转 Micro-B）将主控板连接至供电设备，将电源开关置于 OFF。请注意，该供电模式仅有主板和显示屏被供电。

2. 通过 USB_DEV 接口供电

- 使用方法 2 供电时，应将 DEV_VBUS_EN 设置为高电平，并将电源开关设置为 OFF。该供电模式可同时向 USB HOST 接口供电，如已安装锂电池，会同时对锂电池进行充电。

3. 通过电池接口，使用外部锂电池供电

- 使用方法 3 供电时，应将 BOOST_EN 设置为高电平，将电源开关设置为 ON，并将 1S 锂电池 (3.7 V ~ 4.2 V) 焊接于主控板背面预留的电源焊点。该供电模式可同时向 USB HOST 接口供电。电池接口说明如下图：



图 48: 电池连接图 (click to enlarge)

USB HOST 接口供电选择

USB HOST 接口 (Type-A 母口) 可向已连接的 USB 设备供电，供电电压为 5 V，最大电流为 500 mA。

- 供电电源有以下两个：
 1. 通过 USB_DEV 接口供电，5 V 电源直接来自该接口连接的电源
 2. 通过锂电池接口供电，锂电池 3.6 V ~ 4.2 V 电压通过 Boost 电路升压到 5 V。Boost IC 的工作状态可通过 BOOST_EN/GPIO13 控制，GPIO13 为高电平是使能 Boost。

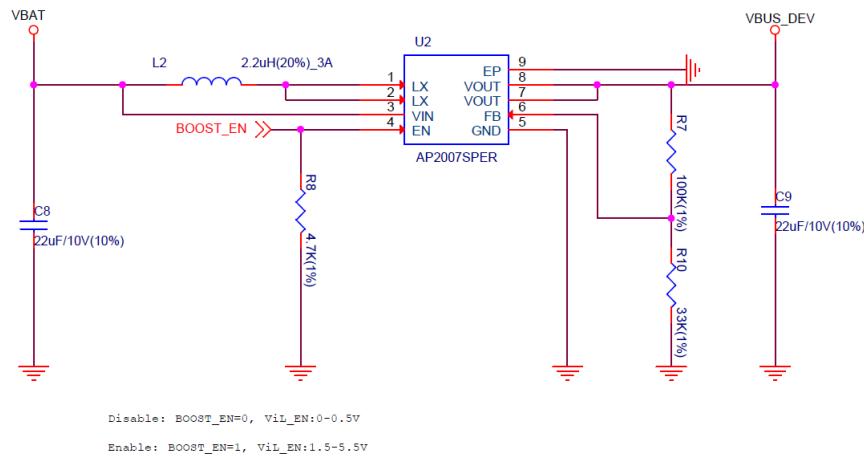


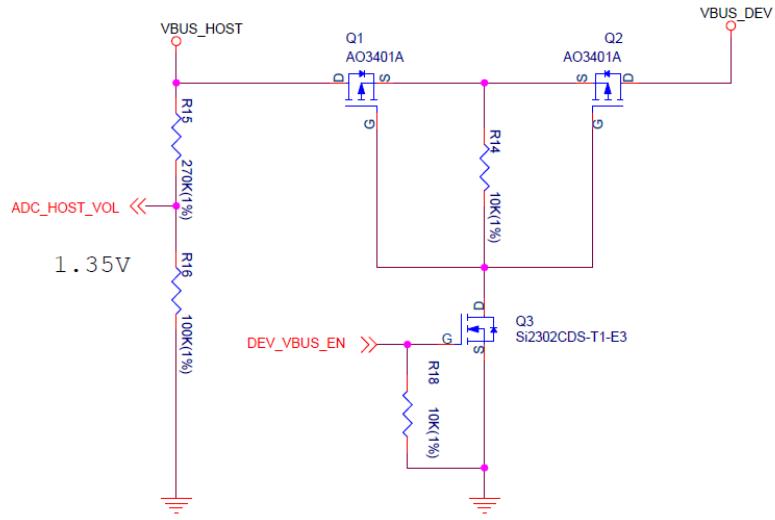
图 49: Boost 电路图 (click to enlarge)

- 供电电源选择：

| BOOST_EN | DEV_VBUS_EN | Power Source |
|----------|-------------|--------------|
| 0 | 1 | USB_DEV |
| 1 | 0 | Battery |
| 0 | 0 | No output |
| 1 | 1 | Undefined |

- 500 mA 限流电路

1. 限流 IC MIC2005A 可将 USB HOST 接口最大输出电流限制为 500 mA。必须设置 IDEV_LIMIT_EN (GPIO17) 为高电平，使能限流 IC，USB HOST 接口才有电压输出。



Switch to VBUS_HOST power mode: Step1: set BOOST_EN=0 & Step2: set DEV_VBUS_EN=1
 Switch to VBAT boost power mode: Step1: set DEV_VBUS_EN=0 & Step2: set BOOST_EN=1

图 50: 供电切换电路图

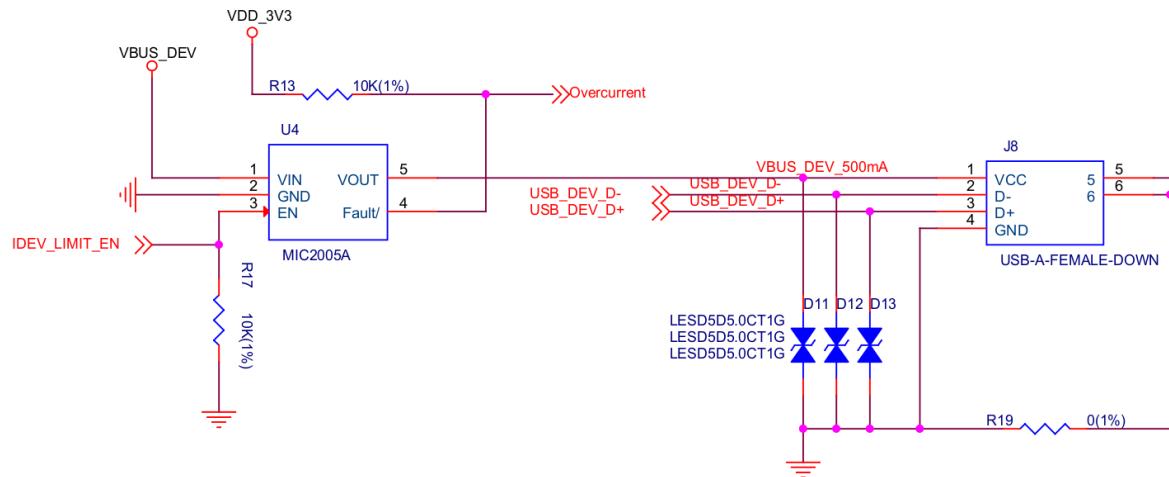


图 51: 500 mA 限流电路图

USB 接口选择电路

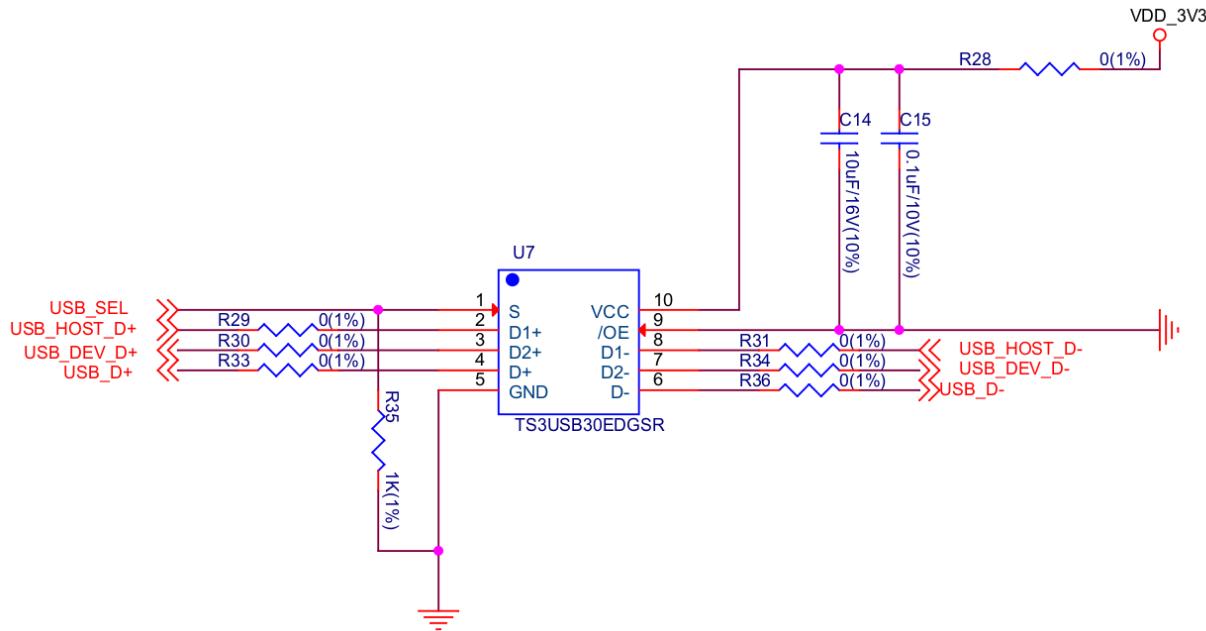


图 52: USB 接口切换电路图

- 当 **USB_SEL** (GPIO18) 引脚为高电平时, USB D+/D- (GPIO19, 20) 信号将连通到 **USB_DEV_D+ D-**, 这时可通过 USB HOST 接口 (Type-A 母口) 连接其它 USB 设备;
- 当 **USB_SEL** (GPIO18) 引脚为低电平时, USB D+/D- (GPIO19, 20) 信号将连通到 **USB_HOST_D+ D-**, 这时可通过 USB DEV 接口 (Type-A 公口) 连接到其它 USB 主机;
- 默认 **USB_SEL** 为低电平。

LCD 接口

请注意, 该接口支持连接 SPI 接口屏幕, 该开发板使用的屏幕型号为: dev-kits: ST7789 <esp32-s3-usb-otg/datasheet/ST7789VW_datasheet.pdf>, LCD_BL (GPIO9) 可用于控制屏幕背光。

SD 卡接口

请注意, SD 卡接口同时兼容 1-线, 4-线 SDIO 模式和 SPI 模式。上电后, 卡会处于 3.3 V signaling 模式下。发送第一个 CMD0 命令选择 bus 模式: SD 模式或者 SPI 模式。

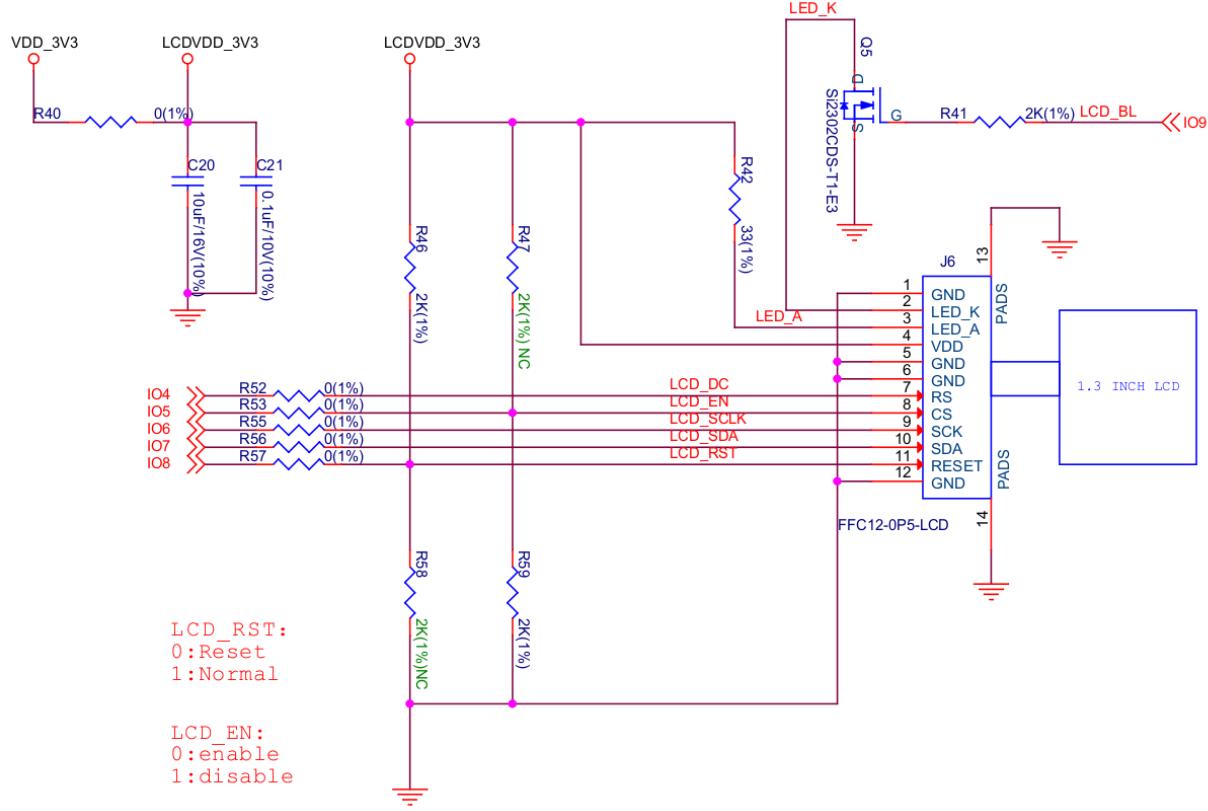


图 53: LCD 接口电路图

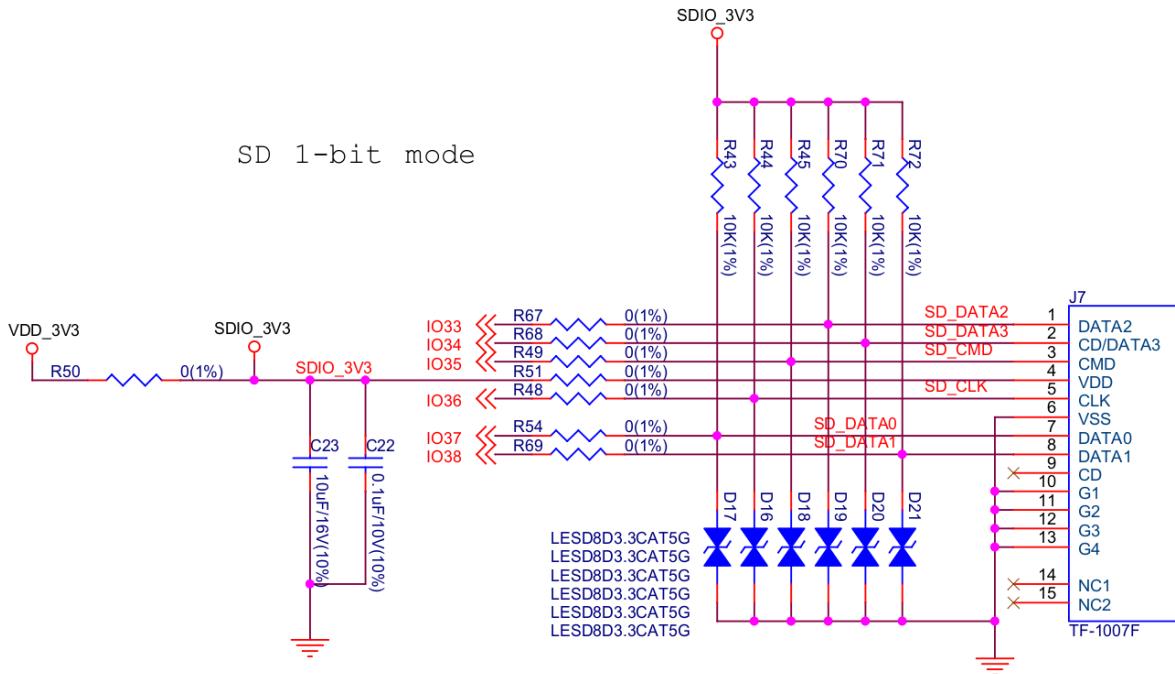


图 54: SD 卡接口电路图

充电电路

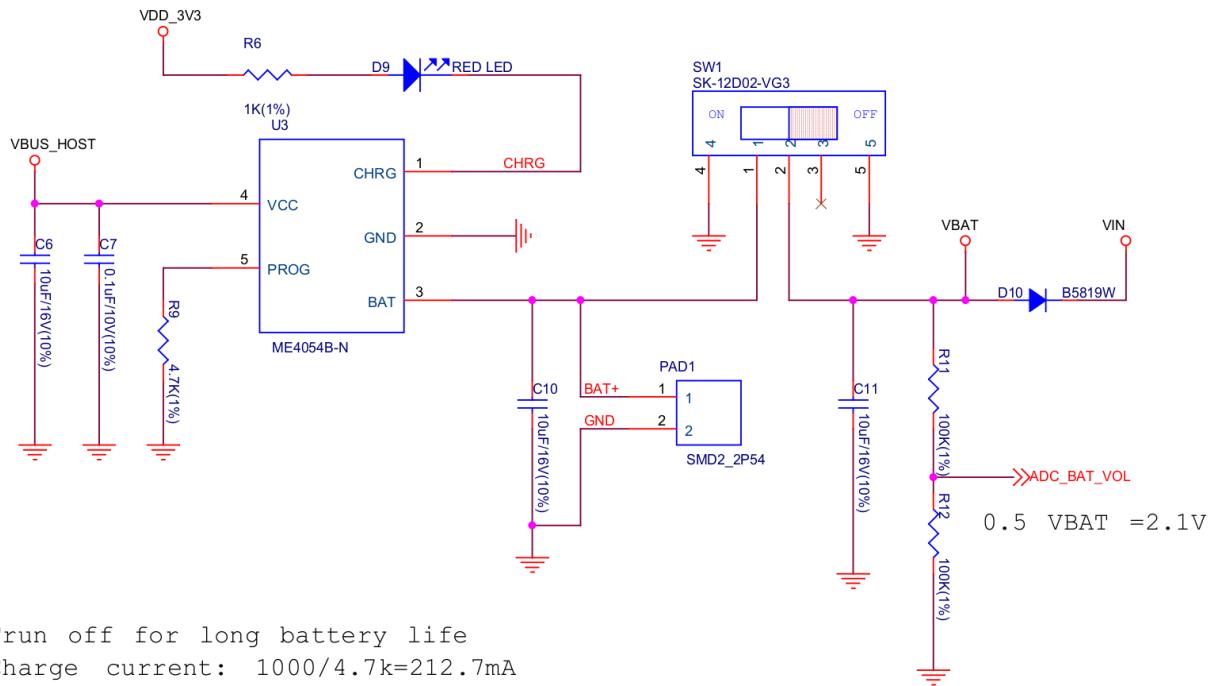


图 55: 充电电路图

请注意，可将 Type-A 公口接入输出为 5 V 的电源适配器，为电池充电时，充电电路上的红色指示灯亮，电池充电完成，红色指示灯熄灭。在使用充电电路时，请将电源开关置于 OFF。充电电流为 212.7 mA。

GPIO 分配

功能引脚：

| No. | ESP32-S3-MINI-1 管脚 | 说明 |
|-----|--------------------|---|
| 1 | GPIO18 | USB_SEL: 用于切换 USB 接口, 低电平时, USB_HOST 接口使能。高电平时, USB_DEV 接口使能。 |
| 2 | GPIO19 | 与 USB_D- 接口相连。 |
| 3 | GPIO20 | 与 USB_D+ 接口相连。 |
| 4 | GPIO15 | LED_GREEN: 用于控制绿色 LED 灯, 高电平时, 灯被点亮。 |
| 5 | GPIO16 | LED_YELLOW: 用于控制黄色 LED 灯, 高电平时, 灯被点亮。 |
| 6 | GPIO0 | BUTTON_OK: OK 按键, 按下为低电平。 |
| 7 | GPIO11 | BUTTON_DW: Down 按键, 按下为低电平。 |
| 8 | GPIO10 | BUTTON_UP: UP 按键, 按下为低电平。 |
| 9 | GPIO14 | BUTTON_MENU: MENU 按键, 按下为低电平。 |
| 10 | GPIO8 | LCD_RET: 用于复位 LCD, 低电平时复位。 |
| 11 | GPIO5 | LCD_EN: 用于使能 LCD, 低电平时使能。 |
| 12 | GPIO4 | LCD_DC: 用于切换数据和命令状态。 |
| 13 | GPIO6 | LCD_SCLK: LCD SPI 时钟信号。 |
| 14 | GPIO7 | LCD_SDA: LCD SPI MOSI 信号。 |
| 15 | GPIO9 | LCD_BL: LCD 背光控制信号。 |
| 16 | GPIO36 | SD_SCK: SD SPI CLK / SDIO CLK。 |
| 17 | GPIO37 | SD_DO: SD SPI MISO / SDIO Data0。 |
| 18 | GPIO38 | SD_D1: SDIO Data1。 |
| 19 | GPIO33 | SD_D2: SDIO Data2。 |
| 20 | GPIO34 | SD_D3: SD SPI CS / SDIO Data3。 |
| 21 | GPIO1 | HOST_VOL: USB_DEV 电压监测, ADC1 通道 0。 |
| 22 | GPIO2 | BAT_VOL: 电池电压监测, ADC1 通道 1。 |
| 23 | GPIO17 | LIMIT_EN: 使能限流芯片, 高电平使能。 |
| 24 | GPIO21 | OVER_CURRENT: 电流超限信号, 高电平代表超限。 |
| 25 | GPIO12 | DEV_VBUS_EN: 高电平选择 DEV_VBUS 电源。 |
| 26 | GPIO13 | BOOST_EN: 高电平使能 Boost 升压电路。 |

扩展功能引脚:

| No. | ESP32-S3-MINI-1 管脚 | 说明 |
|-----|--------------------|-------------------|
| 1 | GPIO45 | FREE_1: 空闲, 可自定义。 |
| 2 | GPIO46 | FREE_2: 空闲, 可自定义。 |
| 3 | GPIO48 | FREE_3: 空闲, 可自定义。 |
| 4 | GPIO26 | FREE_4: 空闲, 可自定义。 |
| 5 | GPIO47 | FREE_5: 空闲, 可自定义。 |
| 6 | GPIO3 | FREE_6: 空闲, 可自定义。 |

相关文档

- [ESP32-S3 技术规格书 \(PDF\)](#)
- [ESP32-S3-MINI-1/IU 技术规格书 \(PDF\)](#)
- [乐鑫产品选型工具](#)
- [ESP32-S3-USB-OTG 原理图 \(PDF\)](#)
- [ESP32-S3-USB-OTG PCB 布局图 \(PDF\)](#)
- [ST7789VW 规格书 \(PDF\)](#)

1.1.7 ESP32-S3-LCD-EV-Board

[English]

ESP32-S3-LCD-EV-Board 是用于评估和验证 ESP32-S3 屏幕应用的开发板，其由主板和子板构成，通过更换子板可方便接入不同的屏幕。

ESP32-S3-LCD-EV-Board

[English]

本指南将帮助您快速上手 ESP32-S3-LCD-EV-Board，并提供该款开发板的详细信息。

本指南包括如下内容：

- [开发板概述](#): 简要介绍了开发板的软件和硬件。
- [应用程序开发](#): 介绍了应用程序开发过程中的软硬件设置。
- [硬件参考](#): 详细介绍了开发板的硬件。
- [硬件版本](#): 暂无历史版本。
- [样品获取](#): 如何获取样品。
- [相关文档](#): 列出了相关文档的链接。

开发板概述

ESP32-S3-LCD-EV-Board 是一款基于 ESP32-S3 芯片的屏幕交互开发板，通过搭配不同类型的 LCD 子板，可以驱动 IIC、SPI、8080 以及 RGB 接口的 LCD 显示屏。同时它还搭载双麦克风阵列，支持语音识别和近/远场语音唤醒，具有触摸屏交互和语音交互功能，满足用户对多种不同分辨率以及接口的触摸屏应用产品的开发需求。目前支持两款开发板：搭配 480x480 LCD 的 ESP32-S3-LCD-EV-Board 和搭配 800x480 LCD 的 ESP32-S3-LCD-EV-Board-2。



图 56: 搭配 480x480 LCD 的 ESP32-S3-LCD-EV-Board



图 57: 搭配 800x480 LCD 的 ESP32-S3-LCD-EV-Board-2

特性列表

该开发板具有以下特性：

- **嵌入式模组：**板载 ESP32-S3-WROOM-1 模组，内置 16 MB flash 以及 8 MB PSRAM
- **屏幕：**可搭配不同屏幕子板使用，支持 RGB, 8080, SPI 以及 I2C 接口屏幕，请查看 *LCD 子板* 了解更多信息
- **音频：**板载音频 Codec + ADC 功放，支持双麦克风拾音
- **USB：**板载 USB 转串口芯片，并且支持 USB Type-C 接口下载调试

功能框图

ESP32-S3-LCD-EV-Board 的主要组件和连接方式如下图所示。

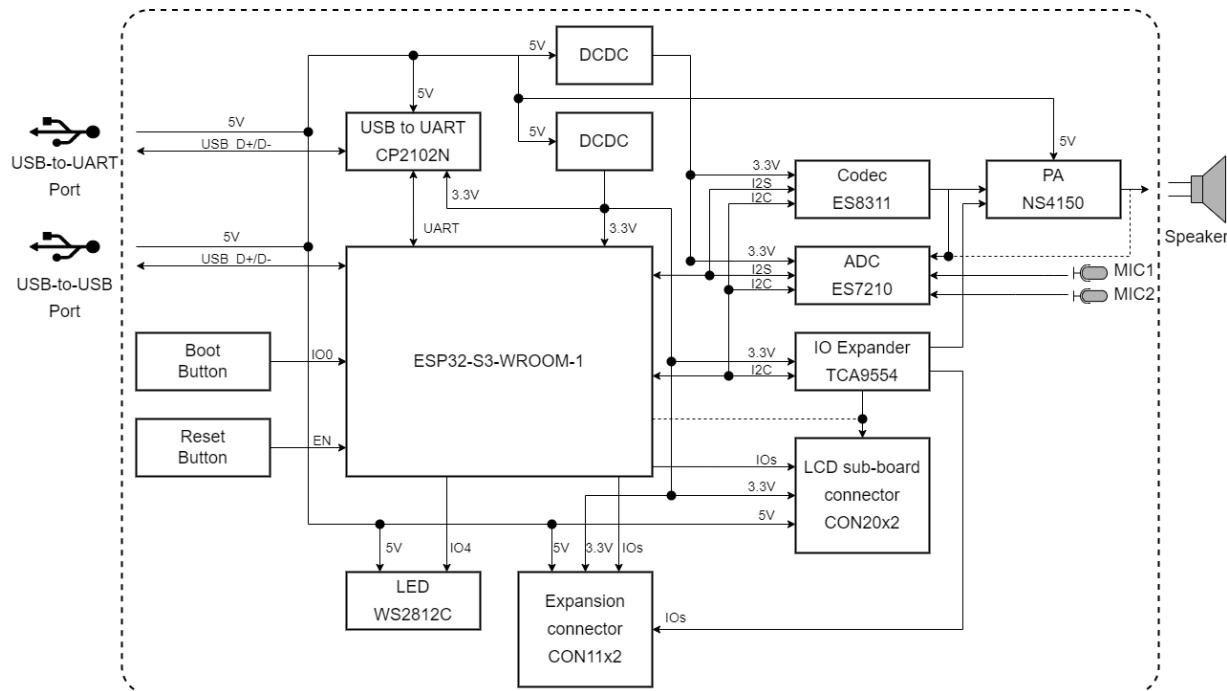


图 58: ESP32-S3-LCD-EV-Board 功能框图（点击放大）

组件介绍

ESP32-S3-LCD-EV-Board 开发板由主板和子板组成。

主板

ESP32-S3-LCD-EV-Board-MB 主板是整个套件的核心，该主板集成了 ESP32-S3-WROOM-1 模组，并提供与 LCD 子板连接的端口。

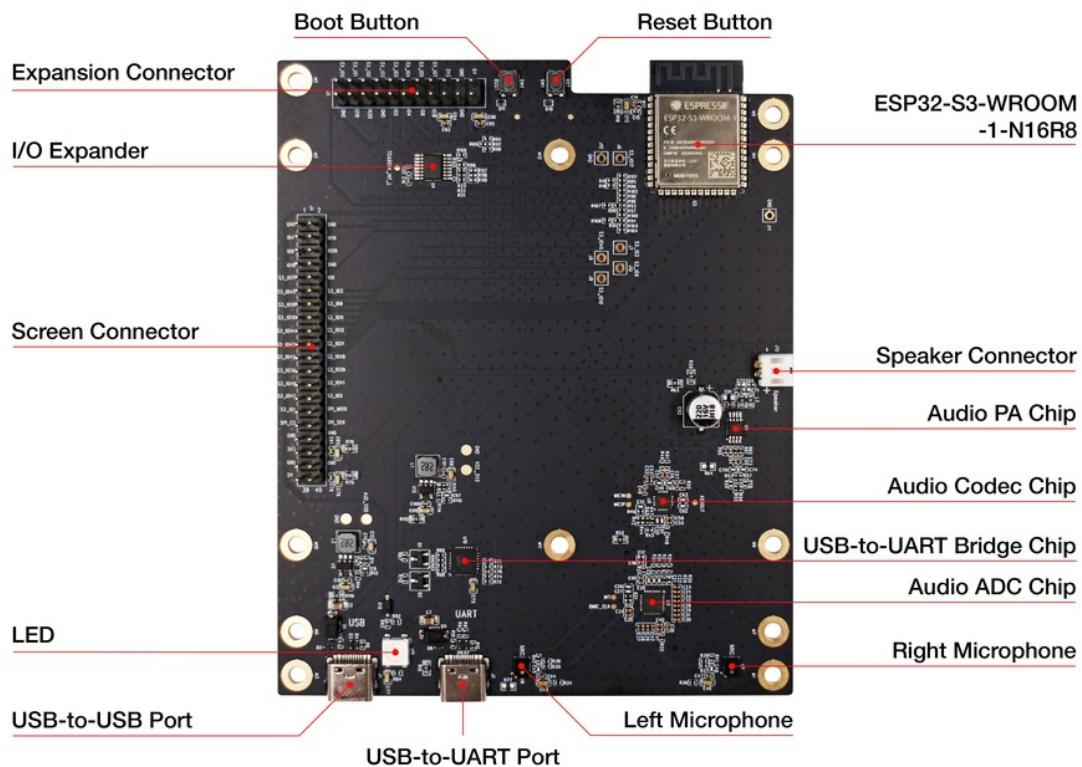


图 59: ESP32-S3-LCD-EV-Board-MB - 正面（点击放大）

以下按照逆时针顺序依次介绍开发板上的主要组件。

| 主要组件 | 介绍 |
|---------------------------|---|
| ESP32-S3-WROOM-1-N16R8 模组 | ESP32-S3-WROOM-1-N16R8 模组是一款通用型 Wi-Fi + 低功耗蓝牙 MCU 模组，搭载 ESP32-S3 系列芯片，内置 16 MB flash 以及 8MB PSRAM。除具有丰富的外设接口外，模组还拥有强大的神经网络运算能力和信号处理能力，适用于 AIoT 领域的多种应用场景。 |
| Reset 按键 | 单独按下此按键会重置系统。 |
| Boot 按键 | 长按 Boot 键时，再按 Reset 键可启动固件上传模式，然后便可通过串口或 USB 上传固件。 |
| 扩展连接器 | 可供连接所有的 IO 扩展芯片管脚、系统电源管脚以及部分模组管脚。 |
| I/O 扩展芯片 | TCA9554 是一款 8 位通用并行输入和输出 I/O 扩展芯片，它通过两线 I2C 通信控制 IO 口的模式以及输出电平，丰富了系统的 IO 应用场景。 |
| LCD 子板连接器 | 通过 2.54 mm 间距的连接器可以连接三种不同类型的 LCD 子板。 |
| LED | 具有 RGB 三色显示功能，可供用户配置用来做状态行为指示。 |
| USB-to-USB 端口 | 为整个系统提供电源（USB-to-USB 端口和 USB-to-UART 端口两者选一）。建议使用至少 5V/2A 电源适配器供电，保证供电稳定。该端口用于 PC 端与 ESP32-S3-WROOM-1 模组的 USB 通信。 |
| USB-to-UART 端口 | 为整个系统提供电源（USB-to-USB 端口和 USB-to-UART 端口两者选一）。建议使用至少 5V/2A 电源适配器供电，保证供电稳定。该端口用于 PC 端与 ESP32-S3-WROOM-1 模组的串口通信。 |
| 左侧麦克风 | 板载麦克风，连接至 ADC。 |
| 右侧麦克风 | 板载麦克风，连接至 ADC。 |
| 音频模数转换器 | ES7210 是一款用于麦克风阵列应用的高性能、低功耗 4 通道音频模数转换器，同时具备声学回声消除 (AEC) 功能，非常适合音乐和语音应用。 |
| USB-to-UART 桥接器 | 单芯片 USB-UART 桥接器 CP2102N 为软件下载和调试提供高达 3 Mbps 的传输速率。 |
| 音频编解码芯片 | ES8311 是一种低功耗单声道音频编解码器，包含单通道 ADC、单通道 DAC、低噪声前置放大器、耳机驱动器、数字音效、模拟混音和增益功能。它通过 I2S 和 I2C 总线与 ESP32-S3-WROOM-1 模组连接，以提供独立于音频应用程序的硬件音频处理。 |
| 音频功率放大器 | NS4150 是一款低 EMI、3 W 单声道 D 类音频功率放大器，用于放大来自音频编解码芯片的音频信号，以驱动扬声器。 |
| 扬声器连接器 | 可通过音频功率放大器的支持，实现外部扬声器播放功能。 |

LCD 子板

主板可搭配以下三种不同类型的子板使用：

| 子板名称 | 屏幕 (英寸) | 分辨率 (px) | LCD 驱动芯片 (接口) | 触摸驱动芯片 | 在售开发板 |
|----------------------------|---------|-----------|-----------------|---------|-------------------------|
| ESP32-S3-LCD-EV-Board-SUB1 | 0.96 | 128 x 64 | SSD1315 (I2C) | N/A | 无 |
| | 2.40 | 320 x 240 | ST7789V (SPI) | XTP2046 | 无 |
| ESP32-S3-LCD-EV-Board-SUB2 | 3.50 | 480 x 320 | ST7796S (8080) | GT911 | 无 |
| | 3.95 | 480 x 480 | GC9503CV (RGB) | FT5x06 | ESP32-S3-LCD-EV-Board |
| ESP32-S3-LCD-EV-Board-SUB3 | 4.30 | 800 x 480 | ST7262E43 (RGB) | GT1151 | ESP32-S3-LCD-EV-Board-2 |

- **ESP32-S3-LCD-EV-Board-SUB1** 子板提供了两种屏幕接口，分别支持连接一块 2.4 英寸 SPI 接口屏或者一块 0.96 英寸 I2C 接口屏。该子板暂未做适配，此处不做进一步讲解。
- **ESP32-S3-LCD-EV-Board-SUB2** 子板提供了两种屏幕接口，分别支持连接一块 RGB 接口屏或者一块 8080 并口屏。当前子板贴装了一块 3.95 英寸、RGB565 接口、分辨率为 480x480 的触摸屏，该屏使用的 LCD 驱动芯片型号为 GC9503CV，触摸驱动芯片型号为 FT5x06。
- **ESP32-S3-LCD-EV-Board-SUB3** 子板仅支持 4.3 英寸、RGB565 接口、分辨率为 800x480 的触摸屏，该屏使用的 LCD 驱动芯片型号为 ST7262E43，触摸驱动芯片型号为 GT1151。

软件支持

ESP32-S3-LCD-EV-Board 的开发框架为 [ESP-IDF](#)。ESP-IDF 是基于 FreeRTOS 的乐鑫 SoC 开发框架，具有众多组件，包括 LCD、ADC、RMT、SPI 等。开发板应用示例存放在 [Examples](#) 中，在示例目录下输入 `idf.py menuconfig` 可以配置工程选项。

注解:

- ESP-IDF 的版本要求在 v5.0 及以上，推荐使用 release/v5.0 分支开发。
- ESP32-S3 RGB 驱动仅支持 16 位 RGB565 接口和 8 位 RGB888 接口。
- 通过补丁使 ESP-IDF 支持 120 MHz 的 8 位 PSRAM 配置，请参考 [此例](#)。

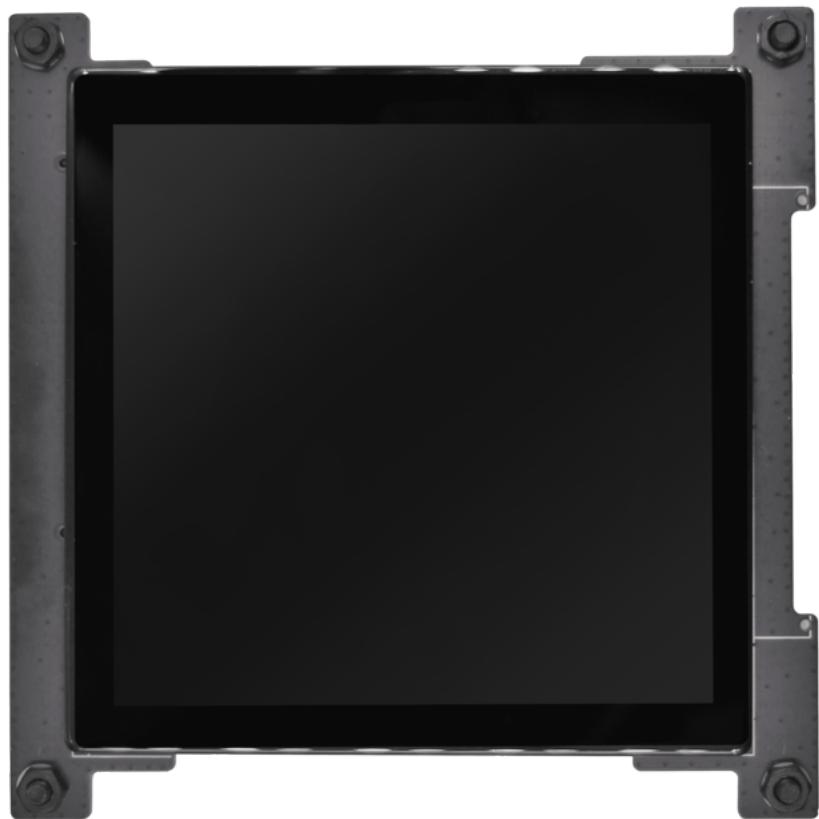


图 60: ESP32-S3-LCD-EV-Board-SUB2 - 正面 (点击放大)

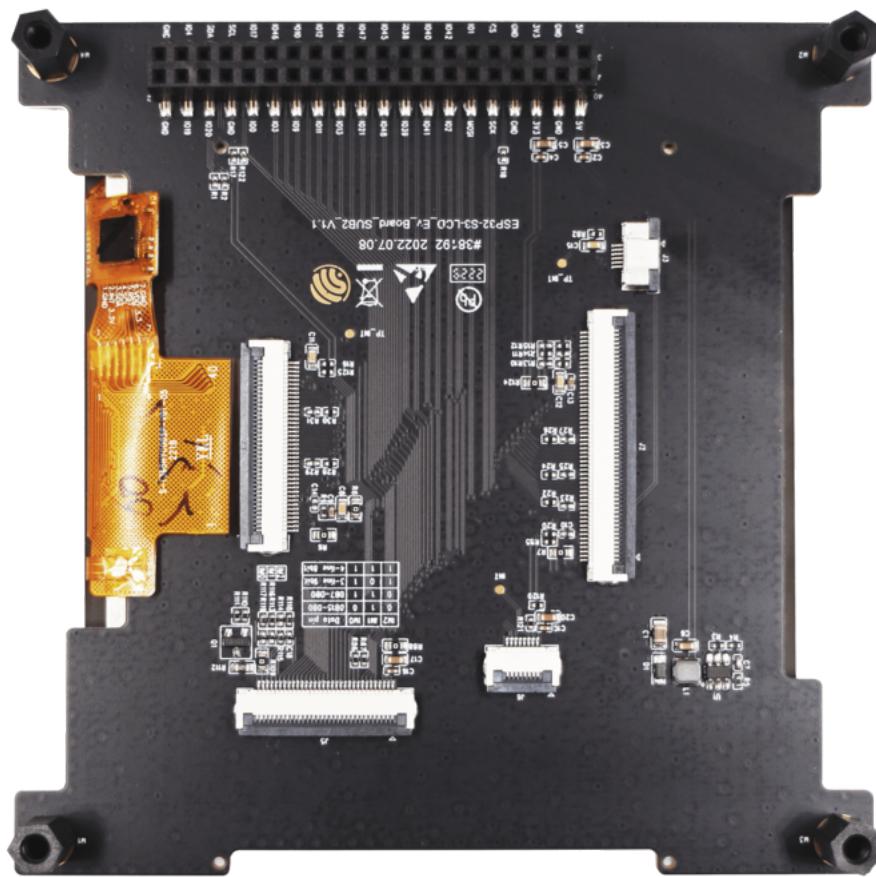


图 61: ESP32-S3-LCD-EV-Board-SUB2 - 反面 (点击放大)



图 62: ESP32-S3-LCD-EV-Board-SUB3 - 正面 (点击放大)

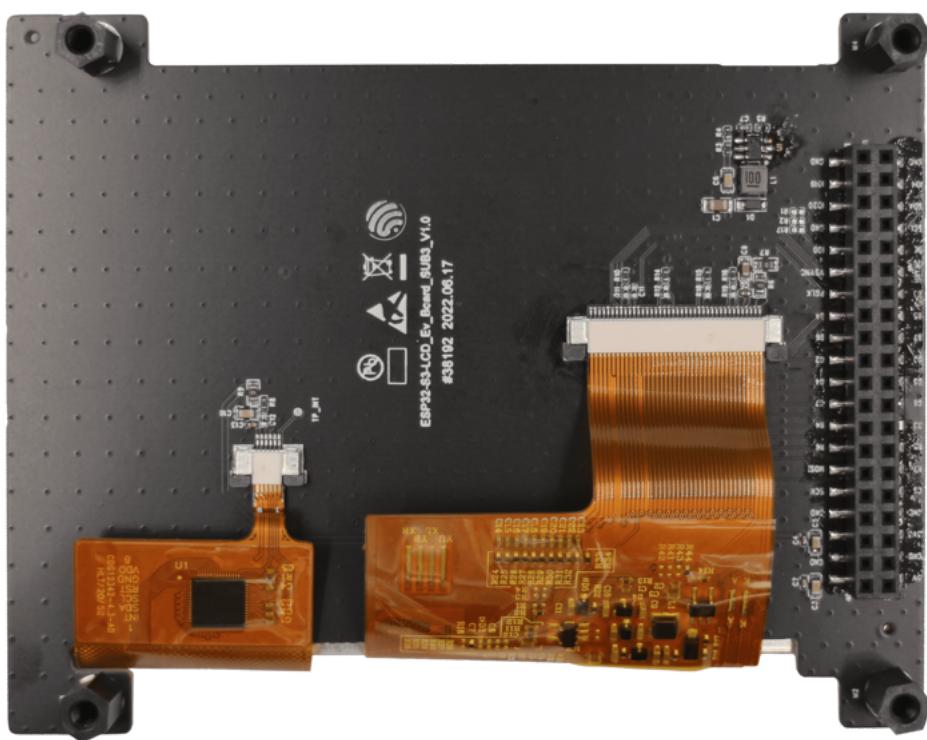


图 63: ESP32-S3-LCD-EV-Board-SUB3 - 反面 (点击放大)

应用程序开发

本节介绍硬件和软件的设置方法，以及烧录固件至开发板以开发应用程序的说明。

必备硬件

- 1 x ESP32-S3-LCD-EV-Board-MB
- 1 x LCD 子板
- 1 x USB 2.0 数据线（标准 A 型转 Type-C 型）
- 1 x 电脑（Windows、Linux 或 macOS）

注解：请确保使用适当的 USB 数据线。部分数据线仅可用于充电，无法用于数据传输和程序烧录。

可选硬件

- 1 x 扬声器

硬件设置

准备开发板，加载第一个示例应用程序：

1. 连接 LCD 子板至 **LCD 子板连接器** 端口。
2. 插入 USB 数据线，分别连接 PC 与开发板的两个 USB 端口之一。
3. LCD 屏幕亮起，可以用手指与触摸屏进行交互。

硬件设置完成，接下来可以进行软件设置。

软件设置

了解如何快速设置开发环境，请前往 [快速入门 > 安装](#)。

了解开发应用程序的更多软件信息，请查看 [软件支持](#)。

硬件参考

本节提供关于开发板硬件的更多信息。

GPIO 分配列表

下表为 ESP32-S3-WROOM-1 模组管脚的 GPIO 分配列表，用于控制开发板的特定组件或功能。

表 3: ESP32-S3-WROOM-1 GPIO 分配

| 管脚 | 管脚名称 | 功能 |
|----|------|-----------------|
| 1 | GND | 接地 |
| 2 | 3V3 | 供电 |
| 3 | EN | RESET |
| 4 | IO4 | LED |
| 5 | IO5 | I2S_MCLK |
| 6 | IO6 | I2S_CODEC_DSDIN |
| 7 | IO7 | I2S_LRCK |
| 8 | IO15 | I2S_ADC_SDOUT |
| 9 | IO16 | I2S_SCLK |
| 10 | IO17 | LCD_DE |
| 11 | IO18 | I2C_SCL |
| 12 | IO8 | I2C_SDA |
| 13 | IO19 | USB_D- |
| 14 | IO20 | USB_D+ |
| 15 | IO3 | LCD_VSYNC |
| 16 | IO46 | LCD_HSYNC |
| 17 | IO9 | LCD_PCLK |
| 18 | IO10 | LCD_DATA0 |
| 19 | IO11 | LCD_DATA1 |
| 20 | IO12 | LCD_DATA2 |
| 21 | IO13 | LCD_DATA3 |
| 22 | IO14 | LCD_DATA4 |
| 23 | IO21 | LCD_DATA5 |
| 24 | IO47 | LCD_DATA6 |
| 25 | IO48 | LCD_DATA7 |
| 26 | IO45 | LCD_DATA8 |
| 27 | IO0 | BOOT |
| 28 | IO35 | 未连接 |
| 29 | IO36 | 未连接 |

下页继续

表 3 - 续上页

| 管脚 | 管脚名称 | 功能 |
|----|------|------------|
| 30 | IO37 | 未连接 |
| 31 | IO38 | LCD_DATA9 |
| 32 | IO39 | LCD_DATA10 |
| 33 | IO40 | LCD_DATA11 |
| 34 | IO41 | LCD_DATA12 |
| 35 | IO42 | LCD_DATA13 |
| 36 | RXD0 | UART_RXD0 |
| 37 | TXD0 | UART_TXD0 |
| 38 | IO2 | LCD_DATA14 |
| 39 | IO1 | LCD_DATA15 |
| 40 | GND | 接地 |
| 41 | EPAD | 接地 |

分配给 IO 扩展芯片的 GPIO 被进一步分配为多个 GPIO。

表 4: IO 扩展芯片 GPIO 分配

| IO 扩展器管脚 | 管脚名称 | 功能 |
|----------|------|--------------|
| 1 | A0 | 接地 |
| 2 | A1 | 接地 |
| 3 | A2 | 接地 |
| 4 | P0 | PA_CTRL |
| 5 | P1 | LCD_SPI_CS |
| 6 | P2 | LCD_SPI_SCK |
| 7 | P3 | LCD_SPI_MOSI |
| 8 | GND | 接地 |
| 9 | P4 | 可做任意用途 |
| 10 | P5 | 可做任意用途 |
| 11 | P6 | 可做任意用途 |
| 12 | P7 | 可做任意用途 |
| 13 | INT | 未连接 |
| 14 | SCL | I2C_SCL |
| 15 | SDA | I2C_SDA |
| 16 | VCC | 供电电压 |

供电说明

USB 供电

开发板有两种 USB 供电方式：

- 通过 USB-to-USB 端口供电

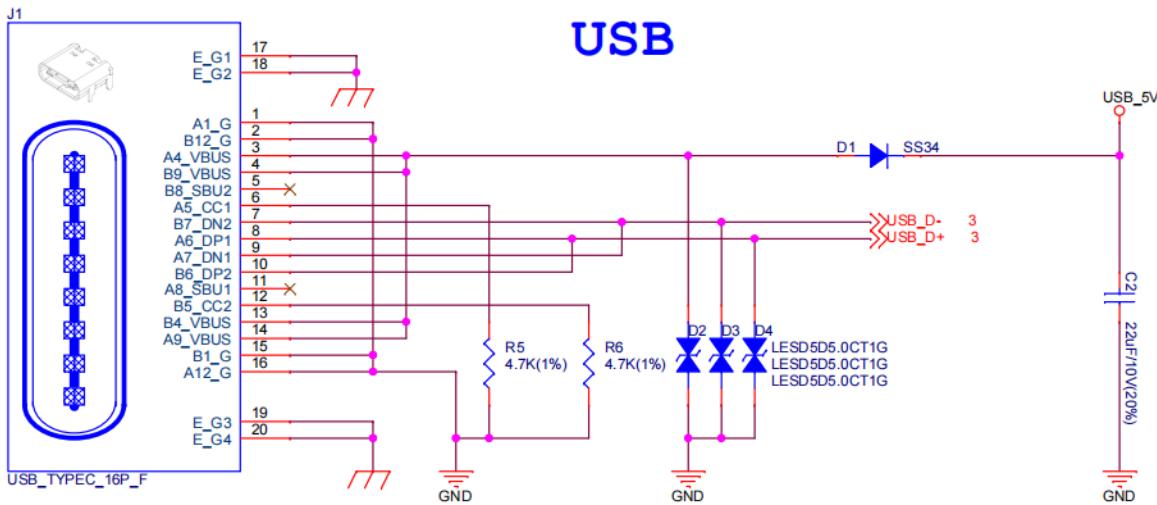


图 64: ESP32-S3-LCD-EV-Board - USB-to-USB 电源供电

- 通过 USB-to-UART 端口供电

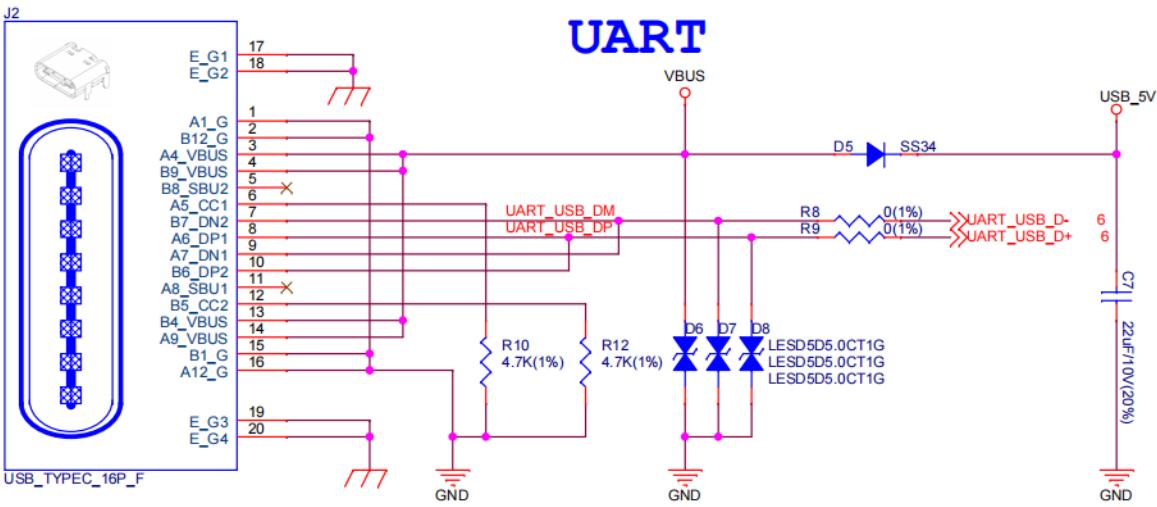


图 65: ESP32-S3-LCD-EV-Board - USB-to-UART 电源供电

音频和数字独立供电

ESP32-S3-LCD-EV-Board 可为音频组件和 ESP 模组提供相互独立的电源，可降低数字组件给音频信号带来的噪声并提高组件的整体性能。

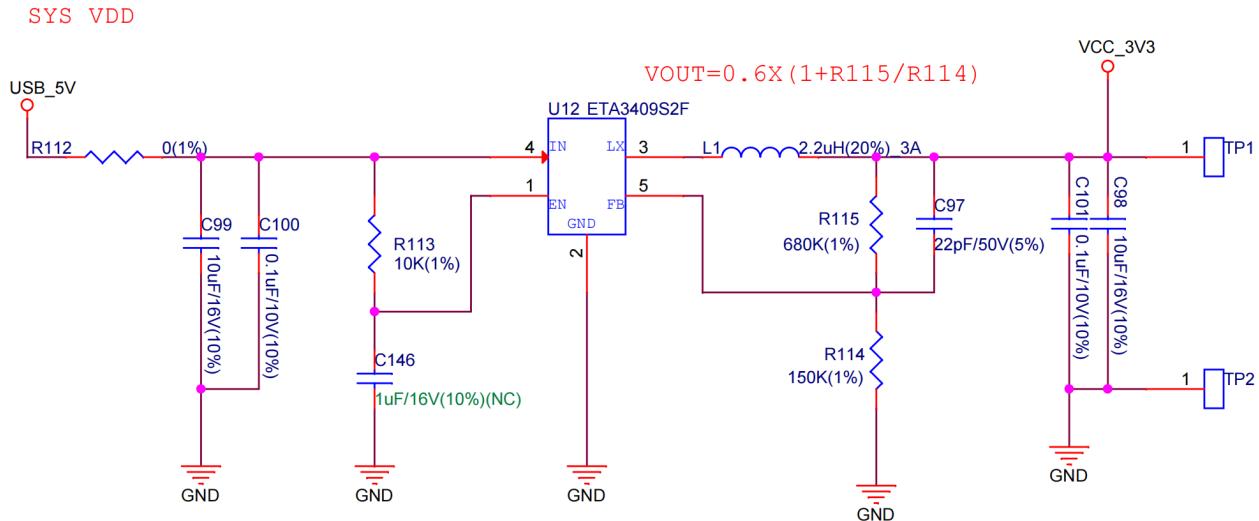


图 66: ESP32-S3-LCD-EV-Board - 数字供电

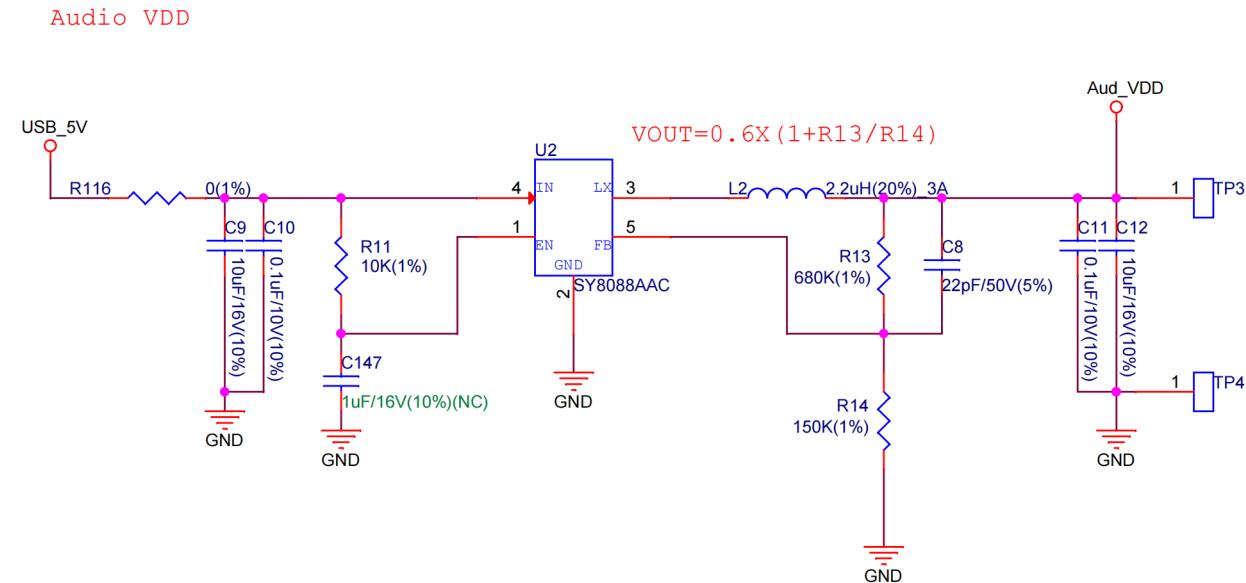


图 67: ESP32-S3-LCD-EV-Board - 音频供电

AEC 电路

AEC 电路为 AEC 算法提供参考信号。

ESP32-S3-LCD-EV-Board 回声参考信号源有两路兼容设计，一路是 Codec (ES8311) DAC 输出 (DAC_AOUTLN/DAC_AOUTLP)，一路是 PA (NS4150) 输出 (PA_OUTL+/PA_OUTL-)。推荐将默认 Codec (ES8311) DAC 输出 (DAC_AOUTLN/DAC_AOUTLP) 作为回声参考信号，下图中电阻 R54、R56 无需连接。

回声参考信号通过 ADC (ES7210) 的 ADC_MIC3P/ADC_MIC3N 采集后送回给 ESP32-S3 用于 AEC 算法。

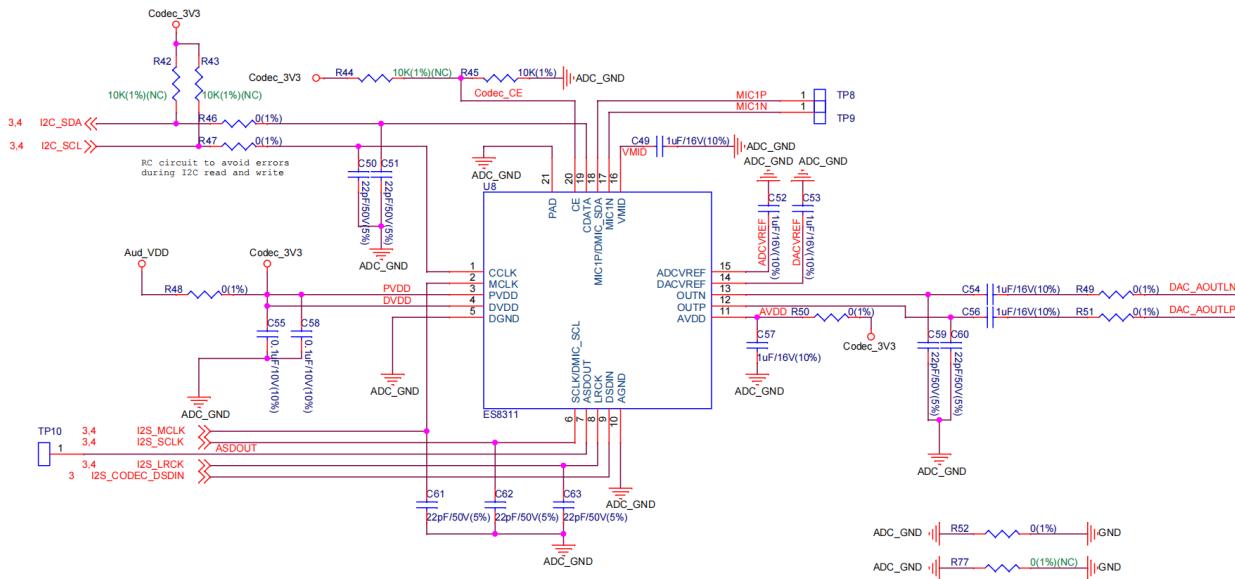


图 68: ESP32-S3-LCD-EV-Board - AEC Codec DAC 输出 (点击放大)

硬件设置选项

自动下载

可以通过两种方式使开发板进入下载模式：

- 按下 Boot 和 Reset 键，然后先松开 Reset 键，再松开 Boot 键。
- 由软件自动执行下载。软件利用串口的 DTR 和 RTS 信号来控制开发板 EN、IO0 管脚的状态。

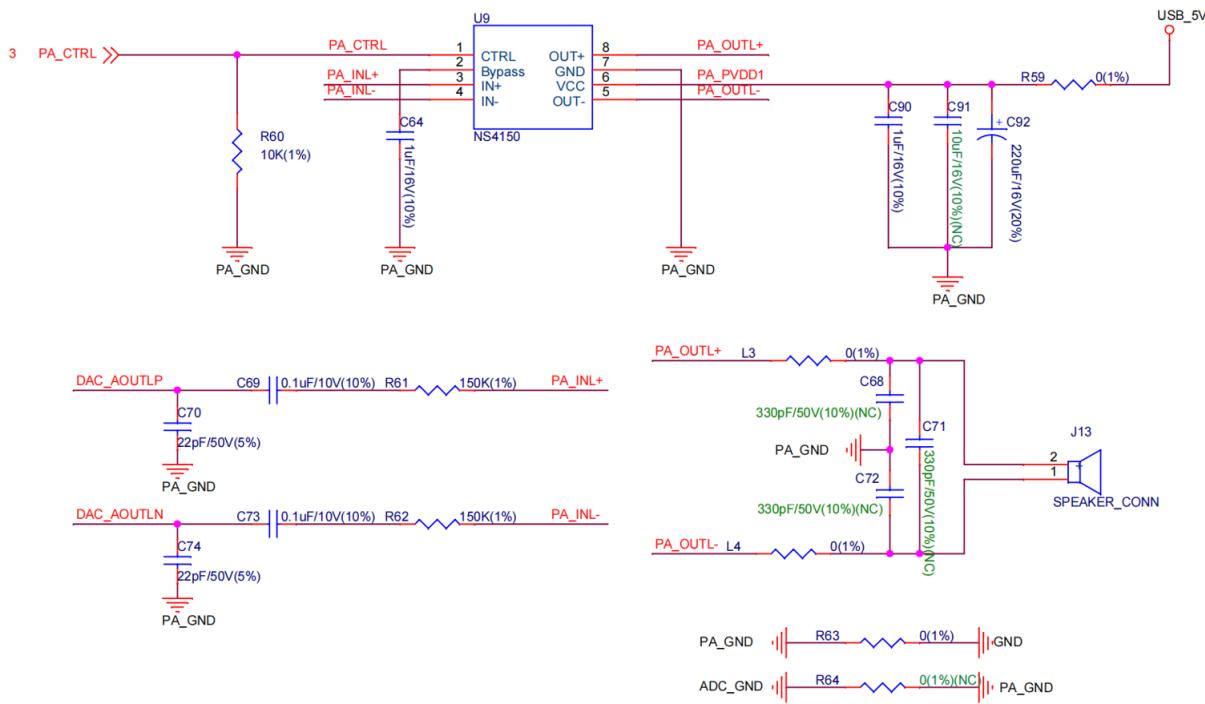


图 69: ESP32-S3-LCD-EV-Board - AEC PA 输出 (点击放大)

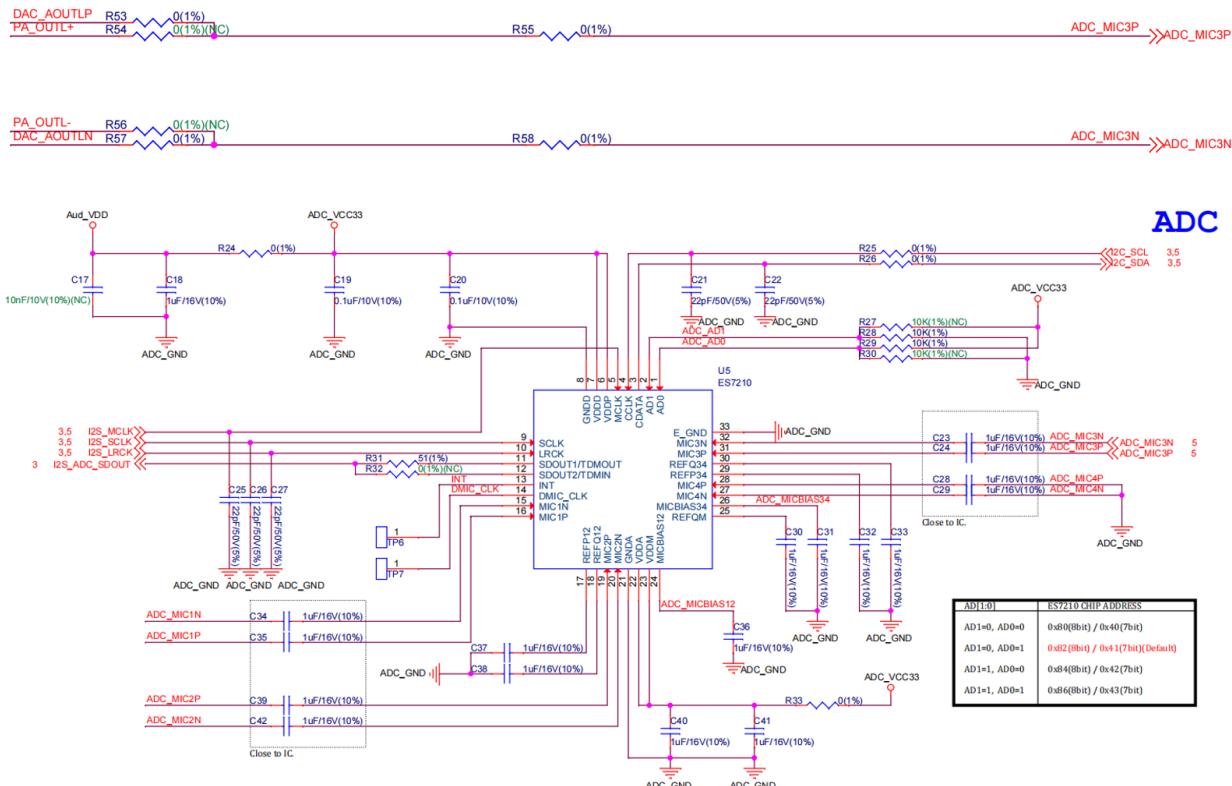


图 70: ESP32-S3-LCD-EV-Board - 参考信号采集 (点击放大)

硬件版本

无历史版本。

样品获取

此开发板适用于评估高性能的 智能屏方案。如有需要, 请前往 [乐鑫官方淘宝商城](#) 进行采购。

相关文档

- [ESP32-S3 技术规格书](#)
- [ESP32-S3-WROOM-1 技术规格书](#)
- [乐鑫产品选型工具](#)
- [ESP32-S3-LCD-EV-Board-MB 原理图](#)
- [ESP32-S3-LCD-EV-Board-MB PCB 布局图](#)
- [ESP32-S3-LCD-EV-Board-SUB1 原理图](#)
- [ESP32-S3-LCD-EV-Board-SUB1 PCB 布局图](#)
- [ESP32-S3-LCD-EV-Board-SUB2 原理图](#)
- [ESP32-S3-LCD-EV-Board-SUB2 PCB 布局图](#)
- [ESP32-S3-LCD-EV-Board-SUB3 原理图](#)
- [ESP32-S3-LCD-EV-Board-SUB3 PCB 布局图](#)
- [TCA9554 规格书](#)
- [ES7210 规格书](#)
- [ES8311 规格书](#)

有关本开发板的更多设计文档, 请联系我们的商务部门 sales@espressif.com。

1.1.8 ESP32-S2-Kaluga-1

[English]

ESP32-S2-Kaluga-1 是一款来自乐鑫的开发套件。

套件的应用示例存放在 Examples 文件夹中。

ESP32-S2-Kaluga-1

[English]

该文档最新版本请前往 [ESP32-S2-Kaluga-1 Kit](#)。

参考文档

[English]

原理图

- [ESP32-S2-Kaluga-1 v1.2 主板](#)
- [ESP-LyraT-8311A v1.2 音频板](#)
- [ESP-LyraP-LCD32 v1.2 显示屏](#)
- [ESP-LyraP-TOUCHA v1.2 触摸板](#)
- [ESP-LyraP-CAM v1.2 摄像头](#)

规格书

- [《ESP32-S2 技术规格书》](#)
- [LCD ST7789](#)
- [Camera OV2640](#)
- [Audio ES8311](#)

1.1.9 ESP32-LCDKit

[English]

ESP32-LCDKit 是一款以乐鑫 ESP32-DevKitC（需另采购）为核心的 HMI（人机交互）开发板。

ESP32-LCDKit

[English]

概述

ESP32-LCDKit 是一款以乐鑫 ESP32-DevKitC (需另采购) 为核心的 HMI (人机交互) 开发板，可外接屏幕，并且集成了 SD-Card、DAC-Audio 等外设，主要用于 HMI 相关开发与评估。开发板预留屏幕接口类型：SPI 串行接口、8 位并行接口、16 位并行接口。

HMI Example 这里提供的一些 HMI 示例可在 ESP32-LCDKit 开发板上运行。

关于 ESP32 详细信息，请参考文档《ESP32 系列芯片技术规格书》。

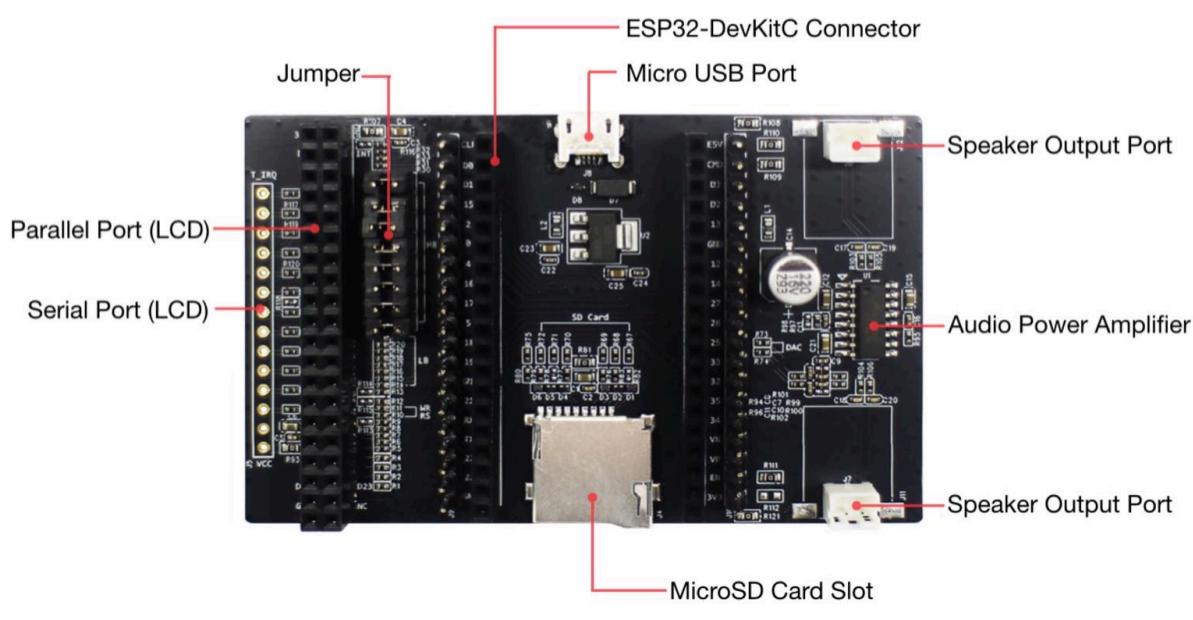


图 71: ESP32-LCDKit

电路设计说明

系统框图

ESP32-LCDKit 开发板的系统框图如下所示：

PCB 布局

ESP32-LCDKit 开发板的 PCB 布局如下所示：

PCB 部件功能说明如下表所示：

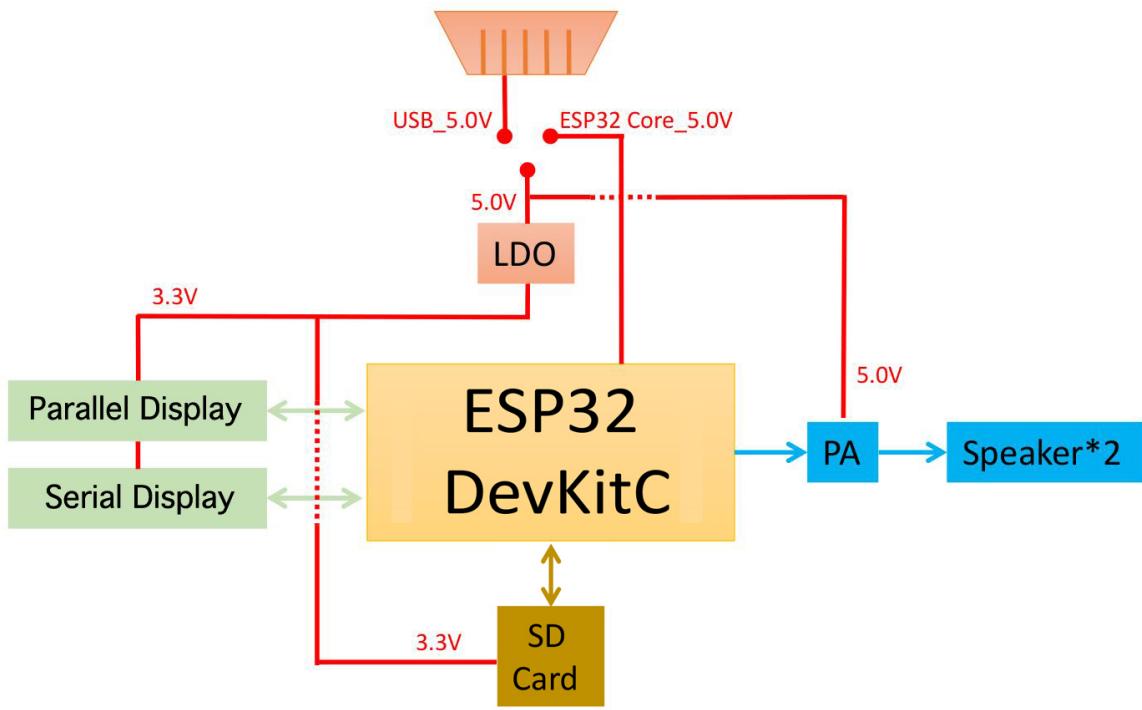


图 72: ESP32-LCDKit 系统框图

| PCB 部件 | 说明 |
|-----------------------|---------------------------|
| 屏幕连接模块 | 连接串行或者并行（8/16 bit）屏幕 |
| ESP32 DevKitC 开发板连接模块 | 与 ESP32 DevKitC 开发板连接 |
| SD-Card 模块 | 连接 SD-Card，扩展存储 |
| DAC-Audio 模块 | 连接喇叭播放音频，包含音频功率放大器 |
| 数据位选择跳冒 | 用于选择所用并行屏幕的数据位宽（8/16 bit） |

硬件模块

本章主要介绍各个功能模块（接口）的硬件实现，以及对这些模块的描述。

- 原理图
- PCB Layout

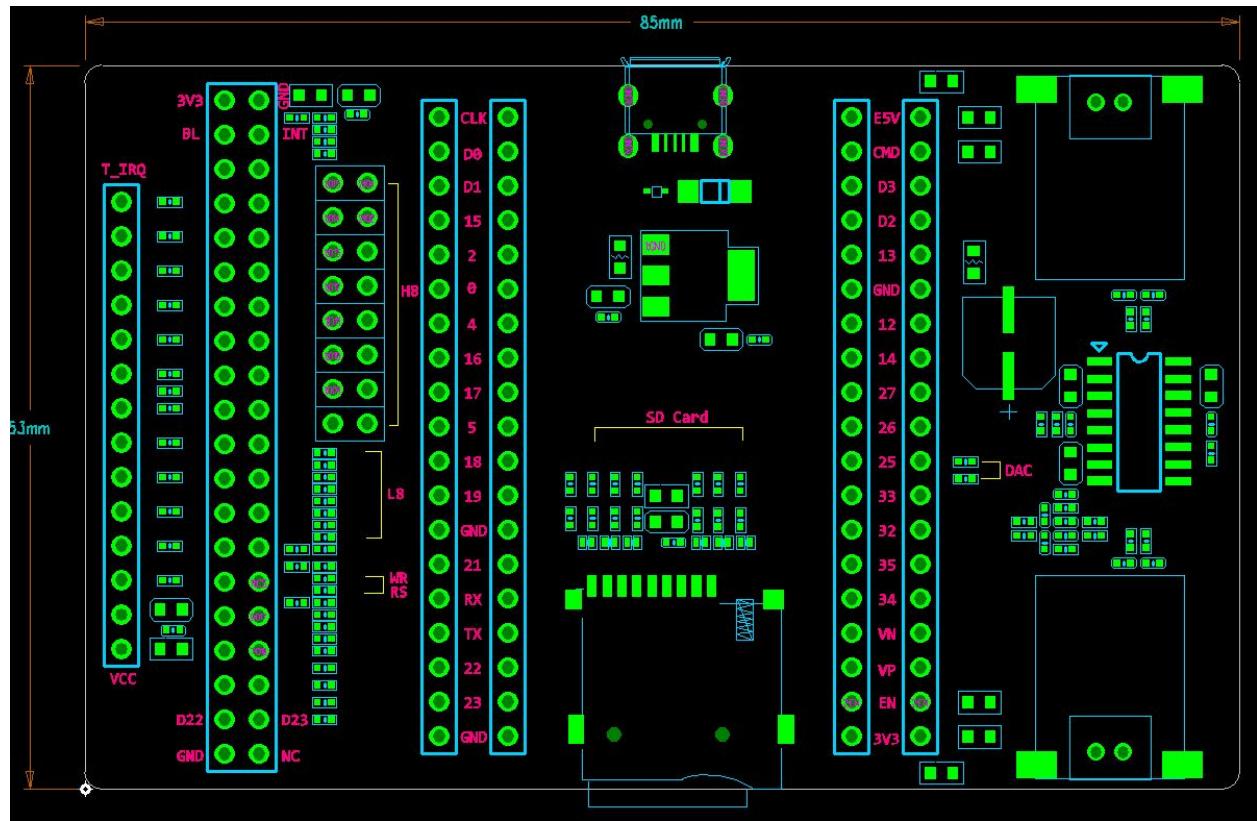


图 73: ESP32-LCDKit PCB 布局

ESP32 DevKitC 开发板连接模块

使用 ESP32-LCDKit 开发板进行 HMI 相关开发时需要搭配使用 ESP32 DevKitC 开发板。

ESP32-LCDKit 开发板与 ESP32 DevKitC 开发板连接模块电路原理图如下所示：

Core Board Connector:

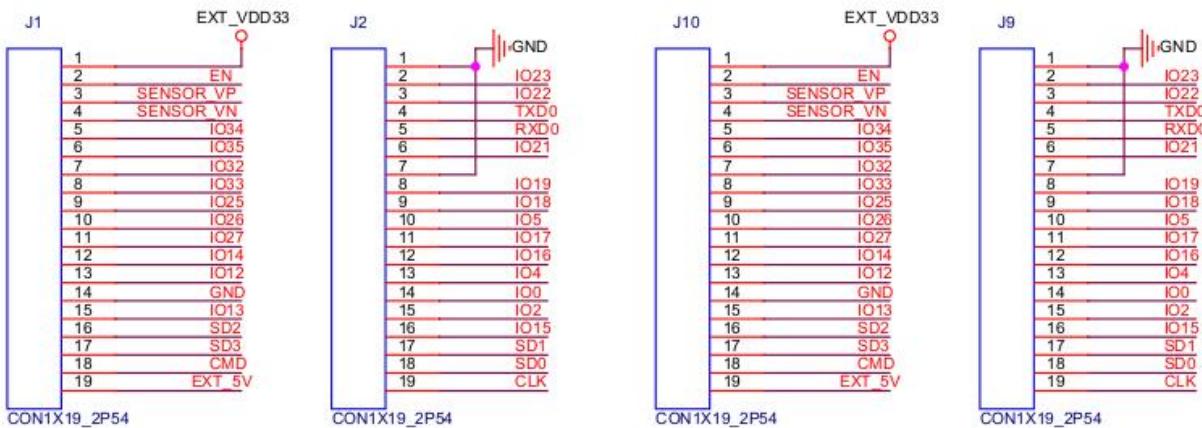


图 74: ESP32 DevKitC 连接模块

电源管理

USB 供电管理模块电路图如下所示：

屏幕连接模块

以下接口可供选择使用：

- SPI 串行接口
- 8 位并行接口
- 16 位并行接口

开发板采用两种不同的连接插座，可以将不同屏幕接至 ESP32-LCDKit 开发板上，以实现 ESP32 模组对屏幕的操作。

外接屏幕电路原理图如下所示：

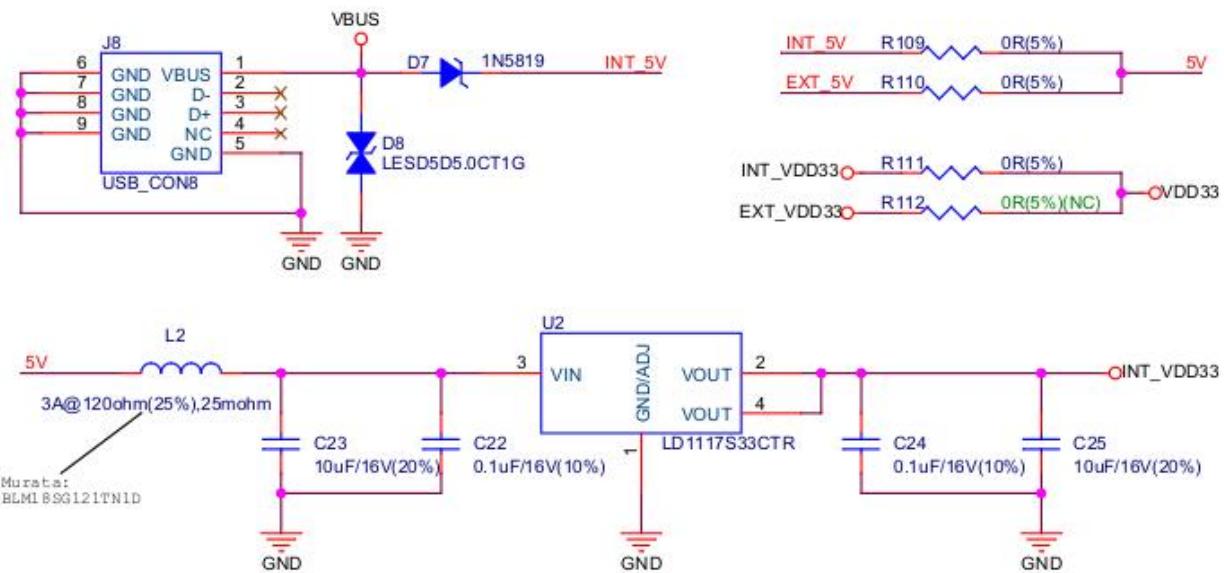
Power:

图 75: ESP32-LCDKit 供电管理模块

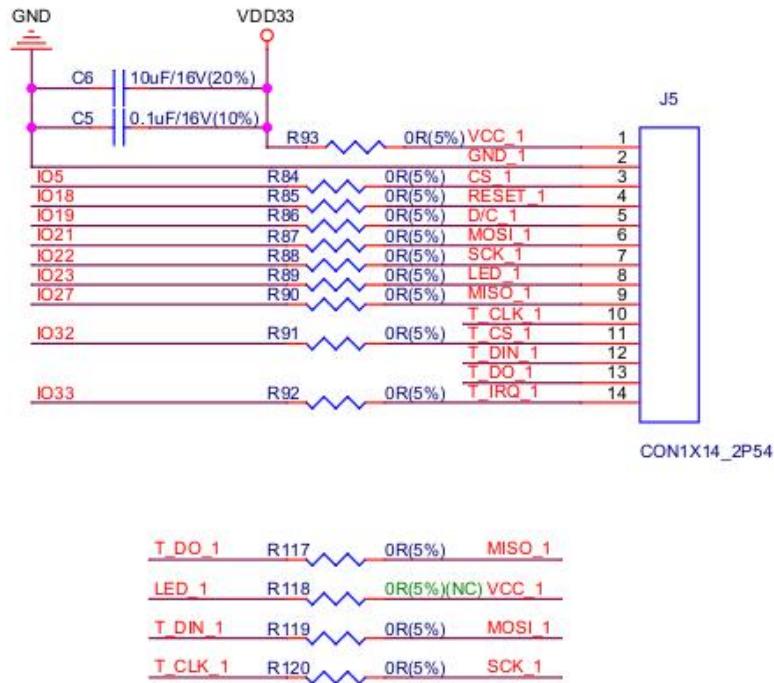
Serial Screen Connector:

图 76: ESP32-LCDKit 外接屏幕模块

SD-Card、DAC-Audio 外设模块

开发板支持外接 SD-Card 以扩展存储，并且有 MIX3006 功率放大器，可以外接喇叭播放音频。

SD-Card、DAC-Audio 外设模块电路原理图如下所示：

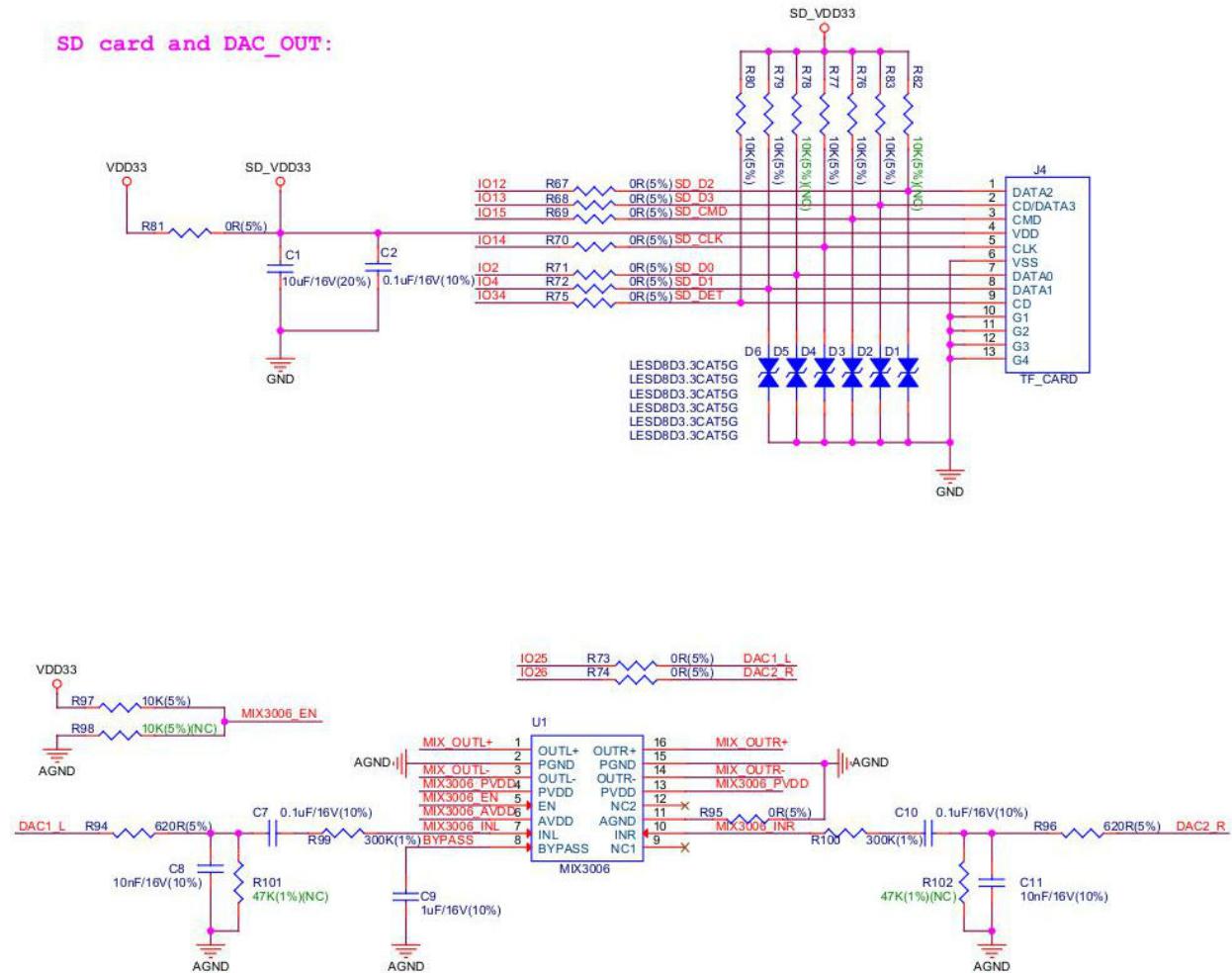


图 77: SD-Card 与 DAC-Audio 外设

相关文档

- ESP32-LCDKit 原理图
- ESP32-LCDKit PCB 布局

1.1.10 ESP-Prog

[English]

ESP-Prog 是一款乐鑫推出的开发调试工具，具有自动下载固件、串口通信、JTAG 在线调试等功能。

ESP-Prog

[English]

本指南将帮助您快速上手 ESP-Prog，并提供该款开发板的详细信息。

ESP-Prog 是一款乐鑫推出的开发调试工具，具有自动下载固件、串口通信、JTAG 在线调试等功能。自动下载固件和串口通信功能适用于 ESP8266、ESP32、ESP32-S2、ESP32-S3 和 ESP32-C3；JTAG 在线调试功能适用于 ESP32、ESP32-S2、ESP32-S3 和 ESP32-C3。

ESP-Prog 使用简单方便，仅用一根 USB 线即可实现与电脑的连接。电脑端可以通过端口号识别出下载功能和 JTAG 功能对应的两个端口。

考虑到不同用户板的电源电压可能不同，ESP-Prog 的两个接口均可通过排针选择 5 V 或者 3.3 V 供电，具备较强的电源兼容性。不过，尽管 ESP-Prog 的电源可以在 3.3 V 和 5 V 之间切换，但 RX/TX 和 JTAG 信号将恒定在 3.3 V。

本指南包括如下内容：

- 入门指南：简要介绍了开发板和硬件、软件设置指南。
- 硬件参考：详细介绍了开发板的硬件。
- 相关文档：列出了相关文档的链接。

入门指南

本小节将简要介绍 ESP-Prog，说明如何在 ESP-Prog 进行初始硬件设置。

组件介绍

以下按照顺时针的顺序依次介绍开发板上的主要组件。

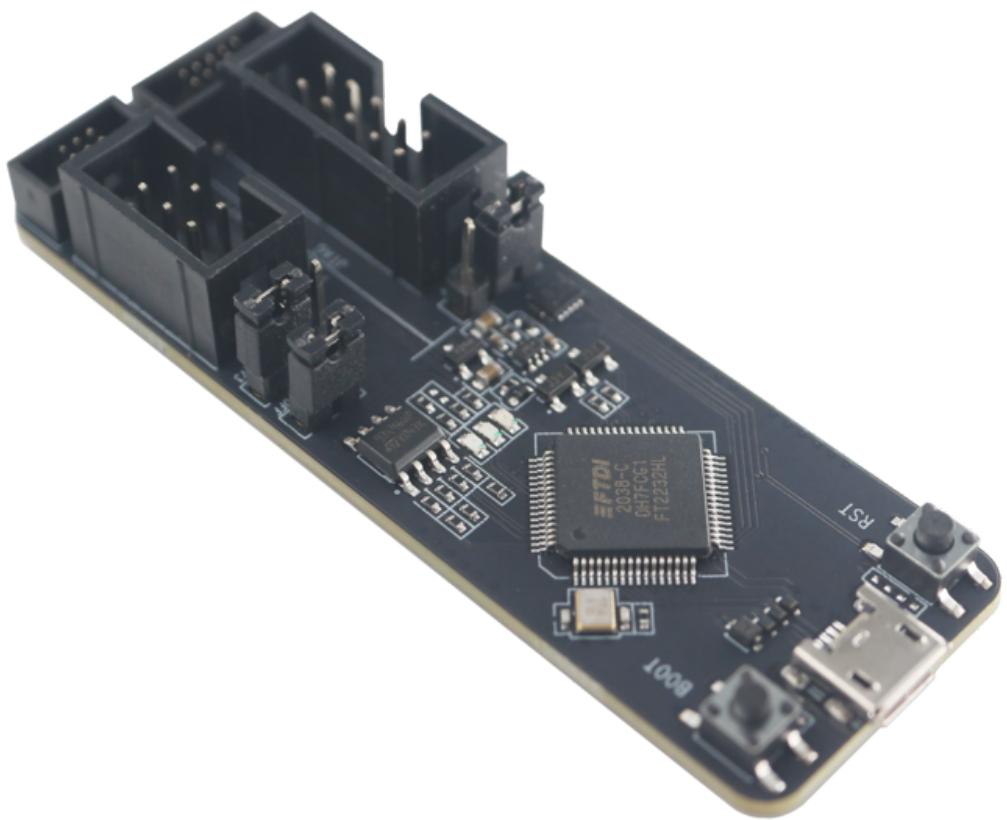


图 78: ESP-Prog (点击放大)

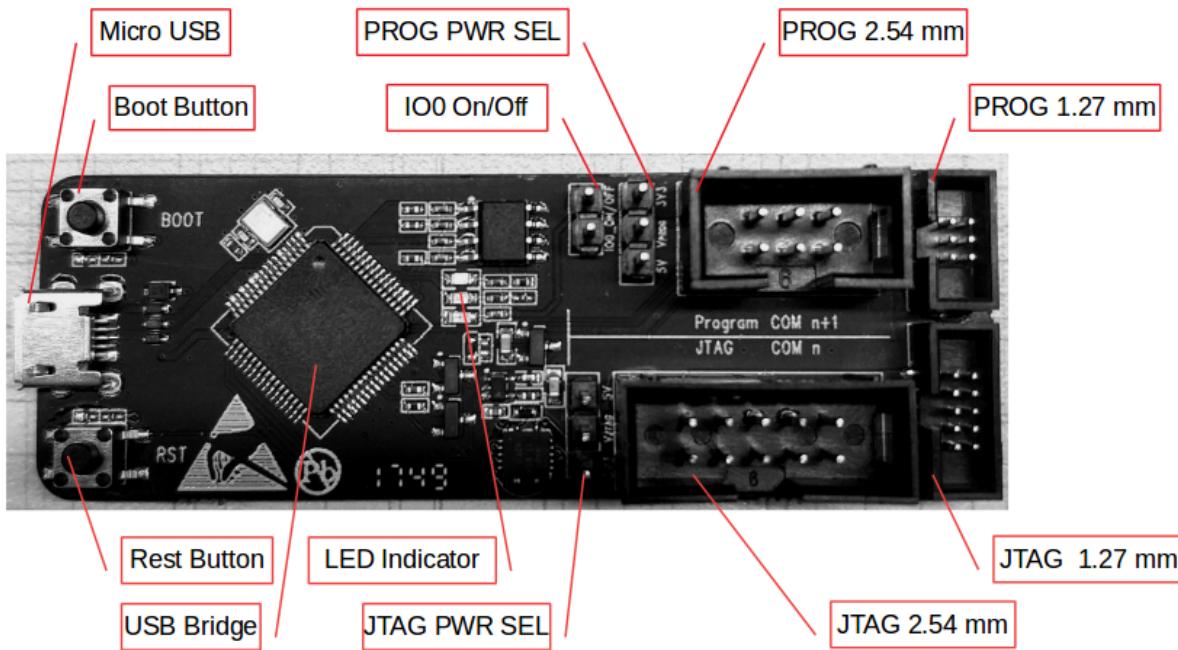


图 79: ESP-Prog - 正面 (点击放大)

| 主要组件 | 介绍 |
|--------------|---|
| Micro USB | 电脑端与 ESP-Prog 的接口。 |
| Boot 按键 | 下载按键。按下 Boot 键并保持，同时按一下 Reset 键，进入“固件下载”模式，通过串口下载固件。正常使用中可以作为确认按钮。 |
| IO0 On/Off | 用于配置 GPIO0 Strapping 管脚状态的排针。 |
| PROG PWR SEL | 用于选择 Program 接口电源输入的排针。 |
| PROG 2.54 mm | 管脚间距为 2.54 mm (0.1") 的 Program 接口。 |
| PROG 1.27 mm | 管脚间距为 1.27 mm (0.05") 的 Program 接口。 |
| JTAG 1.27 mm | 管脚间距为 1.27 mm (0.05") 的 JTAG 接口。 |
| JTAG 2.54 mm | 管脚间距为 2.54 mm (0.1") 的 JTAG 接口。 |
| JTAG PWR SEL | 用于选择 JTAG 接口电源输入的排针。 |
| LED 指示灯 | 显示 ESP-Prog 的状态。共有三种 LED 模式：红色、绿色和蓝色。当系统的 3.3 V 电源通电时，红色 LED 灯亮起；当下载板发送数据时，绿色 LED 灯亮起；当下载板接收数据时，蓝色 LED 灯亮起。 |
| USB 桥接器 | 单芯片 USB 至 UART 桥接器，可提供高达 3 Mbps 的传输速率。 |
| Reset 按钮 | 用于重启系统。 |

开始开发应用

通电前，请确保开发板完好无损。

必备硬件

- ESP-Prog
- 一根 USB 2.0 数据线（标准 A 转 Micro-B）
- 装有 Windows、macOS 或 Linux 系统的电脑
- 杜邦线或是乐鑫提供的排线，用于连接开发板和 ESP-Prog

注解：请确保使用适当的 USB 数据线。部分数据线仅可用于充电，无法用于数据传输和编程。

硬件设置

1. 通过 USB 线连接 ESP-Prog 调试板和电脑端的 USB 口。
2. 在电脑端安装 [FT2232HL 芯片驱动](#)。若电脑端识别到两个端口，则表明驱动已安装成功。
3. 使用 PROG PWR SEL/JTAG PWR SEL 排针选择 Program/JTAG 接口上的电源输出电压。
4. 使用乐鑫提供的排线连接 ESP-Prog 调试板和 ESP 产品板。
5. 使用乐鑫官方软件工具或脚本，实现自动下载和 JTAG 调试功能。

软件设置

请前往 [ESP-IDF 快速入门](#) 中 [详细安装步骤](#) 一节查看如何快速设置开发环境。

内容和包装

零售订单

每一个零售 ESP-Prog 开发板均有独立包装。

包含以下部分：

- 开发板
- ESP-Prog
- 数据线



图 80: ESP-Prog 包装内容

两根排线:

- 一根用于 PROG 2.54 mm 接口，连接 2*5-PIN 的牛角座。
- 一根用于 PROG 1.27 mm 接口，连接 2*3-PIN 的牛角座。

如果您订购了一批样品，根据零售商的不同，每块板子的独立包装会有所差异。

零售订单请前往 <https://www.espressif.com/zh-hans/company/contact/buy-a-sample>。

批量订单

If you order in bulk, the boards come in large cardboard boxes.

批量订单请前往 <https://www.espressif.com/zh-hans/contact-us/sales-questions>。

硬件参考

功能框图

ESP-Prog 的主要组件和连接方式如下图所示。

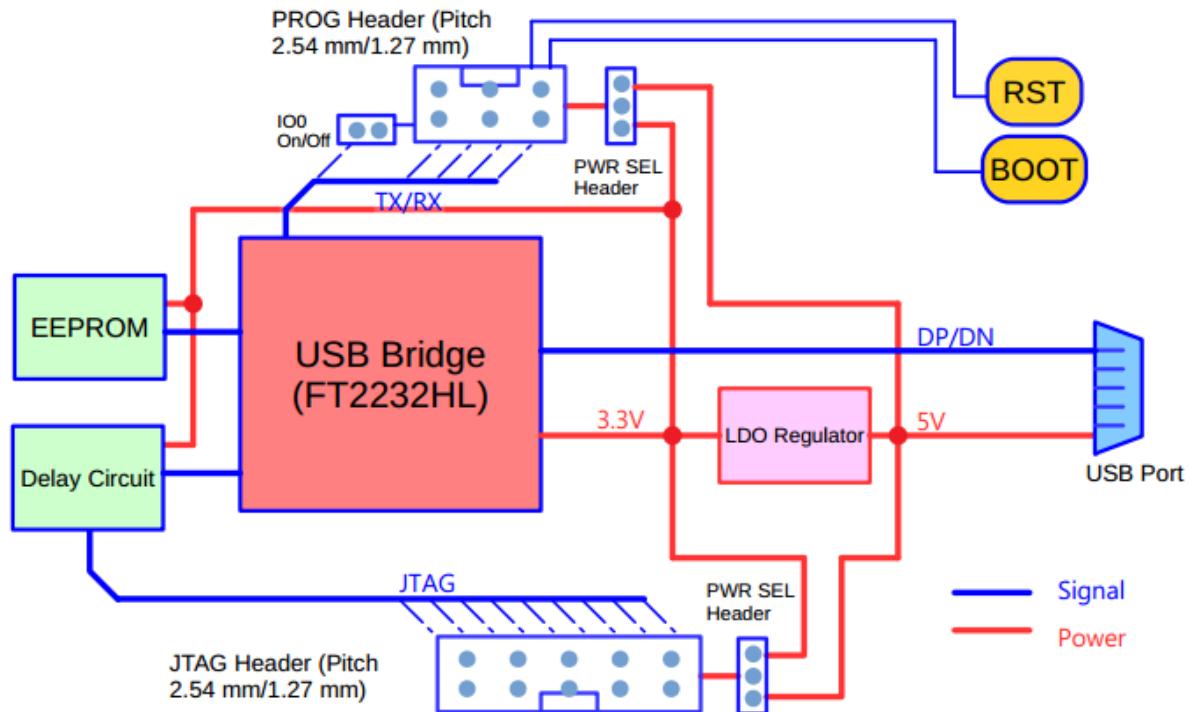


图 81: ESP-Prog 功能框图 (点击放大)

电源选项

开发板有三种互不兼容的供电方式:

- 默认使用 ESP-Prog USB 端口供电 (推荐供电方式)
- 5 V 和 G (GND) 管脚供电
- 3.3 V 和 G (GND) 管脚供电

排针

下列的两个表格提供了开发板两侧管脚 (Program 接口和 JTAG 接口) 的 **名称**和**功能**。管脚名称如 ESP-Prog 正面图所示，管脚编号与 ESP-Prog 原理图 (PDF) 中一致。

Program 接口

| 编号 | 名称 | 功能 |
|----|---------|--------------|
| 1 | ESP_EN | 使能信号 |
| 2 | VDD | 供电 |
| 3 | ESP_TXD | TX 管脚 |
| 4 | GND | 地线 |
| 5 | ESP_RXD | RX 管脚 |
| 6 | ESP_IO0 | Strapping 管脚 |

JTAG 接口

| 编号 | 名称 | 功能 |
|----|---------|-------------------|
| 1 | VDD | 供电 |
| 2 | ESP_TMS | JTAG TMS 管脚, 选择模式 |
| 3 | GND | 地线 |
| 4 | ESP_TCK | JTAG TCK 管脚, 时钟输入 |
| 5 | GND | 地线 |
| 6 | ESP_TDO | JTAG TDO 管脚 |
| 7 | GND | 地线 |
| 8 | ESP_TDI | JTAG TDI 管脚 |
| 9 | GND | 地线 |
| 10 | NC | 无 |

相关文档

- [ESP-Prog 原理图 \(PDF\)](#)
- [ESP-Prog PCB 布局图 \(PDF\)](#)
- [ESP-Prog 尺寸图 \(PDF\)](#)

有关本开发板的更多设计文档, 请联系我们的商务部门 sales@espressif.com。

参考文档

[English]

功能介绍

USB 桥接器的工作模式

ESP-Prog 采用 FTDI 公司的 FT2232HL 作为 USB 桥接控制器芯片，可通过配置将 USB 2.0 接口转换为支持多种工业标准的串行和并行接口。ESP-Prog 使用 FT2232HL 默认的双异步串行接口模式，用户只需在电脑上安装相应的 [FT2232HL 芯片驱动](#) 即可使用。

注解：在电脑端识别出的两个端口中，序号大的是 Program 接口，序号小的是 JTAG 接口。

通讯接口

ESP-Prog 可以通过 Program 接口和 JTAG 接口连接 ESP32 产品板。

- **Program 接口**

Program 接口有 6 个管脚，包括了 UART 接口 (TXD, RXD)、启动模式选择管脚 (ESP_IO0) 和复位管脚 (ESP_EN)。用户板上 Program 接口管脚设计应如下图。

- **JTAG 接口**

用户板上 JTAG 接口管脚设计应如下图。

- **防呆设计**

ESP-Prog 接口使用牛角座连接器 (DC3-6P/DC3-10P)，具有防反接功能。建议用户使用同类型的连接器，如 FTSH-105-01-S-DV-* 或 DC3-*P。

注解：由于排线具有方向性，ESP-Prog 的每个接口都与排线一一对应，如果使用不配套的排线会导致接错接口，所以请使用官方提供的排线。

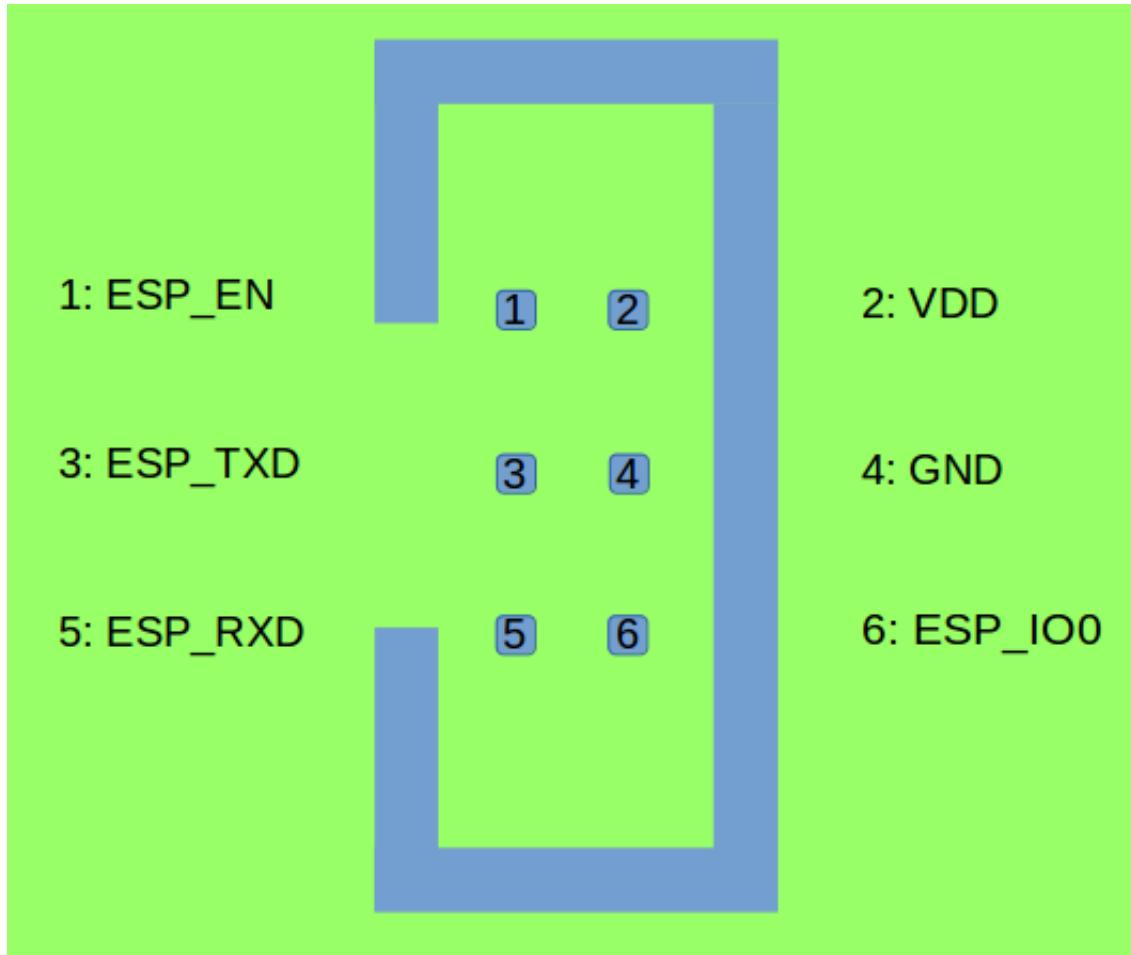


图 82: Program 接口 (点击放大)

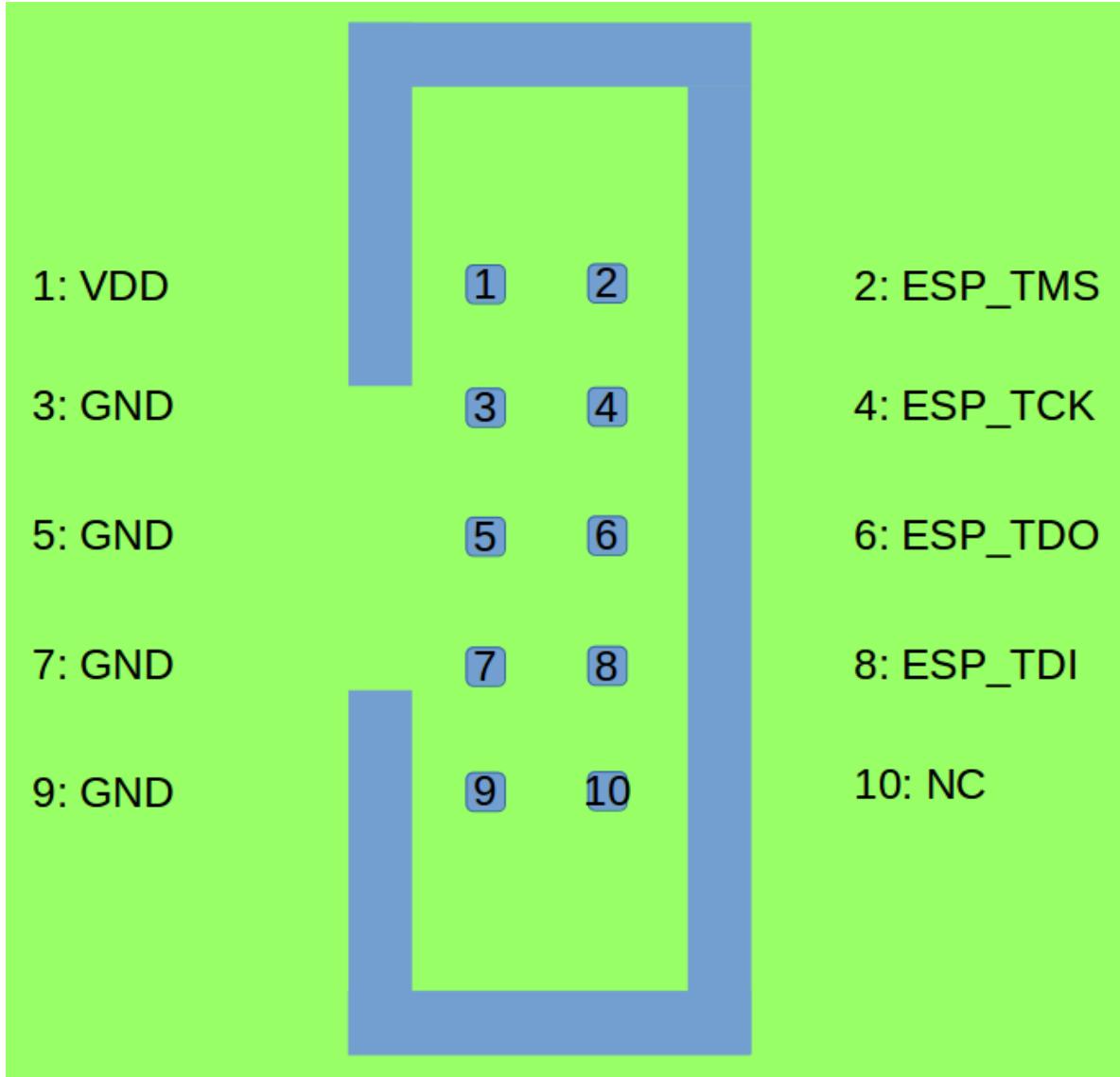


图 83: JTAG 接口 (点击放大)

自动下载功能

ESP-Prog 支持自动下载功能。连接 Program 接口到用户板之后，下载程序会控制复位管脚和启动选择模式管脚的状态，执行下载命令后即可实现设备的自动下载和运行，无需用户手动重启和选择下载模式。ESP-Prog 上的两个按键能实现手动复位设备和控制设备的启动模式。

自动下载电路的原理图如下。

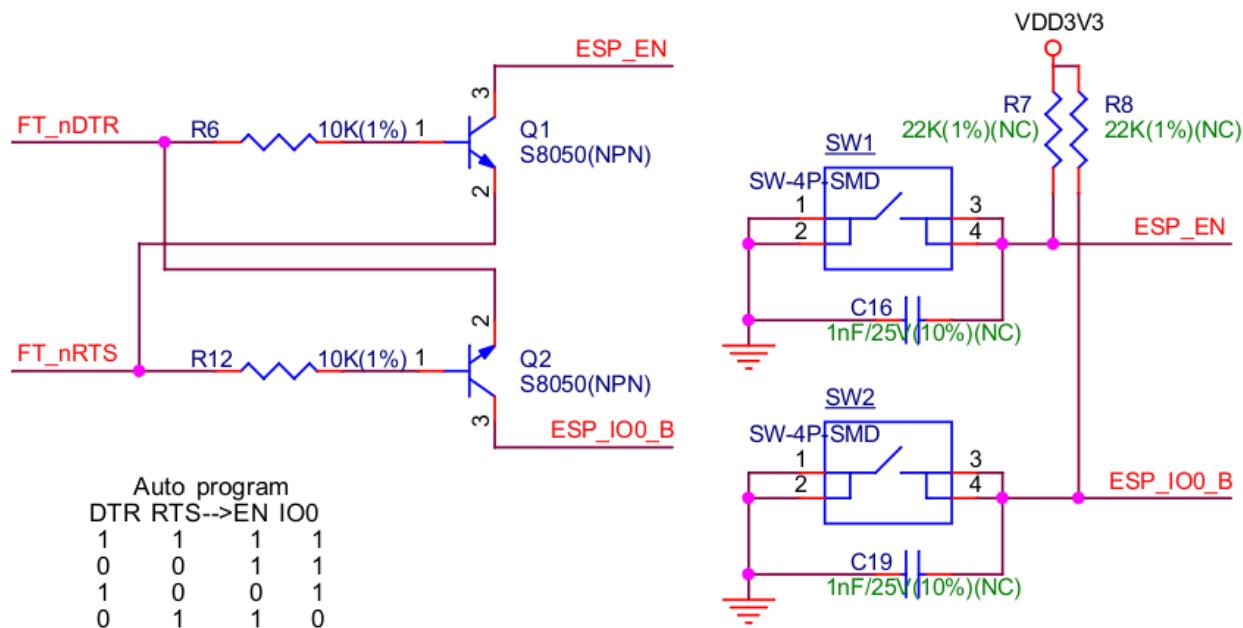


图 84: 自动下载电路 (点击放大)

延时电路

延时电路包括了总线缓存、反相器、MOS 管、一阶 RC 电路等器件。延时电路可以保证 ESP32 芯片在上电或复位过程中，先完成自身的上电启动或复位，再接通 JTAG 信号，以免受到 JTAG 影响。

LED 状态指示灯

- 当系统的 3.3 V 电源通电时，红色 LED 灯亮起；
- 当下载板向 ESP32 发送数据时，绿色 LED 灯亮起；
- 当下载板接收来自 ESP32 的数据时，蓝色 LED 灯亮起。

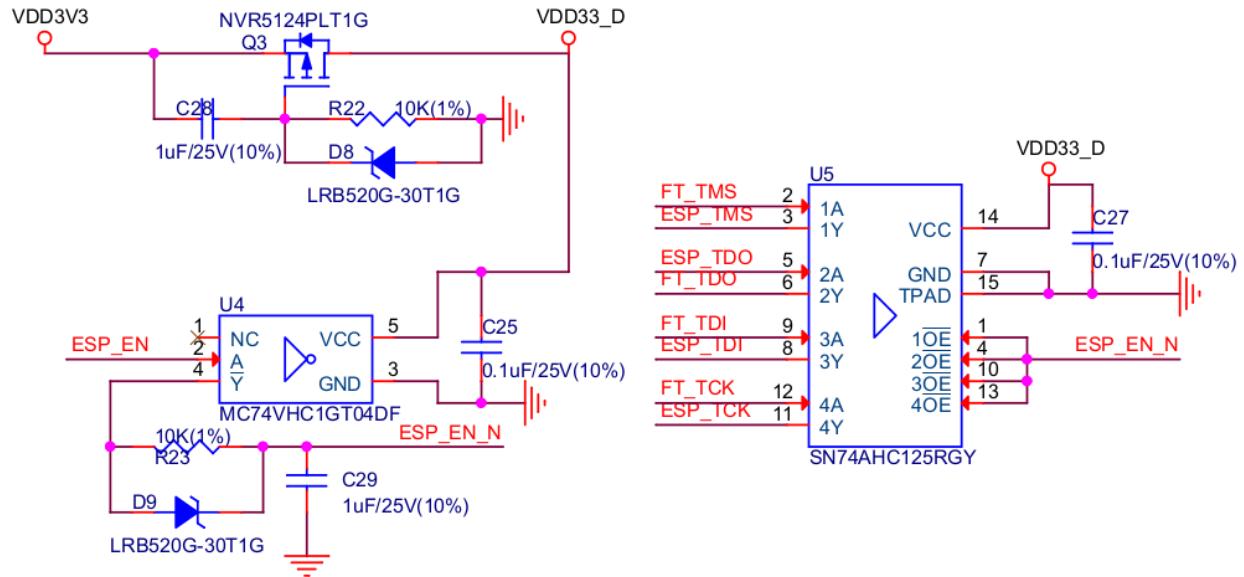


图 85: ESP-Prog 延时电路 (点击放大)

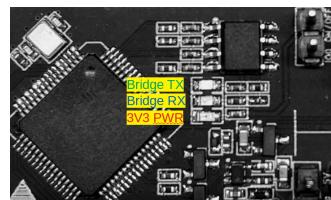


图 86: LED 状态 (点击放大)

排针

可以通过排针选择 Program 和 JTAG 接口中的参考电源，如下图所示。

- **接口电源选择排针**

中间的排针是每个接口的电源输入管脚。与 5 V 连接时，接口的电源输出为 5 V；与 3.3 V 连接时，接口电源输出则为 3.3 V。

- **IO0 On/Off 排针**

IO0 是 ESP8266 和 ESP32 的 Boot 模式选择管脚，芯片上电后，IO0 可作为正常 GPIO 使用。为了不让 ESP-Prog 自动下载电路影响用户板上 IO0 的正常使用，用户可手动控制 IO0 的通断。



图 87: 排针（点击放大）

有关本开发板的更多设计文档，请联系我们的商务部门 sales@espressif.com。

1.1.11 ESP32-S2-HMI-DevKit-1

[\[English\]](#)

ESP32-S2-HMI-DevKit-1 面向 GUI 应用场景设计，可实现智能家居交互面板，带显示屏的音箱、闹钟等人机交互界面的智能控制。

开发板的应用示例存放在 Examples 文件夹中。

ESP32-S2-HMI-DevKit-1 V1.0

[\[English\]](#)

本指南将帮助您快速上手 ESP32-S2-HMI-DevKit-1，并提供该款开发板的详细信息。

ESP32-S2-HMI-DevKit-1 面向 GUI 应用场景设计，可实现智能家居交互面板，带显示屏的音箱、闹钟等人机交互界面的智能控制。该开发板具有丰富的板载传感器和拓展接口，方便用户快速进行二次开发，实现多样的功能。

该开发板的主要功能如下：

- **内置模组：**ESP32-S2-WROVER 模组，内置 4 MB flash 和 2 MB PSRAM
- **显示：**使用 16 位 8080 并口的 4.3 寸 TFT-LCD，分辨率为 480×800，配有 256 级硬件 DC 背光调节电路，并配备电容式触摸面板，使用 I2C 接口进行通讯

- **音频:** 音频放大器、板载模拟麦克风、扬声器接口
- **存储:** microSD 卡接口
- **传感器:** 三轴加速计和三轴陀螺仪的惯性传感器、环境光传感器、温湿度传感器
- **扩展:** SPI 接口、TWAI（兼容 CAN 2.0）接口、I2C 接口、UART/Prog 接口
- **LED:** 可编程 RGB LED 和 IR LED
- **按键:** Wake Up 键和 Reset 键
- **USB:** 1 x USB-C OTG (DFU/CDC) 接口、1 x USB-C 调试接口
- **电源:** 5 V 和 3.3 V 电源接口
- **可选电池:** 1950 mAh 单芯锂离子电池，配有充电 IC



图 88: ESP32-S2-HMI-DevKit-1 (板载 ESP32-S2-WROVER 模组)

本指南包括如下内容：

- [入门指南](#): 简要介绍了开发板和硬件、软件设置指南。
- [硬件参考](#): 详细介绍了开发板的硬件。
- [相关文档](#): 列出了相关文档的链接。

入门指南

本小节将简要介绍 ESP32-S2-HMI-DevKit-1，说明如何在 ESP32-S2-HMI-DevKit-1 上烧录固件及相关准备工作。

组件介绍

该开发板是基于 ESP32-S2 的 HMI 开发板，其板载资源如下图所示。

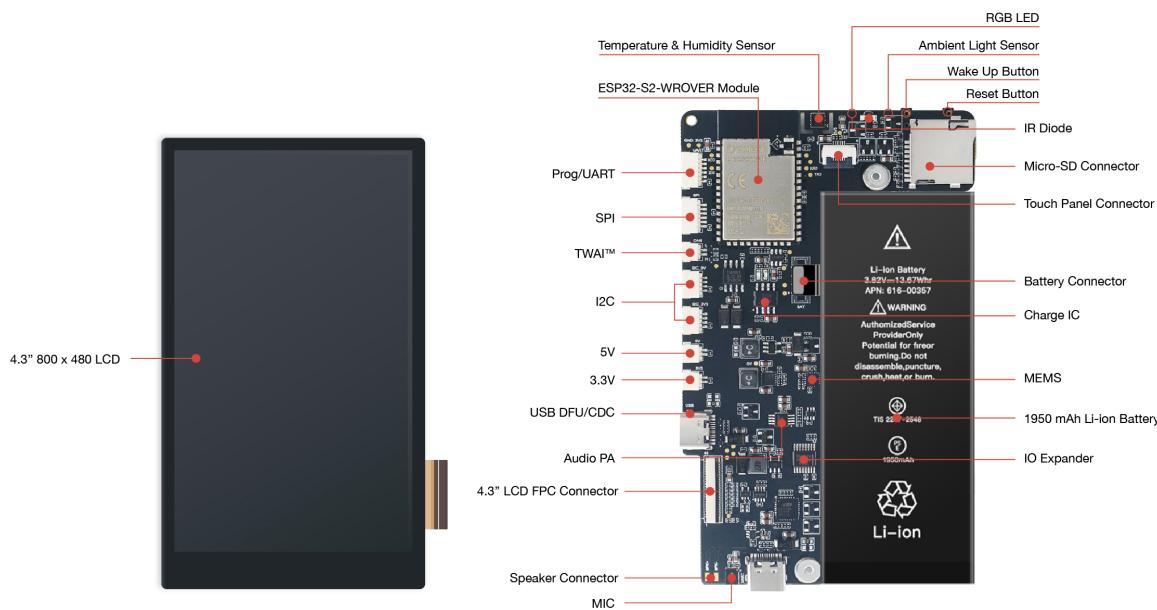


图 89: ESP32-S2-HMI-DevKit-1 (点击放大)

以下按照顺时针的顺序依次介绍开发板上的主要组件（开发板接口将在[接口](#) 小节单独介绍）。

| 主要组件 | 介绍 |
|------------------------|--|
| ESP32-S2-WROVER 模组 | ESP32-S2-WROVER 集成 ESP32-S2，是通用型 Wi-Fi MCU 模组，功能强大。该模组采用 PCB 板载天线，配置了 4 MB SPI flash 和 2 MB SPI PSRAM。 |
| 温湿度传感器 | 温湿度传感器，用于检测环境温度与湿度，通过 I2C 总线读取。 |
| RGB LED | 可寻址 RGB 发光二极管 (WS2812)，由 GPIO21 驱动。需要通过 IO 扩展器在 RGB LED 与 IR LED 间切换。 |
| 环境光传感器 | 环境光传感器，用于检测环境光强度，通过 I2C 总线读取。 |
| Wake Up Button (下载键) | 下载按键。按住 Boot 键的同时按一下 Reset 键进入“固件下载”模式，通过串口下载固件。该按键同时也可被配置用于从 Deep Sleep 模式唤醒设备。 |
| Reset Button (Reset 键) | 复位按键。 |
| IR LED | 红外发射二极管，由 GPIO21 驱动。需要通过 IO 扩展器在 RGB LED 与 IR LED 间切换。 |
| 电源 IC | 用于为电池充电。 |
| MEMS | 3 轴加速计和 3 轴陀螺仪。 |
| 1950 mAh 锂电池 | 可选 1,950 mAh 可充电锂电池。 |
| IO 扩展器 (IO 扩展器) | 通过 I2C 总线扩展 GPIO。 |
| MIC | 板载模拟麦克风。 |
| 音频 PA | 音频放大器。 |
| 4.3" 800 × 480 LCD | 使用 16 位 8080 并口的 4.3 寸 TFT-LCD，分辨率为 480×800，配有 256 级硬件 DC 背光调节电路，并与一个 I2C 电容式触摸板相连。 |

开始开发应用

通电前，请确保开发板完好无损。

必备硬件

- 1 x 装有 Windows、macOS 或 Linux 系统的 PC（推荐使用 Linux 操作系统）
- 1 x ESP32-S2-HMI-DevKit-1
- 1 x USB-C 数据线（如果想同时评估 MCU 的 USB 功能，推荐准备 2 根 USB-C 数据线）
- 1 x 扬声器 (8 Ohm, 2 W)
- 1 x microSD 卡（部分示例可能需要大容量存储）

注解：请确保使用适当的 USB 数据线。部分数据线仅可用于充电，无法用于数据传输和编程。

硬件设置

为了方便您对所有例程进行快速评估，请按照以下步骤设置开发板：

1. 将 microSD 卡插入卡槽。请确保您的重要资料已经备份，若分区不为 FAT 格式，该 microSD 可能会被格式化。
2. 如果需要评估音频播放功能，请将开发板下方 USB 口附近的扬声器焊盘连接至附赠的扬声器，或者连接至其它相近规格的扬声器 (8 Ohm, 2 W)。

软件设置

首先，请确保您已经正确完成 IDF 环境配置。为确保这一点，请在终端中输入 `idf.py --version`，如果输出结果类似于 `ESP-IDF v4.2-dev-2084-g98d5b5dfd`，则代表安装成功。详细的安装和配置说明请参考 [快速入门文档](#)。

配置完成后，请切换到仓库所在的目录。所有的示例代码都被放置在 `examples` 目录下。您可以运行 `idf.py build` 对工程进行编译。

工程选项

我们为 ESP32-S2-HMI-DevKit-1 开发板提供了丰富示例，如下所示：

- 在屏幕上打印 “Hello world!”： `esp32-s2-hmi-devkit-1/examples/get-started/hello_world`
- 使 WS2812 LED 灯闪烁并在屏幕上显示颜色： `esp32-s2-hmi-devkit-1/examples/get-started/led_blink`
- 启动交互界面以配置 Wi-Fi 凭证： `esp32-s2-hmi-devkit-1/examples/get-started/provision`
- 利用 ADC 从模拟 MIC 获取音频： `esp32-s2-hmi-devkit-1/examples/audio/audio_record`
- 播放音乐： `esp32-s2-hmi-devkit-1/examples/audio/music_player`
- 使特定区域进入 Deep-sleep 模式： `esp32-s2-hmi-devkit-1/examples/examples/power`
- 使用 Freetype 渲染字体： `esp32-s2-hmi-devkit-1/examples/freetype`
- 使用板载传感器： `esp32-s2-hmi-devkit-1/examples/sensors`
- 使用智慧触摸板： `esp32-s2-hmi-devkit-1/examples/smart-panel`
- 查看 SD 卡上的文件： `esp32-s2-hmi-devkit-1/examples/storage/sdcard_fatfs`
- USB flash disk： `esp32-s2-hmi-devkit-1/examples/storage/usb_msc`

您可以在各示例目录下，输入 `idf.py menuconfig` 配置工程选项。

在 `menuconfig` 中，请确保正确配置以下选项：

- (Top) > HMI Board Config > HMI board： 开发板版本选择，默认请选择 `ESP32-S2-HMI-DevKit-V2`；

- (Top) > HMI Board Config > Audio HAL: 音频输出接口选择，使用 PWM 或 DAC;
- (Top) > HMI Board Config > LCD Drivers: LCD 显示 IC 型号选择，ESP32-S2-HMI-DevKit-1 使用的显示 IC 为 RM68120;
- (Top) > Component config > ESP32S2-specific 中，请进入 Support for external, SPI-connected RAM 选项：
 - 在 SPI RAM config > Set RAM clock speed 中，将 PSRAM 的时钟设置为 80 MHz clock speed;
- (Top) -> Component config -> FreeRTOS: Tick rate (Hz) 设置为 1000。

在每个示例下，我们都提供了名为 `sdkconfig.defaults` 的默认配置文件，已将上述选项配置完成。

ESP-IDF 版本依赖

`esp32-s2-hmi-devkit-1/examples/storage/usb_msc` 需要在 IDF v4.3 下进行编译，其它示例程序可以在 IDF v4.2 及以后版本进行编译。

内含组件和包装

零售订单

每一个零售 ESP32-S2-HMI-DevKit-1 开发套件均有独立包装。

内含以下部分：

- 开发板 - ESP32-S2-HMI-Devit-1
- 线缆 - SH1.25 至 2.54mm 连接线 x 7

零售订单请前往 <https://www.espressif.com/zh-hans/company/contact/buy-a-sample>。

批量订单

如批量购买，开发板将以大纸板箱包装。

批量订单请前往 <https://www.espressif.com/zh-hans/contact-us/sales-questions>。



图 90: ESP32-S2-HMI-DevKit-1 包装

硬件参考

功能框图

ESP32-S2-HMI-DevKit-1 的主要组件和连接方式如下图所示。

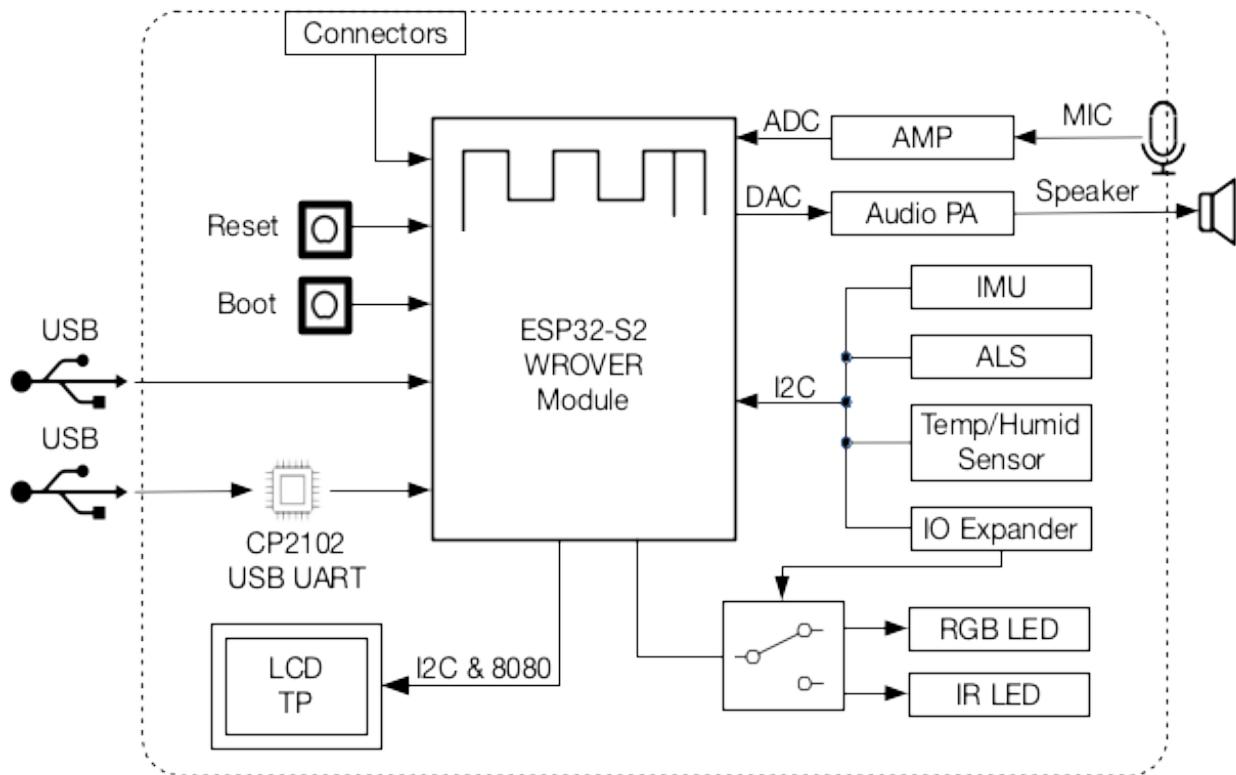


图 91: ESP32-S2-HMI-DevKit-1 功能框图

电源选项

为了降低电源功耗、提高电源效率并支持电池供电，ESP32-S2-HMI-DevKit-1 的电源域分成了 5 V 电源域和 3.3 V 电源域，其中部分电源可以通过软件控制，另一部分在硬件设计中始终开启。

开发板出厂时烧录的固件已经关闭所有可控电源域的电源，并将所有 IC 配置为低功耗模式，以降低电流消耗。

更多信息请参阅[电源](#)。

接口

该开发板提供了多个扩展接口方便用户进行二次开发，以下按照顺时针顺序依次介绍开发板上的接口，请参考 [ESP32-S2-HMI-DevKit-1 板载资源](#)。

| 连接器 | 介绍 |
|----------------------|---|
| 扬声器接口 | 用于连接扬声器。 |
| 4.3” LCD FPC 连接器 | (保留) 可使用 FPC 线连接配套的 4.3” LCD 扩展板。 |
| USB DFU/CDC | 1 个 USB-C OTG (DFU/CDC) 端口，1 个 USB-C 调试端口。 |
| 3.3 V | 3.3 V 电源接口。 |
| 5 V | 5 V 电源接口。 |
| I2C | 带 5 V/3.3 V 电源的 I2C 连接器。 |
| TWAI (兼容 CAN 2.0) 接口 | 使用双线车载串口 (Two-wire Automotive Interface, TWAI®) 协议的连接器。 |
| SPI | 在不使用 SD 卡时可以使用该连接器连接 SPI 总线的设备。 |
| Prog/UART | 可以通过该串口接口观察日志输出与固件烧录。 |
| microSD 连接器 | 可以插入 microSD 卡以扩展设备存储空间。 |
| 电源端口 | 用于连接电池。 |

下图展示了该开发板的背面，供您参考。

相关文档

- [ESP32-S2 技术规格书 \(PDF\)](#)
- [ESP32-S2-WROVER 技术规格书 \(PDF\)](#)
- [ESP32-S2-HMI-DevKit-1 原理图 \(PDF\)](#)
- [ESP32-S2-HMI-DevKit-1 PCB 布局图 \(PDF\)](#)
- [ESP32-S2-HMI-DevKit-1 尺寸图 \(PDF\)](#)
- [ESP32-S2-HMI-DevKit-1 尺寸图源文件 \(DXF\) - 可使用 Autodesk Viewer 查看](#)

有关本开发板的更多设计文档，请联系我们的商务部门 sales@espressif.com。

参考文档

[English]



图 92: ESP32-S2-HMI-DevKit-1 背面

音频

[English]

ESP32-S2-HMI-DevKit-1 支持音频播放与采集。您可以在 `esp32-s2-hmi-devkit-1/examples/audio/` 目录下找到音频播放和采集的示例。

音频播放

ESP32-S2-HMI-DevKit-1 可以通过 DAC 或 PWM 进行音频输出。我们推荐使用 PWM 进行音频输出，因为 PWM 输出具有更低的噪声和更高的分辨率（DAC 输出分辨率为 8 位，PWM 输出则最高可以在 19.2 kHz 的采样率下，达到 12 位的分辨率）。

通过 IO 口输出的信号将先进入数字电位器 TPL0401 进行无损音量调节，然后通过一个 100 nF 的隔直电容 C33 与一个 200 kOhm 的电阻 R52。该 RC 电路将截止频率控制在 8 Hz 附近。除此之外，该信号还将被送至一个功率为 3 W 的 D 类音频功率放大器 NS4150，通过外部电阻将增益设定为 1.5 倍，从而将 3.3 V 的最大输出信号放大至略低于 PA 电源（5 V）的 4.95 V，在尽量增大输出音量的同时减少饱和失真。

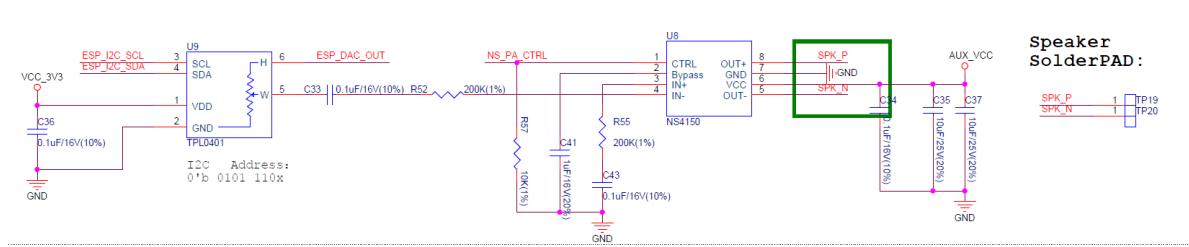


图 93: ESP32-S2-HMI-DevKit-1 音频播放原理图（点击放大）

音频功放由 5 V 电源域供电。在使用音频播放功能前，请确保该电源域处于开启状态（参阅 5 V 电源域 小节）。

音频采集

ESP32-S2-HMI-DevKit-1 可以通过芯片内置的 ADC 对模拟麦克风进行音频采集。

开发板搭载了一个灵敏度为 -38 dB 的模拟麦克风，并会将输出信号发送至固定增益的运算放大器 TLV6741 以放大信号。

麦克风与运算放大器由可控 3.3 V 电源域供电。在使用音频采集功能前，请确保该电源域处于开启状态（参阅 3.3 V 电源域 小节）。

请使用 Timer Group 中断来进行音频采集，请勿在任务中使用诸如以下形式的代码进行音频采集：

```
size_t index = 0;
uint16_t audio_data[configMAX_ACQUIRE_COUNT];
do {
    audio_data[index] = adc1_get_raw(CONFIG_AUDIO_CHANNEL);
```

(下页继续)

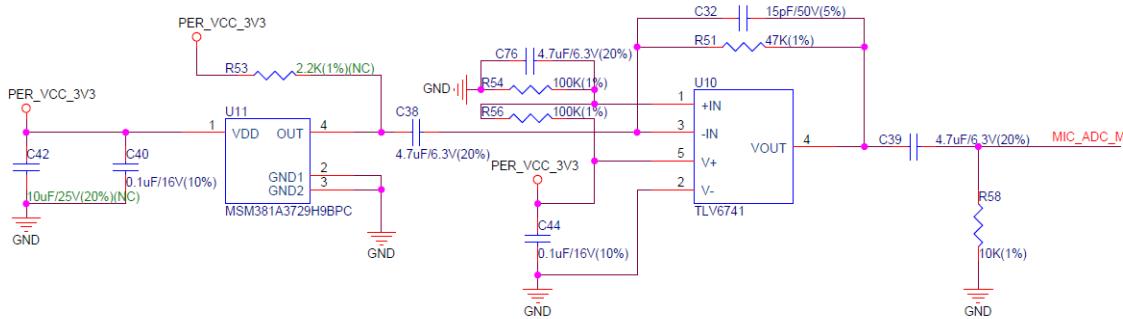


图 94: ESP32-S2-HMI-DevKit-1 模拟麦克风原理图 (点击放大)

(续上页)

```

    ets_delay_us(1000000 / CONFIG_AUDIO_FREQ);
} while (++index < CONFIG_MAX_ACQUIRE_COUNT)

```

若使用上述方式进行采集，将导致 CPU 被占用而无法释放，从而触发任务看门狗（如果未被禁用），并使得其他更低优先级的任务（包括 IDLE Task）无法运行。

在中断函数中进行 ADC 读数采集时，您需要重写 ADC 采集的函数，确保这些函数为 IRAM_ATTR 以降低中断的响应时间，并且将变量放置于 DRAM 中。同样的，在该函数中，不要使用任何信号量。您可以参考 examples 文件夹下的 audio/audio_record 的实现方式。

ADC 精度

尽管缺少基准电压源以及采用 Buck 供电或将导致整体电源噪声较大，ESP32-S2 的 ADC 仍然具有较高的重复性。

ADC 使用 13 位分辨率，衰减为 11 dB，对应满量程电压为 2.6 V。经由 ADC1_CH8 轮询采集 AD584T 基准电压源的 2.5 V 电压输出 4096 次的未经校准原始值转换为电压后，我们得到了下图数据：

可以看到，在未经校准的数据下，绝大多数数据的误差在 ± 0.005 V 内，其标准差 σ 仅为 3.339 LSB (0.00106 V)。这些误差主要来自于绝对精度，即偏置上。因此，对于 ADC 采样的音频，可将其失真性和噪声控制在较低的水平。

AD584T 的 10 V 电压在 0.1 ~ 10 Hz 的输出噪声峰值为 50 uV，2.5 V 电压的输出噪声峰值则由内部经过高精度激光调整的电阻分压得到，并由 10 V 输出测的晶体管提供最大 30 mA 的推挽能力。其在 2.5 V 下的输出噪声小于 16 位 ADC 的分辨能力，可以作为测试基准。

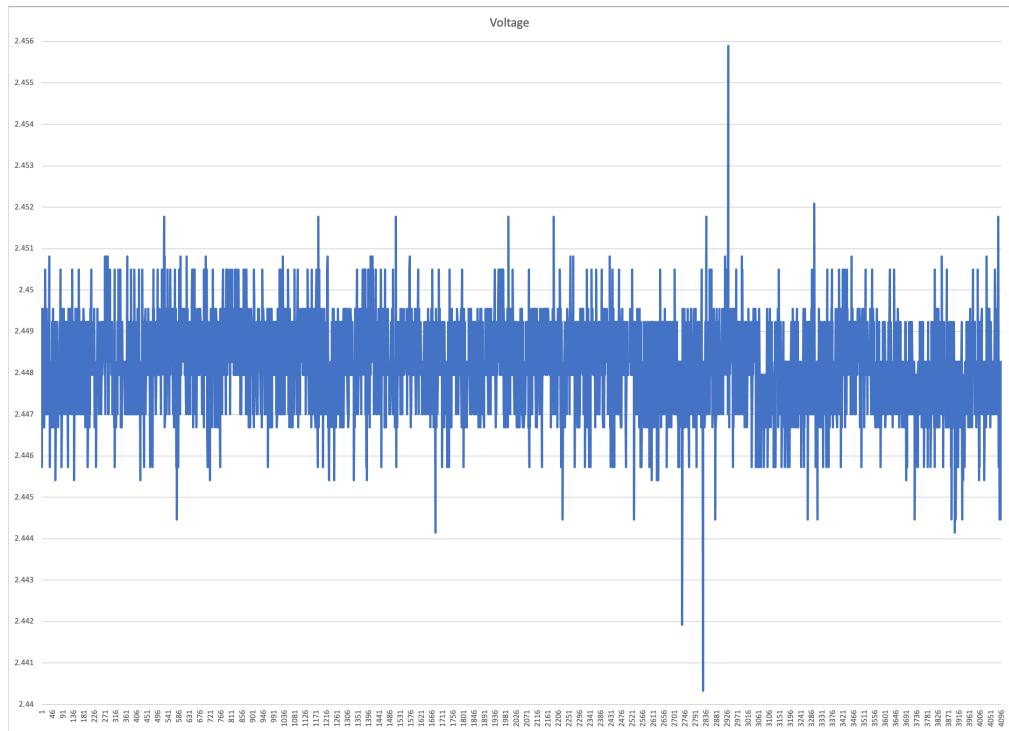


图 95: ESP32-S2-HMI-DevKit-1 ADC (点击放大)

显示屏与触摸面板

[English]

ESP32-S2-HMI-DevKit-1 配备了一块分辨率为 800×480 的 4.3 英寸彩色液晶显示屏，并附带有 I2C 接口的触摸面板。该屏幕的接口类型与数据位宽由可编程引脚控制。该开发板已通过电阻配置为 16 位的 8080 并口通讯。

该 LCD 使用的显示 IC 为 RM68120，触控 IC 为 FT5436。

通讯

ESP32-S2-HMI-DevKit-1 所用 LCD 的显示 IC 已配置为使用 16 位的 8080 接口进行通讯，共使用 18 个 GPIO 接口，包括 16 位数据线 (LCD_D0…LCD_D15)，位时钟信号 (LCD_WR) 及数据命令区分信号 (LCD_DC/LCD_RS)。

触控 IC 使用 I2C 接口与 MCU 通讯，可与其他 I2C IC 共用，无需额外占用 GPIO 接口。触控 IC 支持中断信号输出。该中断信号将被发送至 IO 扩展器的 P2 引脚，由该引脚产生的下降沿会使 IO 扩展器的中断输出脚产生低电平，从而使 MCU 接收到中断信号。此时，可通过读取 IO 扩展器的输入电平寄存器判读中断来源，从而确定该中断是否来自触控 IC。完成一次输入电平的读取会自动清除中断标志。

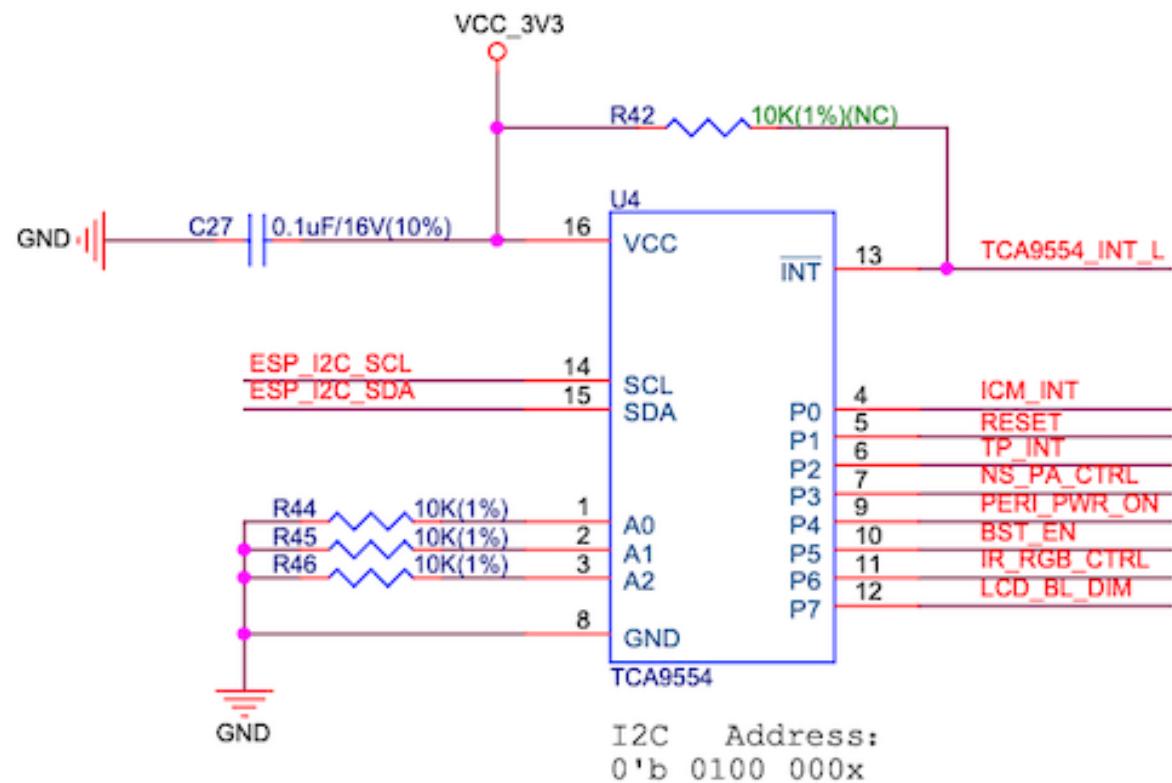


图 96: ESP32-S2-HMI-DevKit-1 IO 扩展器原理图

背光

该 LCD 内置了串联型 LED，需要使用 Booster 电路进行恒流驱动。额定电流为 18 mA，此时电压约为 24 V (可能有误差，仅供参考)。为了防止屏幕未连接时 Booster 电路的反馈电压始终为 0，从而导致高压加载到背光滤波电容 C21 的两端，请确保该电容的耐压在 38 V 以上。

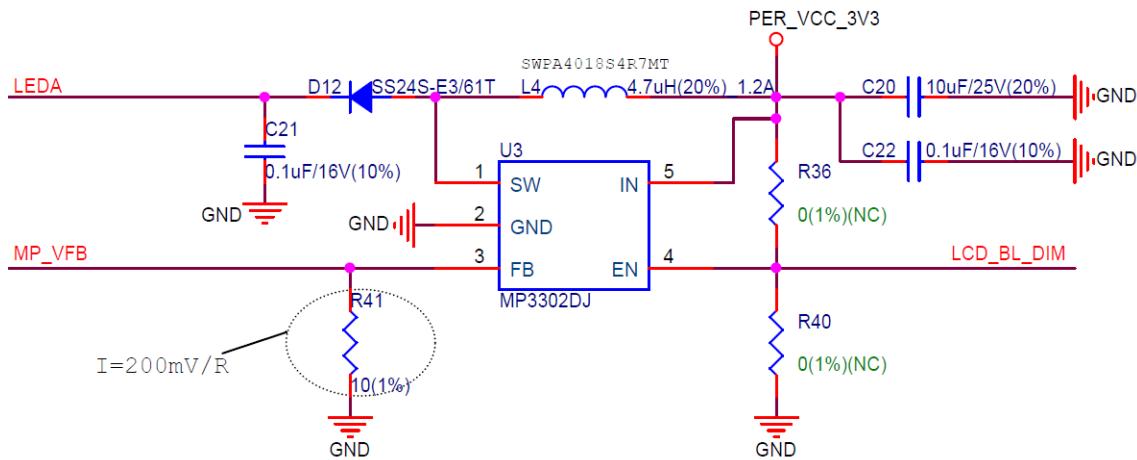


图 97: ESP32-S2-HMI-DevKit-1 PWM 调光电路原理图 (点击放大)

由于 PWM 调光可能会带来闪烁，且部分 Booster IC 不支持高频 PWM 信号控制，该开发板提供了 DC 调光电路以解决上述问题，如下图所示：

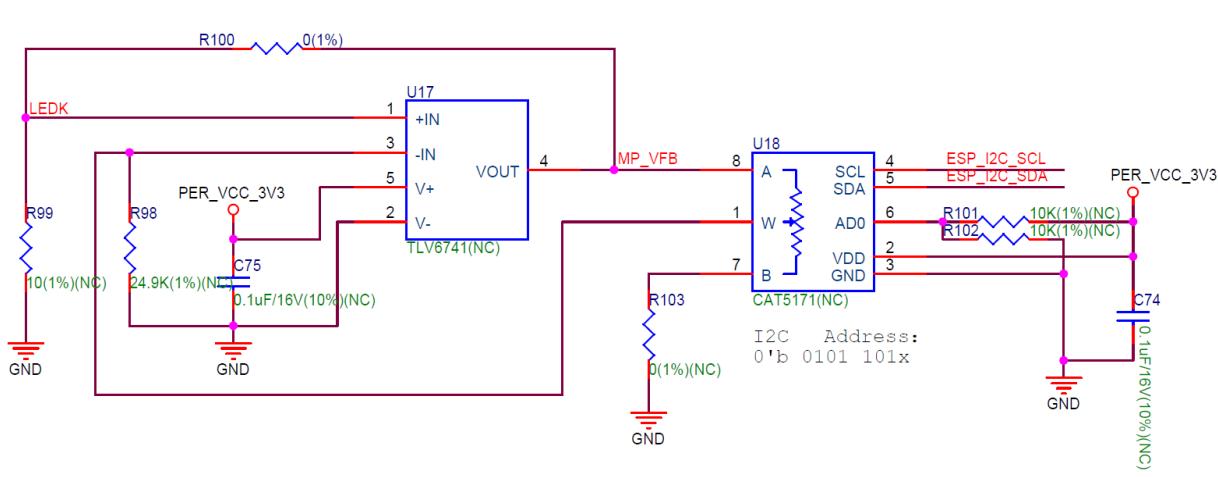


图 98: ESP32-S2-HMI-DevKit-1 DC 调光电路原理图 (点击放大)

该 DC 调光电路将 VFB 电压输入到运算放大器 TLV6741，其增益电阻为数字电位器，可以通过 I2C 总线修改其阻值以达到修改增益大小的目的。使用的数字电位器为 CAT5171，具有 256 级分辨率，最大电阻值为 50 kOhm。

Booster IC 的 EN 脚由 IO 扩展器的 P7 脚控制，高电平有效。如需在关闭屏幕的同时保留显存的内容，可以将该引脚设置为低电平来关闭屏幕背光。

触摸

开发板搭载的电容式触摸面板使用的触控 IC 分辨率为 800×480，支持最多 5 点触摸和硬件手势识别。

该显示 IC 不支持硬件坐标方向转换，因此，对于显示时使用不同的旋转方向，可能需要软件对触摸 IC 读取到的数据做交换或求其相对于分辨率的差值。多点触摸属于硬件支持，在驱动中，我们也提供了读取多个触摸点的 API 供您使用该功能，但是使用的 GUI 库 LVGL 暂时不支持多点触控的处理，您可能需要在应用层自行处理这些触摸点的数据。

电源

[English]

为了降低电源功耗、提高电源效率并支持电池供电，ESP32-S2-HMI-DevKit-1 的电源花分成了 5 V 电源域和 3.3 V 电源域，其中部分电源可以通过软件控制，另一部分在硬件设计时被配置为始终开启。

开发板出厂时烧录的固件已经关闭所有可控电源域的电源，并将所有 IC 配置为低功耗模式，以降低电流消耗。

3.3 V 电源域

该电源域负责为绝大部分的 IC 和模组供电，但其分为两部分：不可控 3.3 V 电源域和可控 3.3 V 电源域。

不可控 3.3 V 电源域无法通过软件关闭，该电源由 Buck 电路得到。在 USB 有电源输入的情况下，将从 USB 线缆输入的 5 V 电源取电；在 USB 断开的情况下，则由锂电池提供 3.6 ~ 4.2 V 的电源。该电源域负责为 ESP32-S2-WROVER 模组以及其他可通过软件控制进入低功耗模式的器件供电。

可控 3.3 V 电源域来自于不可控 3.3 V 电源域，通过 PMOS 控制开关，该 PMOS 栅极连接至 IO 扩展器的 P4 脚，低电平开启该电源。该电源负责为一些具有较大静态功耗且无法进入低功耗模式的 IC 进行供电。

5 V 电源域

```
<div id="3vpowerdomain" >
```

开发板的 5 V 电源域负责为音频功放和 TWAI® 收发器供电。其来源有以下几种方式：

- USB 接口
- 外部输入至连接器的 5 V 电源口
- 锂电池经过 Booster 电路后的电源

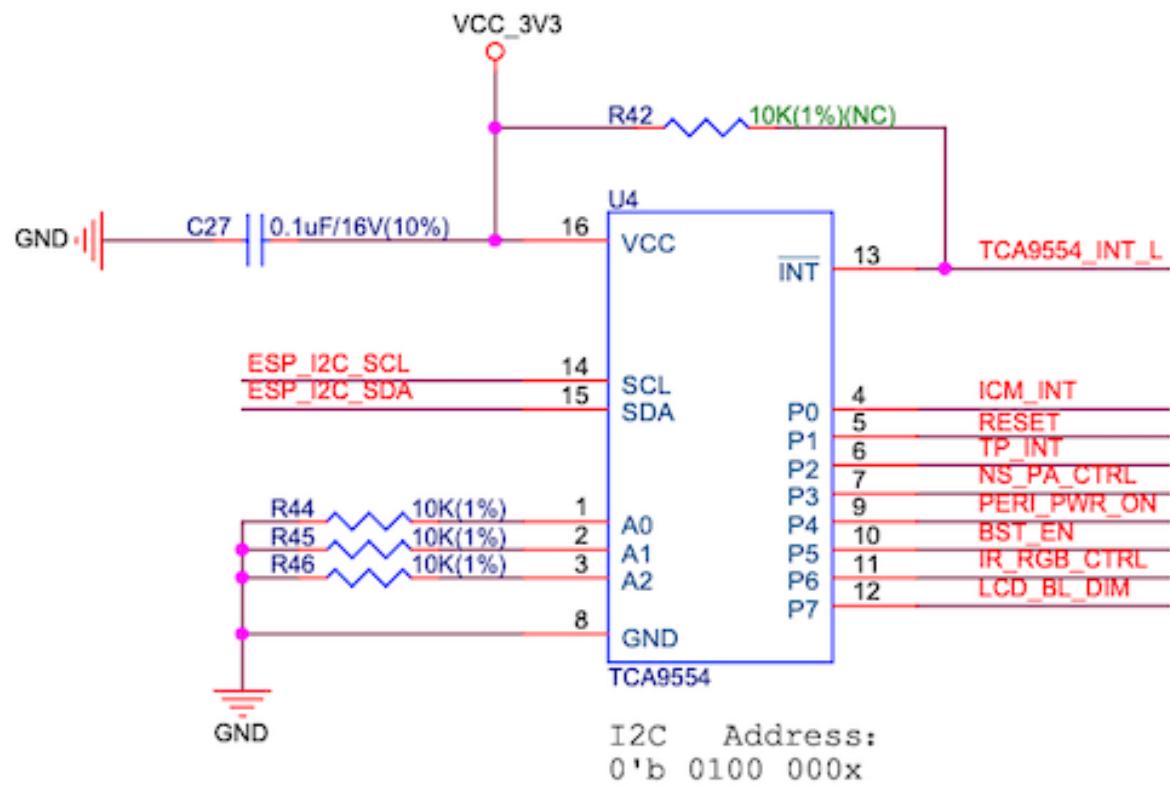


图 99: ESP32-S2-HMI-DevKit-1 IO 扩展器原理图

通过 USB 和外部 5 V 输入的电源会为所有需要 5 V 供电的器件（除 CP2102N）进行供电，且无法通过软件断开。通过电池供电时，则可以通过 IO 扩展器的 P5 引脚来控制 Booster IC 的 EN 脚电平，通过高电平开启 5 V 电源。

通过开发板底部 USB 接口输入的电源将分为两路，一路负责为 CP2102N 供电，另一路在经过二极管后成为 USB_5V。由于 CP2102N 只在连接 PC 时才需进入工作状态，因此只有在该 USB 口连接后，CP2102N 才会上电。任何 5 V 的电源输入都会关闭 Booster IC，并经由充电 IC 为板载锂离子电池进行充电。

电源依赖情况

以下功能依赖 5 V 电源域：

- TWAI®（从 USB 5 V 或 Booster 5 V 自动选择可用电源）
- 音频功放（从 USB 5 V 获取，若失败，尝试从电池获取）
- 5 V 电源输出连接器（同 TWAI®）

以下功能依赖可控 3.3 V 电源域：

- Micro-SD 卡
- 麦克风及其偏置电路、运算放大器
- 显示屏与触摸功能
- WS2812 RGB LED 与 IR LED
- IR LED

电源工作状态

当通过 USB 线缆连接开发板时，5 V 电源域自动开启，充电 IC 输出电压为电池充电，可控 3.3 V 电源域由 IO 扩展器的 P4 脚进行控制。

当使用电池为开发板进行供电时，可控 3.3 V 电源域由 IO 扩展器的 P4 脚进行控制，5 V 电源域由 IO 扩展器的 P5 脚控制，充电 IC 不会工作。

1.1.12 ESP32-Sense-Kit

[English]

ESP32-Sense-Kit 开发套件用于评估和开发 ESP32 触摸传感器功能。

ESP32-Sense-kit

[English]

概述

ESP32-Sense-Kit 开发套件用于评估和开发 ESP32 触摸传感器功能。评估套件包含一个主板和若干子板，主板包含显示单元，主控单元，调试单元；子板的触摸电极的形状和排列方式多样，包括线性滑条、矩阵按键、弹簧按键、轮式滑条等，可以满足不同的使用场景。用户也可以自行设计子板以满足特殊的使用场景。

下图是 ESP32-Sense-Kit：

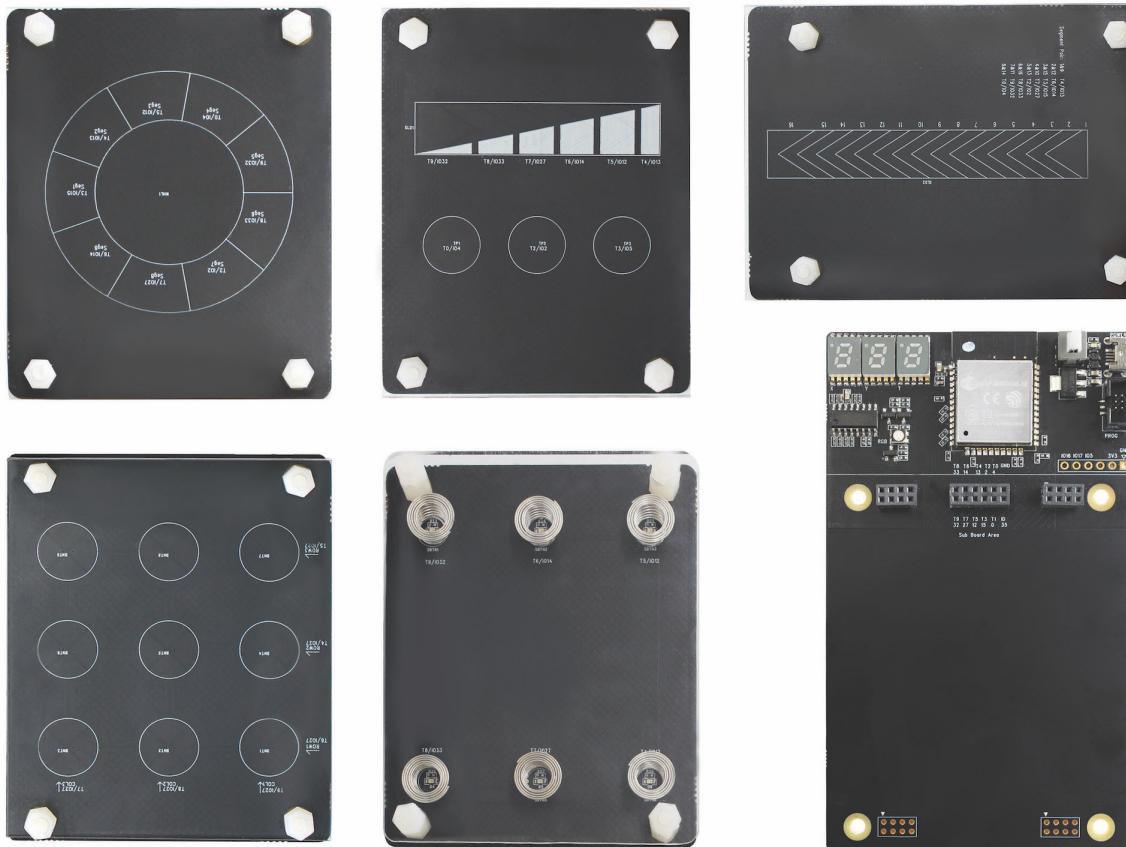


图 100: ESP32-Sense-Kit

开发板操作

- **保护盖板安装**

塑料材质保护盖板厚度应小于 3 mm。空气间隙会导致手指的触摸感应变弱，所以保护盖板应紧贴子板安装。如果有空气间隙可以使用双面胶填充。弹簧子板上应使用 7 mm 螺柱安装保护盖板。

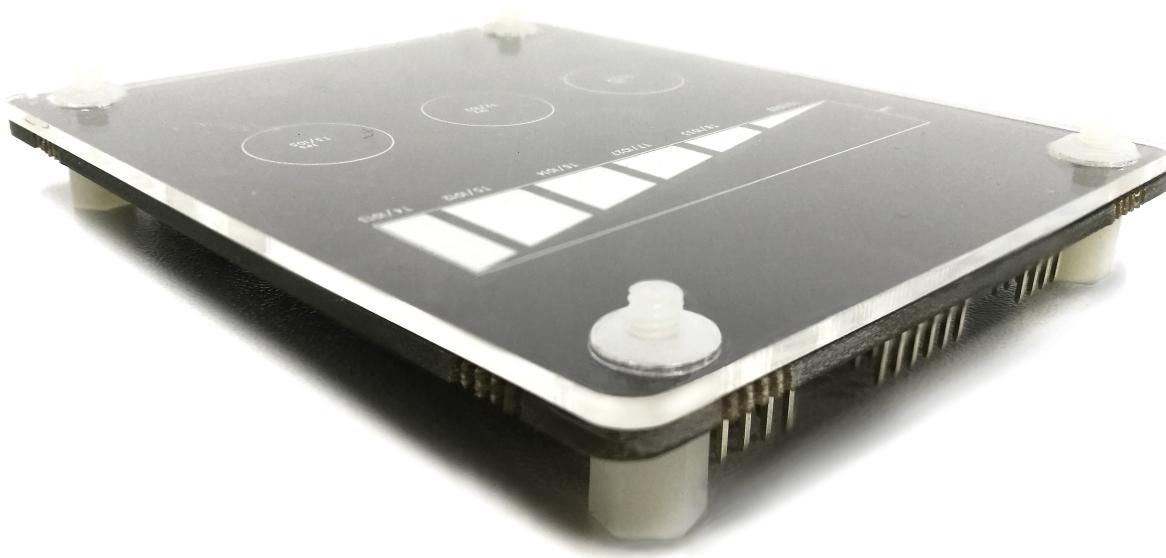


图 101: 保护盖板安装

- **子板安装**

子板与主板之间使用连接器连接，连接器既有连通作用，也可以固定住子板。子板安装孔可安装 4 个 7 mm 塑料螺柱，使子板水平支撑在主板上，如下图：

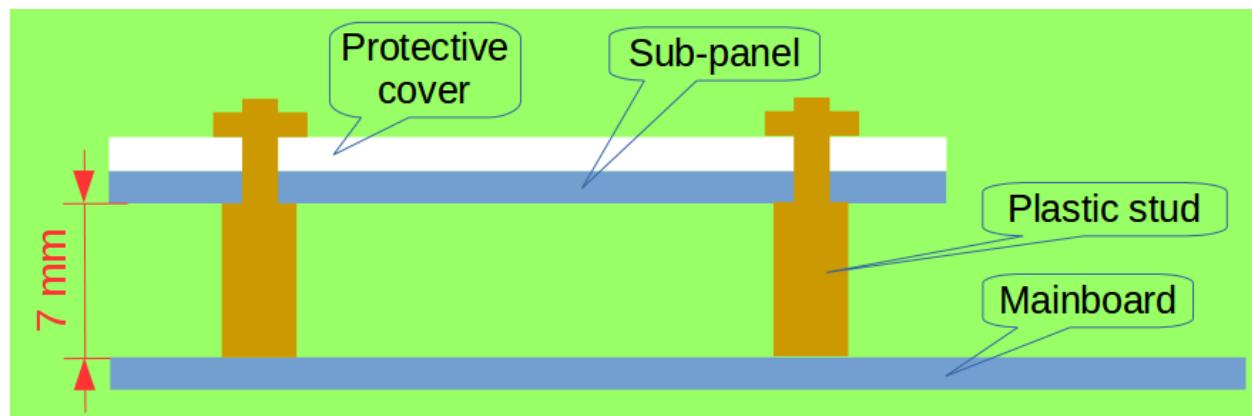


图 102: 子板安装

- **ESP-Prog 调试器设置**

ESP-Prog 起到下载程序和供电作用。调试器上有两组跳针，IO0 跳针和供电跳针，供电跳针上的供电电压选择为 5 V。因为 IO0 既有启动模式（下载模式、运行模式）选择功能，同时也是触摸功能管脚，如果 IO0 作为触摸功能使用，则运行模式时，调试器上 IO0 跳针应断开。调试器设置如下图：

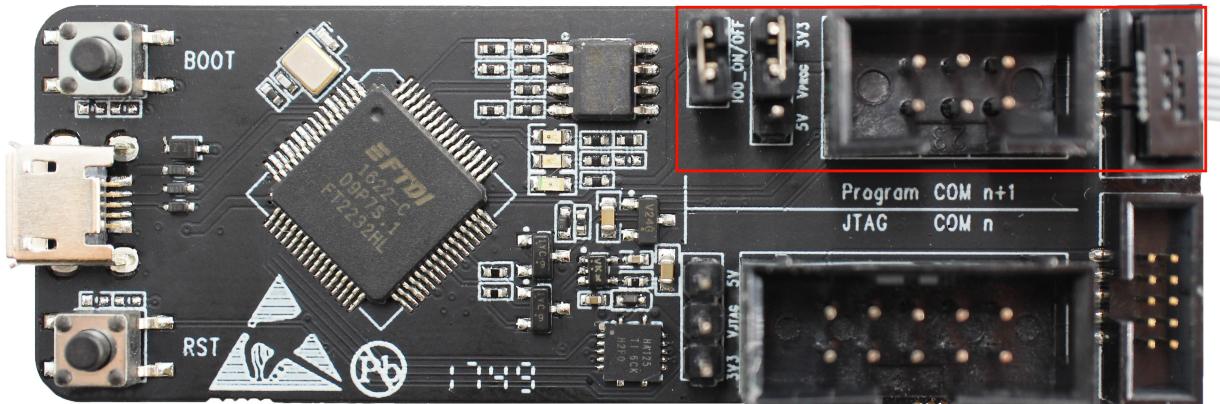


图 103: 调试器设置

- **ESP-Prog 调试器连接主板**

调试器包含 Jtag 接口和 Program 接口。与主板连接使用 Program 下载接口。

- **编译下载**

配置 [ESP32-Sense 项目工程](#) 的 config 参数，执行命令 `make menuconfig` 配置参数，如下图所示。执行命令 `make flash`，程序会自动下载到开发板中。

- **更换子板**

ESP32 上电时会检测子板上的分压电阻的分压值，识别不同的子板。更换子板后需给开发板重新上电。

硬件参考

主板

- **主板系统框图**

下图是主板的功能框图。

- **主板硬件资源**

显示单元包括数码管和 RGB 三色灯电路。调试单元包括 ESP-Prog 调试器接口。主控单元包含 ESP32 模组。Mini USB 供电作用。

- **电源管理系统**

ESP32-Sense-Kit 开发套件上 Mini USB 和 ESP-Prog 均可供电，两者之间有保护二极管隔离，供电不相互影响。USB 只具有供电功能。ESP-Prog 接口除了供电功能还具有自动烧写固件功能。下图是电源管理系统的硬件原理图：

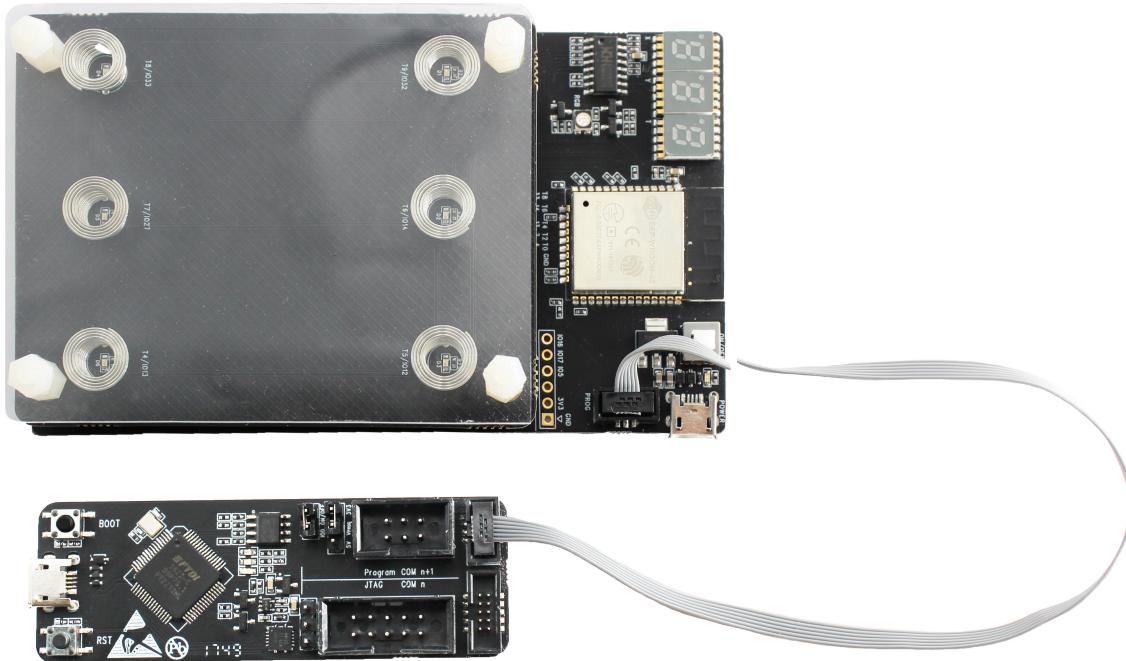


图 104: 调试器连接主板

```
/home/espressif/esp-iot-solution/examples/touch_pad_evb/sdkconfig - Espressif IoT Development Kit Configuration

Espressif IoT Development Framework Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module

SDK tool configuration --->
IoT Solution settings --->
Bootloader config --->
Security features --->
Serial flasher config --->
IoT Touch EB settings --->
Partition Table --->
Compiler options --->
Component config --->
```

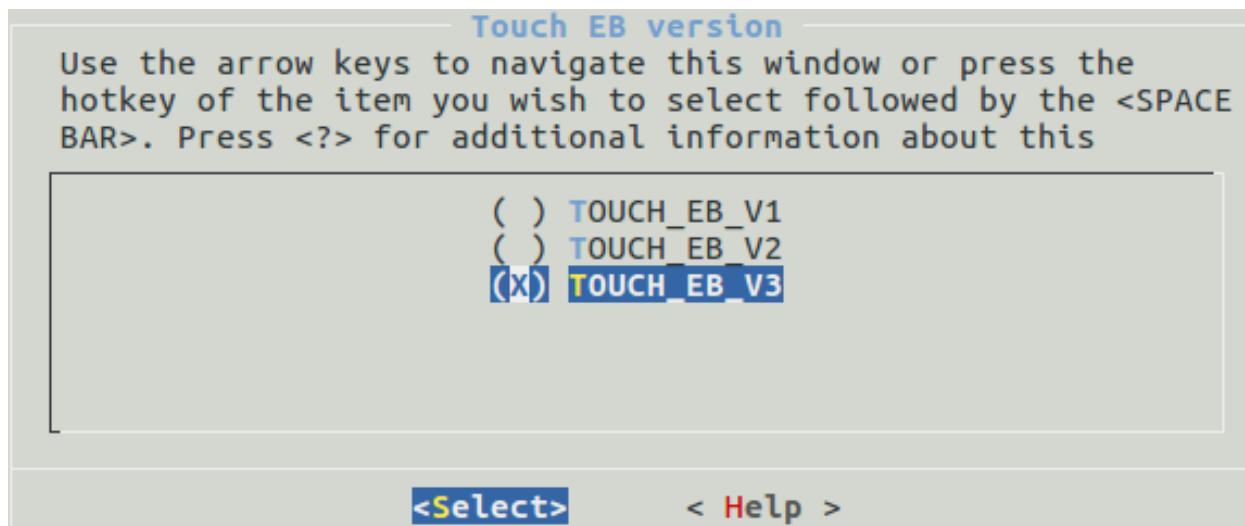


图 105: 编译下载

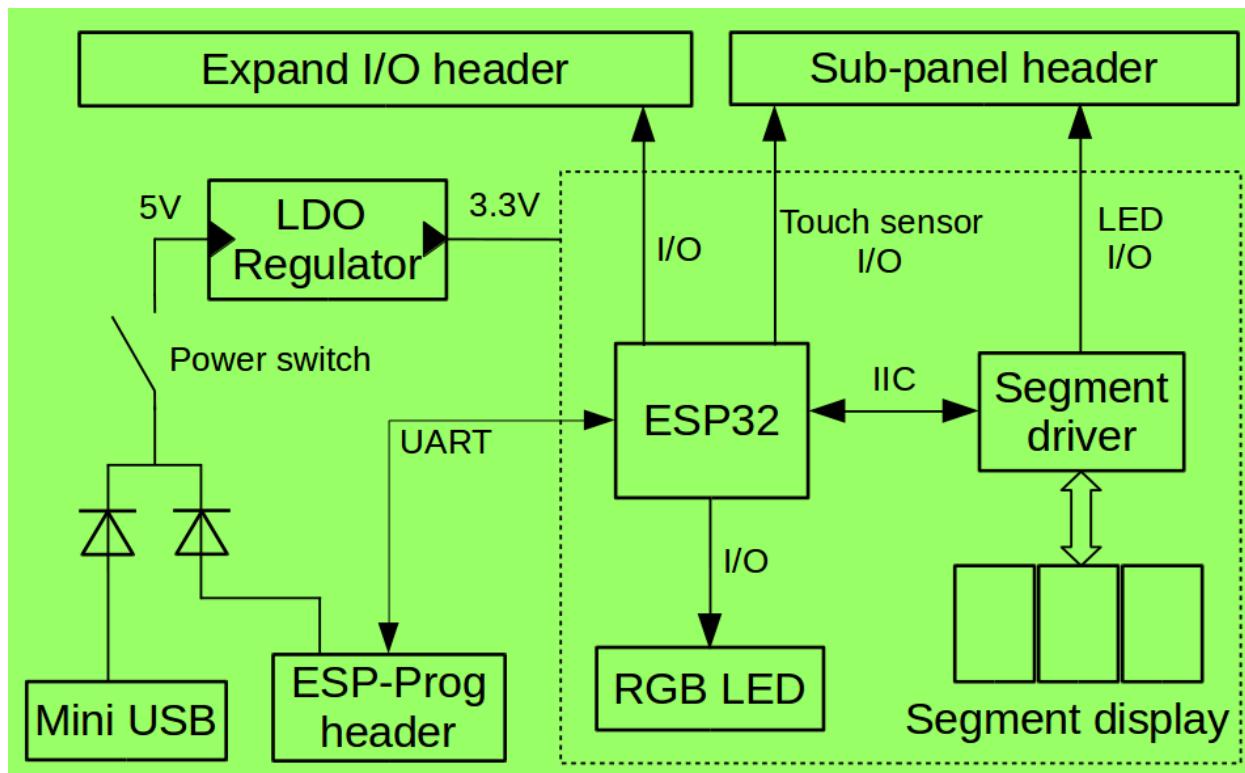


图 106: 主板功能框图

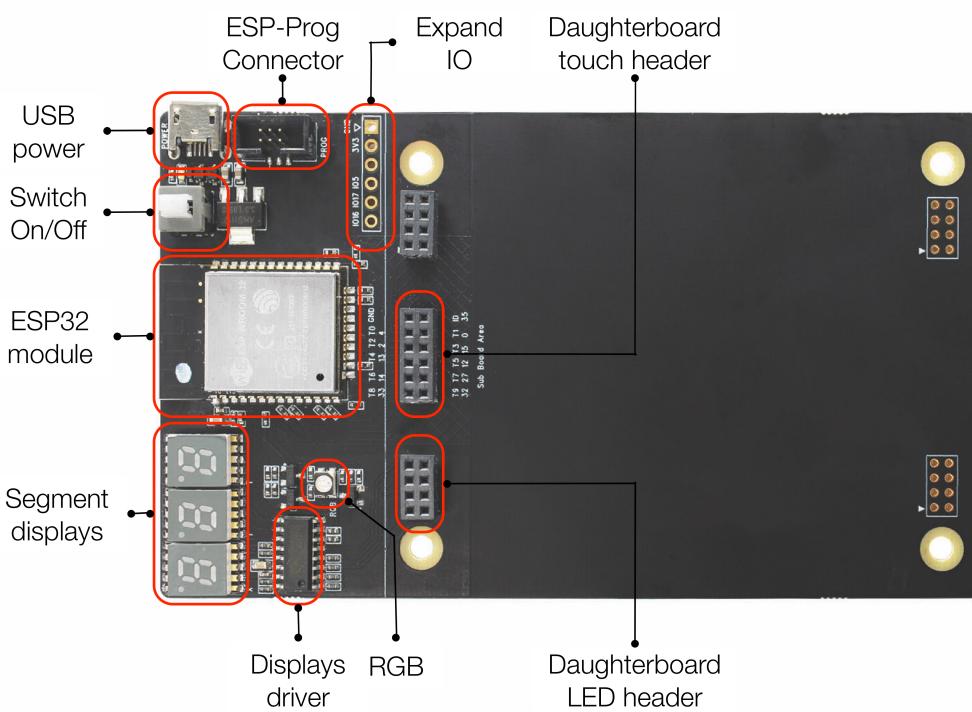


图 107: 主板硬件资源

Power Supply

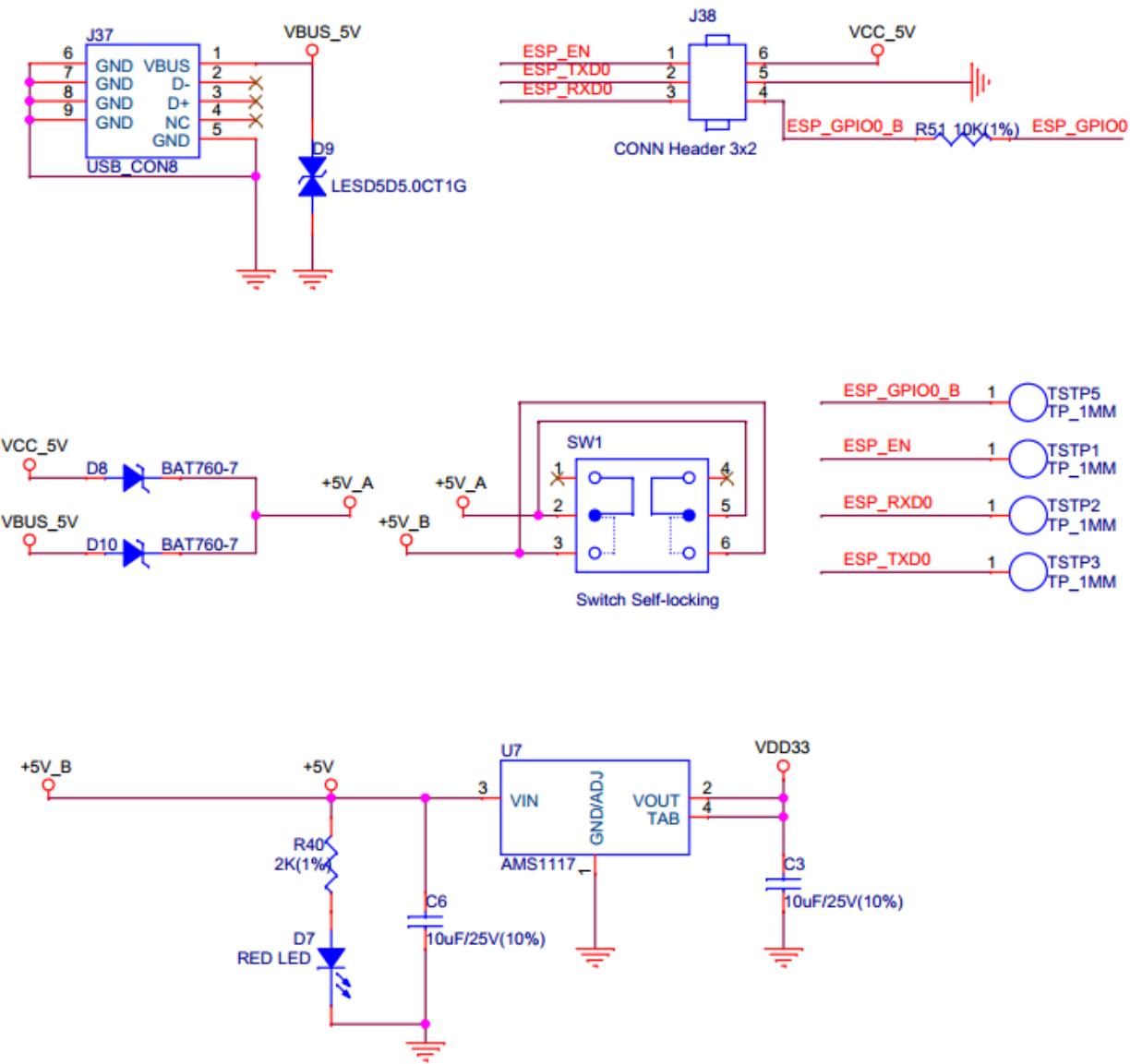


图 108: 电源管理系统

- 显示电路

ESP32-Sense-Kit 开发套件的主板上有显示单元，可直观地反馈触摸动作。三个数码管分别显示被触摸按键的位置和触摸动作的持续时间。数码管驱动芯片为 CH455G，使用 I2C 接口控制。RGB 灯用于触摸时的颜色反馈。用户滑动滑条，RGB 灯的颜色会相应变化。

下图是显示单元的硬件原理图：

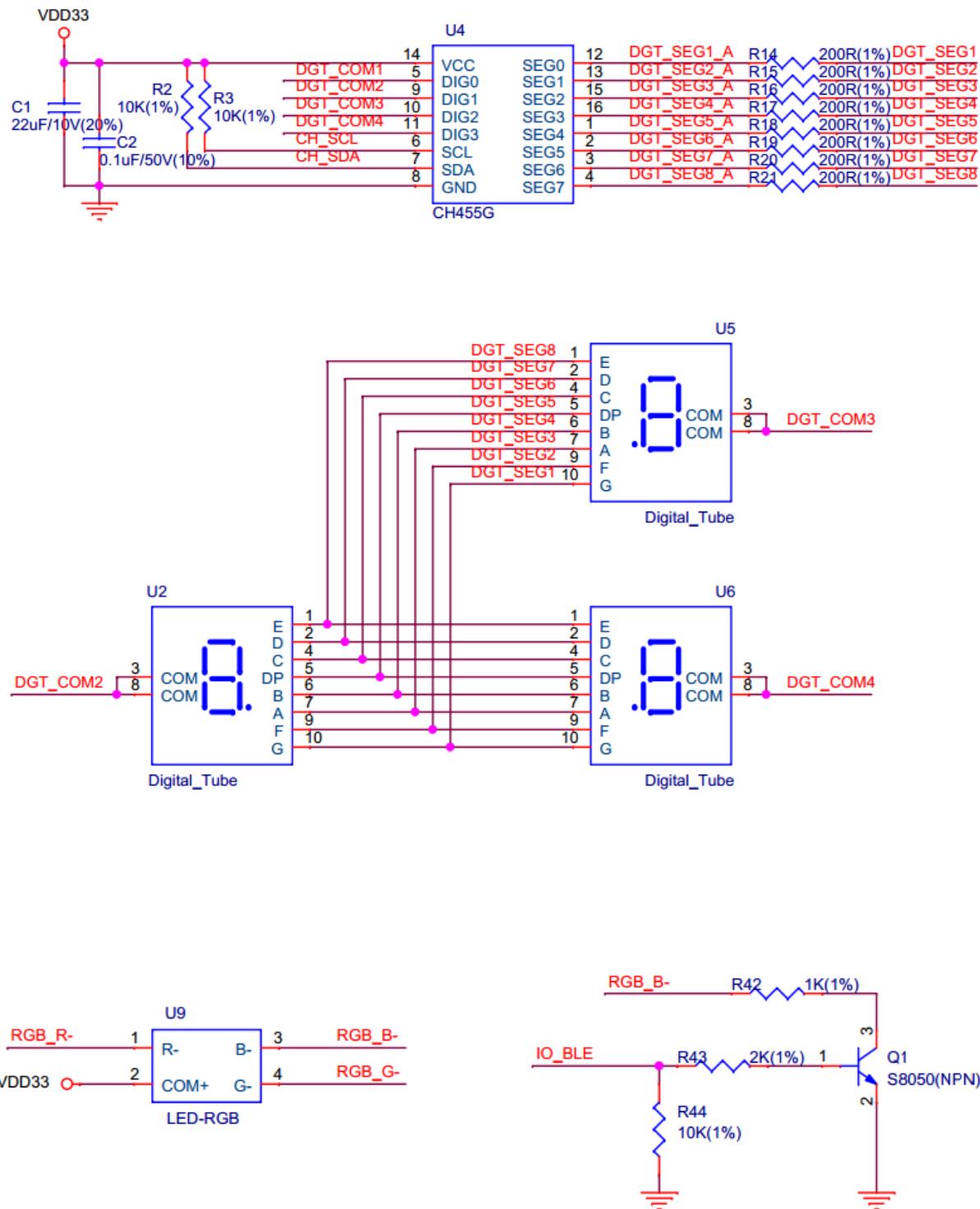


图 109: 显示电路

子板

- 子板分压电阻

子板上的触摸电极有多种形状和排列方式。每个子板上有分压电阻，每个分压电阻的阻值不同，主板应用程序通过 ADC 读取分压值，识别不同类型的子板。分压电路如下图所示：

| 子板 | 分压电阻 (Kohm) | ADC 读数 (Min) | ADC 读数 (Max) |
|------|-------------|--------------|--------------|
| 弹簧按键 | 0 | 0 | 250 |
| 线性滑条 | 4.7 | 805 | 1305 |
| 矩阵按键 | 10 | 1400 | 1900 |
| 双工滑条 | 19.1 | 1916 | 2416 |
| 轮式滑条 | 47 | 2471 | 2971 |

应用程序介绍

ESP32 IoT Solution 工程下的 [ESP32-Sense 项目](#) 是 ESP32-Sense 开发套件对应的应用程序。目录结构如下图所示：

```

├── main
|   ├── evb_adc.c           // 使用 ADC 功能识别不同子板，设置每个子板对应的 ADC 阈值
|   ├── evb.h                // 主板应用程序参数设置（触摸阈值，ADC I/O, I2C I/O）
|   ├── evb_led.cpp          // RGB LED 初始化程序
|   ├── evb_seg_led.c        // 数码管驱动程序
|   ├── evb_touch_button.cpp // 子板驱动程序-触摸按键
|   ├── evb_touch_wheel.cpp // 子板驱动程序-轮式滑条
|   ├── evb_touch_matrix.cpp // 子板驱动程序-矩阵按键
|   ├── evb_touch_seq_slide.cpp // 子板驱动程序-双工滑条
|   ├── evb_touch_slide.cpp  // 子板驱动程序-线性滑条
|   ├── evb_touch_spring.cpp // 子板驱动程序-弹簧按键
|   └── Kconfig.projbuild
|
|   └── main.cpp             // 主程序入口
|
└── Makefile
└── sdkconfig.defaults

```

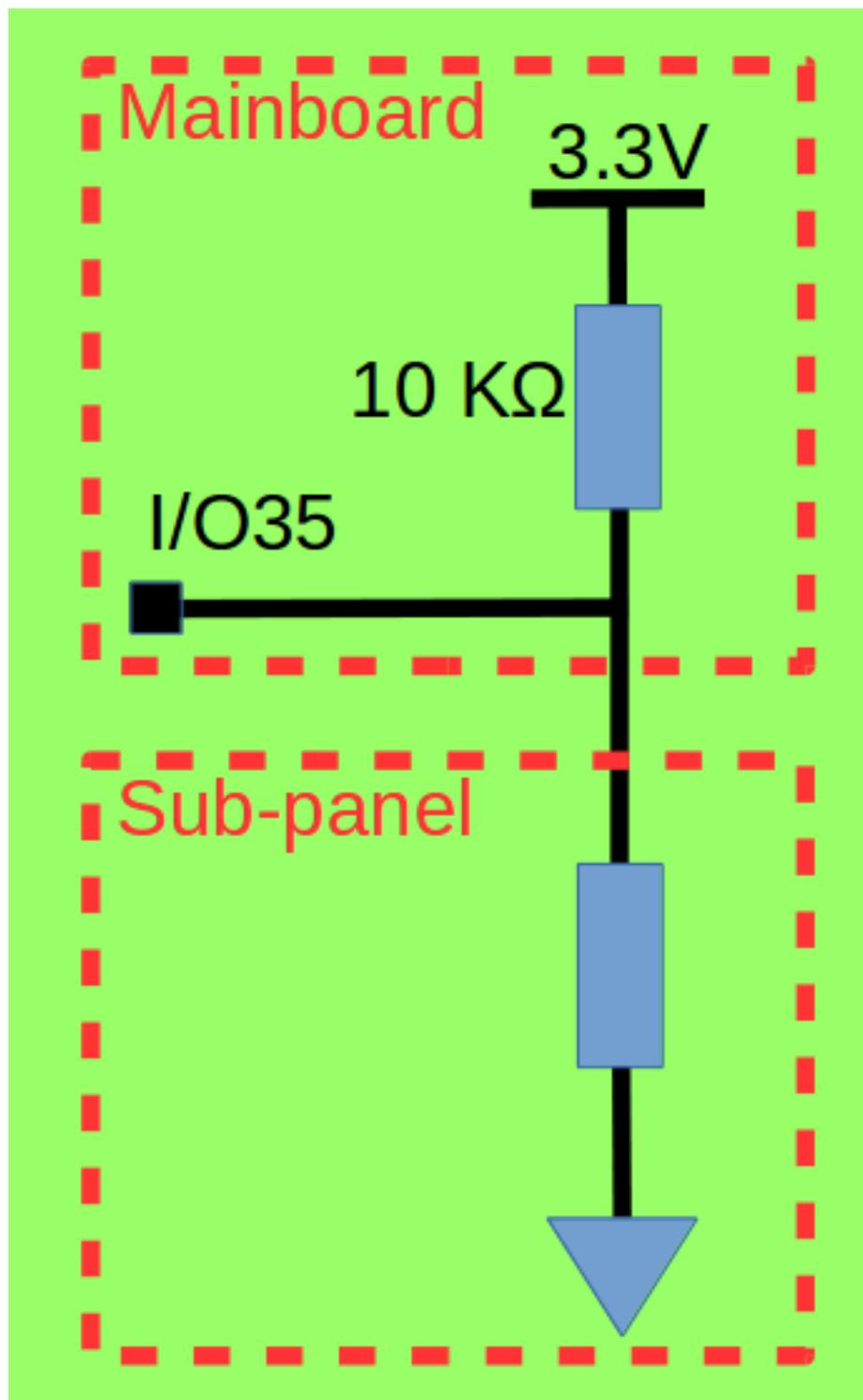


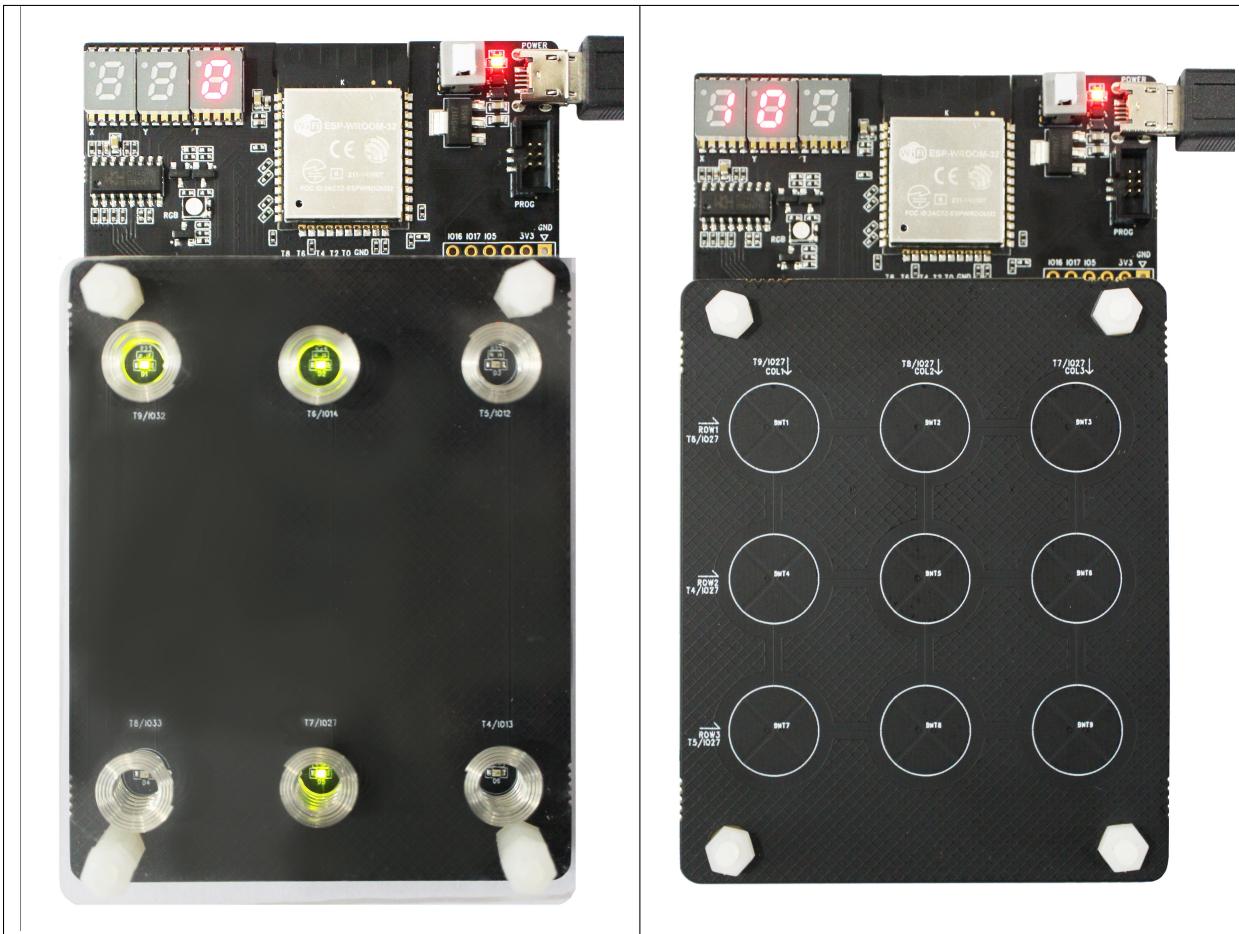
图 110: 分压电路
主板上的分压电阻是 $10\text{ K}\Omega$, 下表是各子板上对应的分压电阻阻值:

参数配置

当使用不同厚度或不同材质的盖板时，需要重新设置各通道触摸变化率参数，即灵敏度。各通道触摸变化率是由脉冲计数值计算得到。计算公式为：(Non-touch value - Touch value) / Non-touch value。“Non-touch value”是指不触摸时通道的脉冲计数值。“Touch value”是指正常触摸时通道的脉冲计数值。这两个参数需要用户测量得出。系统初始化时，由触摸变化率自动计算出触摸阈值，触摸阈值与触摸变化率成正比关系。

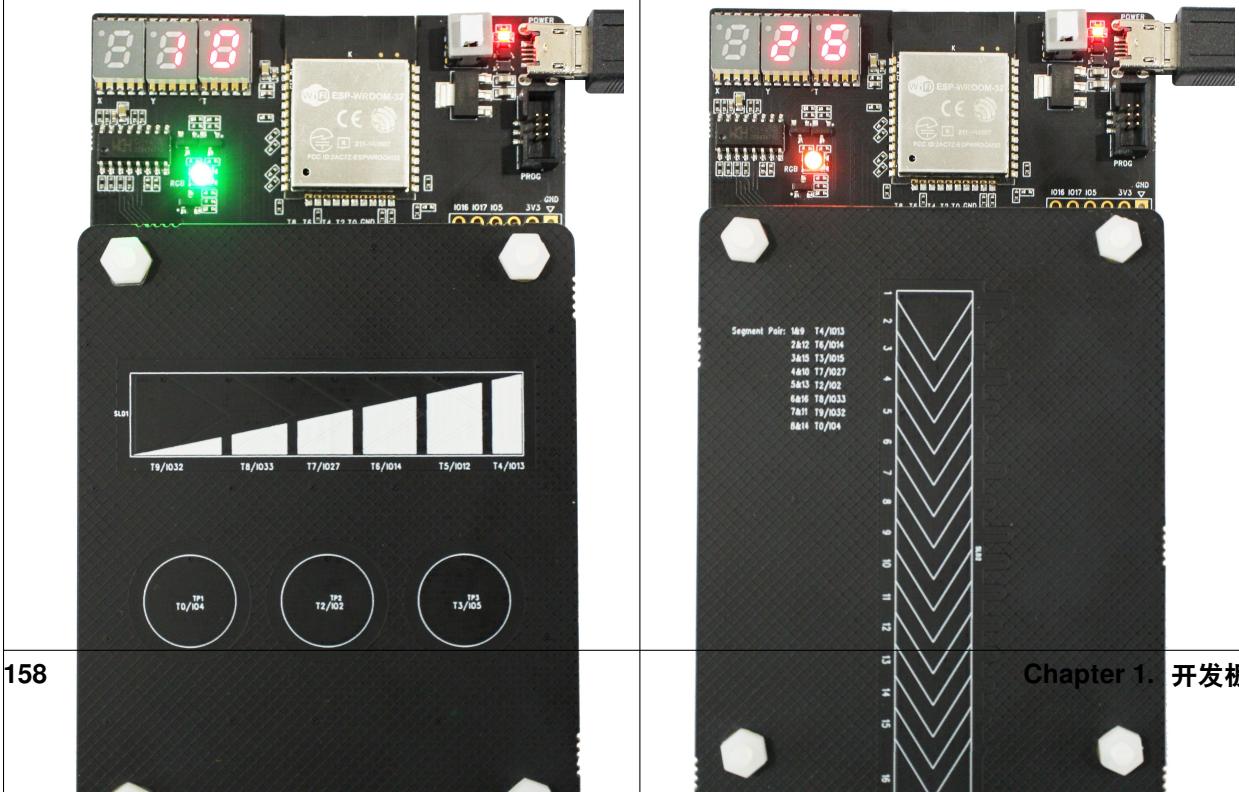
触摸变化率确定后，填写到 ESP32-Sense 工程中的 `evb.h` 文件。

效果展示



Spring Button

Matrix Button



相关资源

- 原理图
 - [ESP32-Sense-kit 主板原理图](#)
 - [ESP32-Sense-kit 子板原理图](#)

- PCB 布局
 - [ESP32-Sense-kit 主板 PCB 布局](#)
 - [ESP32-Sense-kit 子板 PCB 布局](#)

- 开发环境
 - [ESP-IDF](#) 是 ESP32 平台的软件开发包。文档 [Get Started](#) 介绍编译环境的搭建和软件开发包的使用说明。
 - [ESP-Prog](#) 是 ESP32 调试工具，有下载和 Jtag 调试功能。

- ESP32 IoT 应用方案
 - [ESP32 IoT Solution](#) 基于 ESP-IDF 开发，包含多种应用解决方案。文档 [build-system-and-dependency](#) 介绍了如何编译解决方案。
 - [ESP32-Sense 项目工程](#) 是 ESP32-Sense 开发套件对应的软件工程文件。下载程序到主板即可使用触摸功能。

- 硬件手册
 - [ESP32-Sense-Kit](#) 开发套件的硬件原理图、PCB 文件、BOM 等文件，请点击 [ESP32-Sense Kit 参考设计](#) 下载。

- 其他参考资料
 - [Espressif 官网](#)。
 - [ESP32 编程指南](#)：ESP32 相关开发文档的汇总平台，包含硬件手册、软件 API 介绍等。
 - [触摸传感器应用设计参考文档](#)：ESP32 触摸传感器功能应用设计手册，包括触摸传感器原理介绍、软件设计、PCB 设计等内容。

- 技术支持
 - 若在 ESP32-Sense-Kit 开发套件使用时遇到问题，请在 ESP32-Sense 工程中提交 [issue](#)。

- 购买方式
 - 微信公众号：[espressif_systems](#)
 - 商务咨询

1.1.13 ESP32-S2-Touch-Devkit-1

[中文]

ESP32-S2-Touch-Devkit-1 is a development kit that is aimed at helping evaluate and develop capacitive touch sensor applications on ESP32-S2. It is made up of a Motherboard-Subboard structure.

ESP32-S2-Touch-Devkit-1

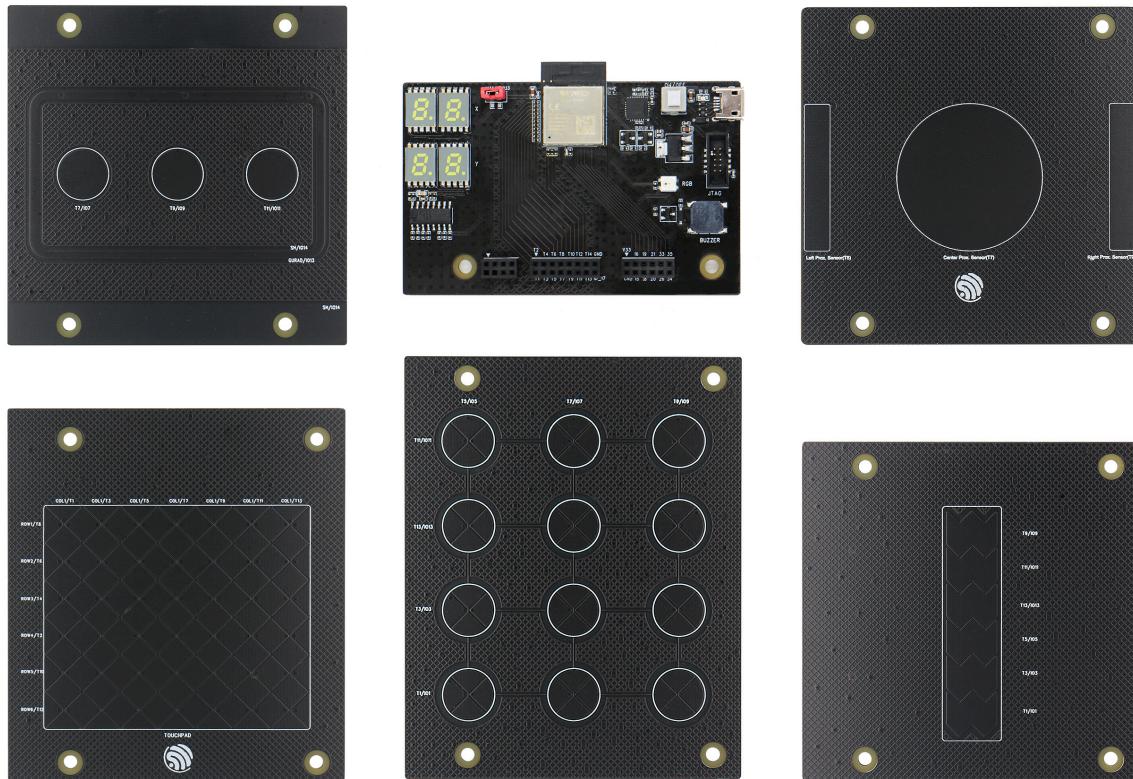


图 111: ESP32-S2-Touch-Devkit-1 Board-set

Overview

ESP32-S2-Touch-Devkit-1 is a development kit that is aimed at helping evaluate and develop capacitive touch sensor applications on ESP32-S2. It is made up of Motherboard-Subboard structure. The motherboard of ESP32-S2-Touch-Devkit-1 integrates [ESP32-S2-MINI-1](#) controller module and several useful little components such as buzzer, digital tube, RGB light, and so on. There are several kinds of subboards in ESP32-S2-Touch-Devkit-1 with different kinds of capacitive touch sensor pads, developers can choose one of them and connect it with motherboard so that they could

develop different kinds of capacitive touch sensor applications. The motherboard and subboard use the pin header/female pin header as the socket connector which makes it plug in and plug out smoothly.

Motherboard

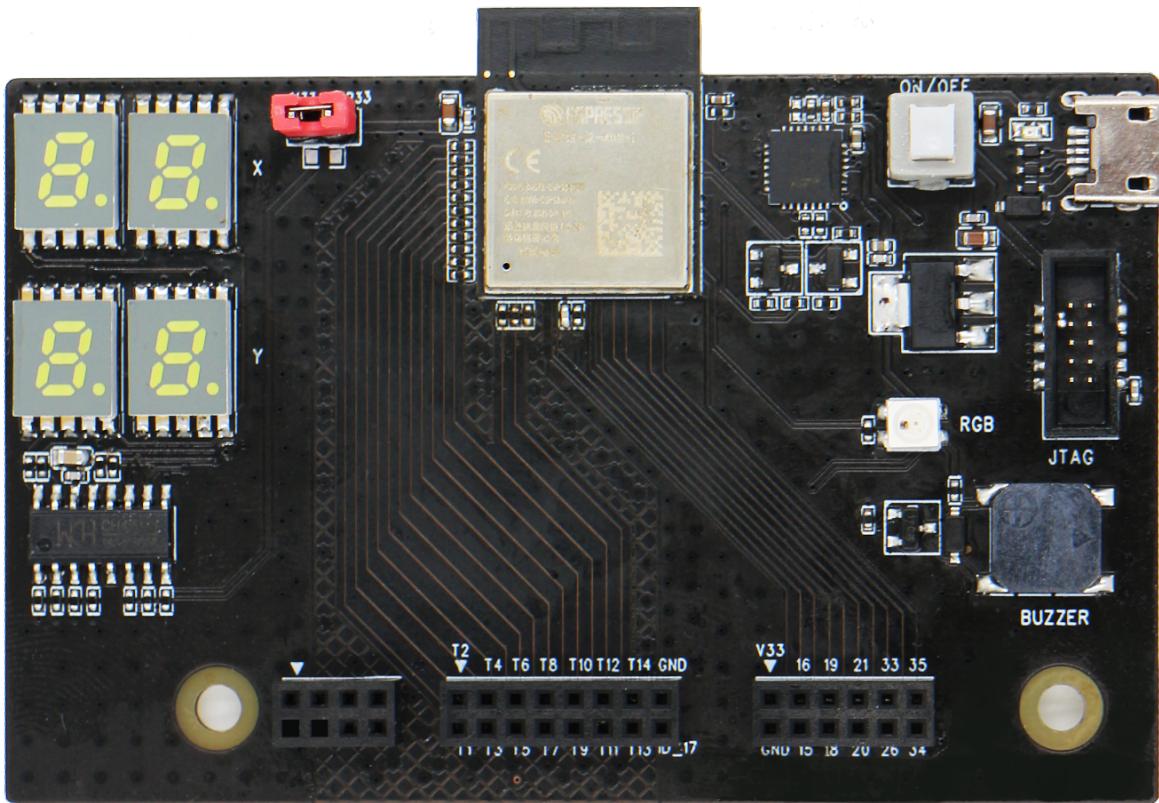
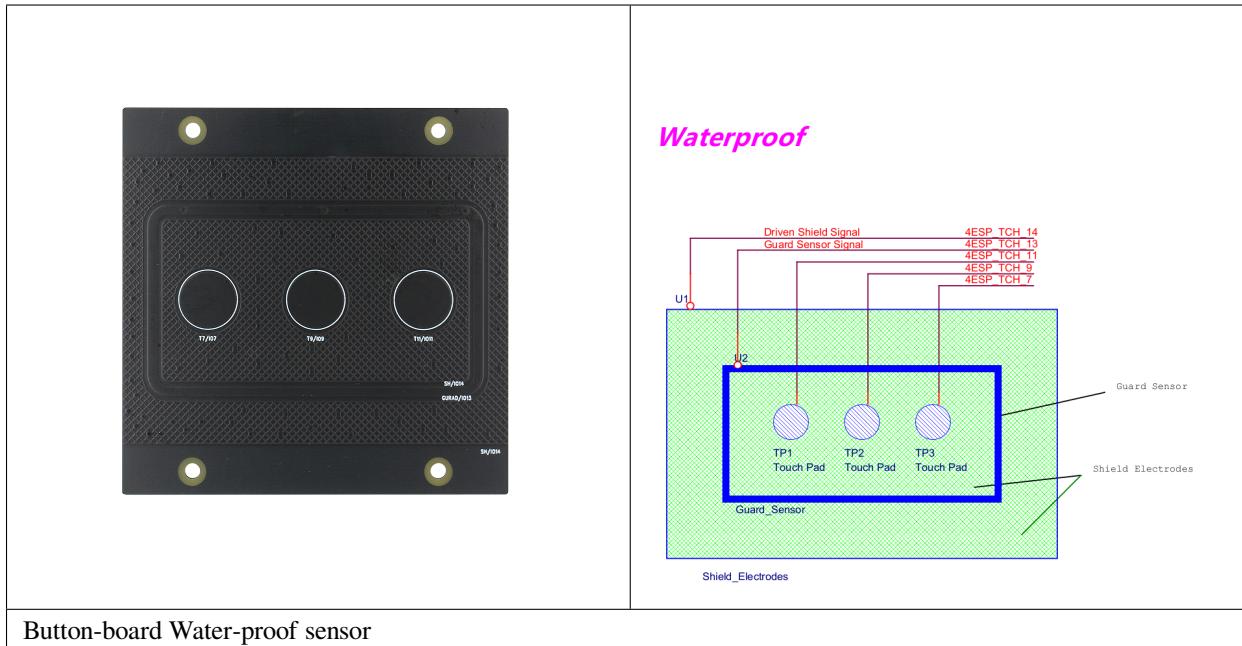


图 112: Motherboard

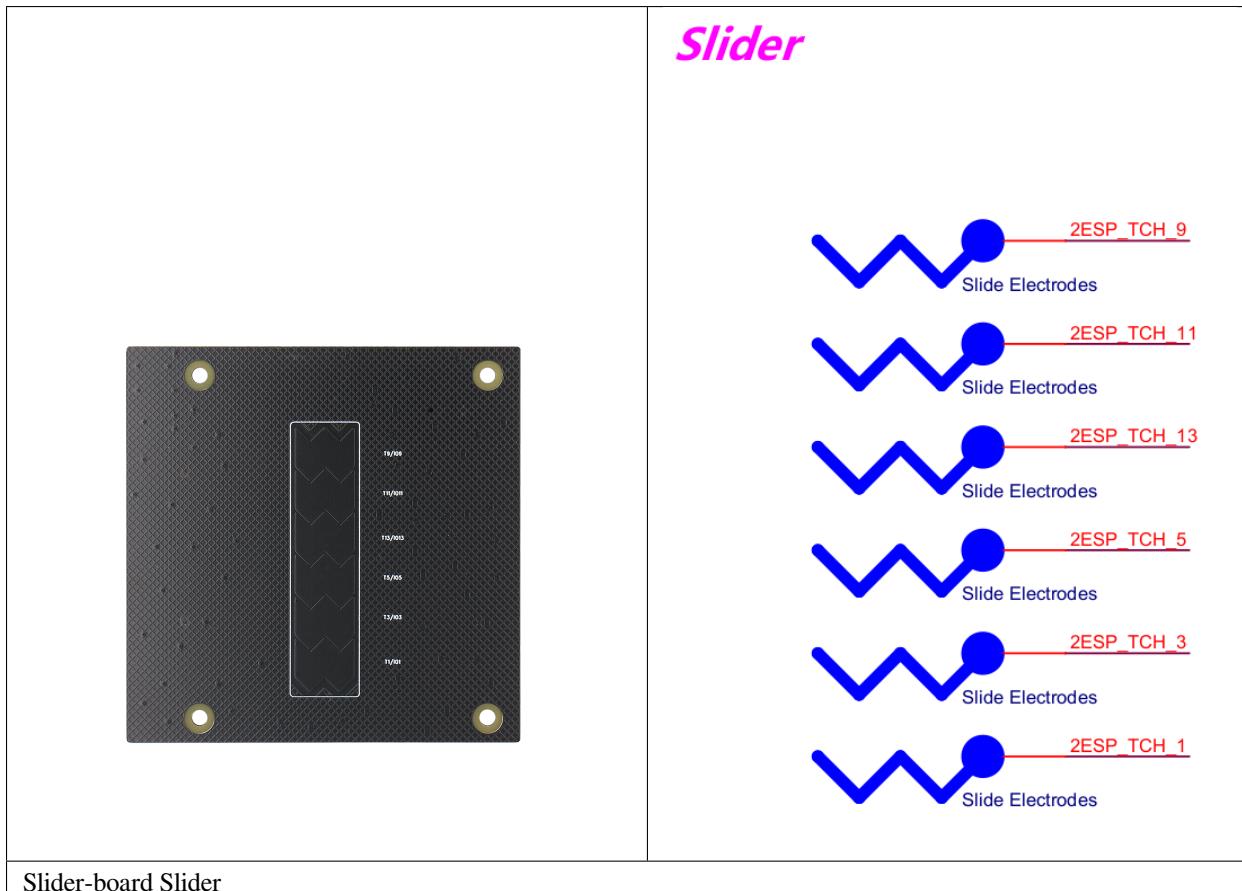
Subboards

- Button-board: Three capacitive touch buttons with waterproof sensor.



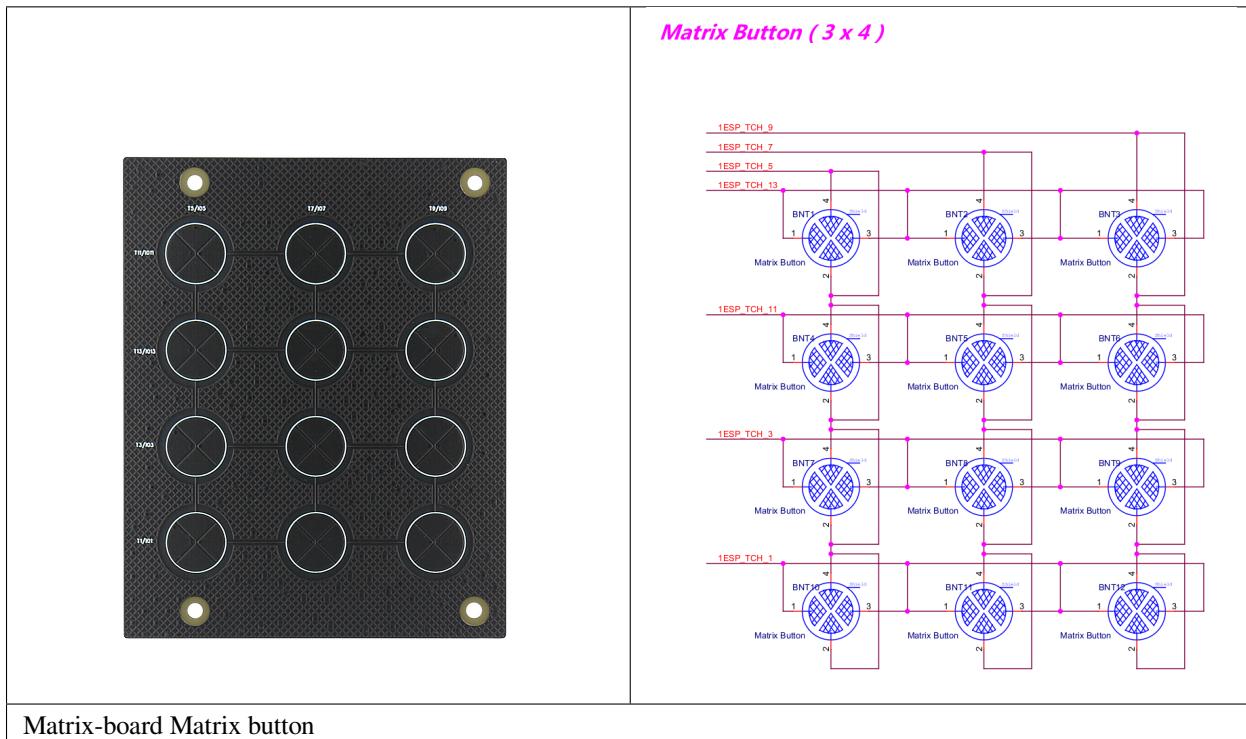
Button-board Water-proof sensor

- Slider-board: Capacitive touch linear slider, the relative distance of slider has up to 8-bit precision.

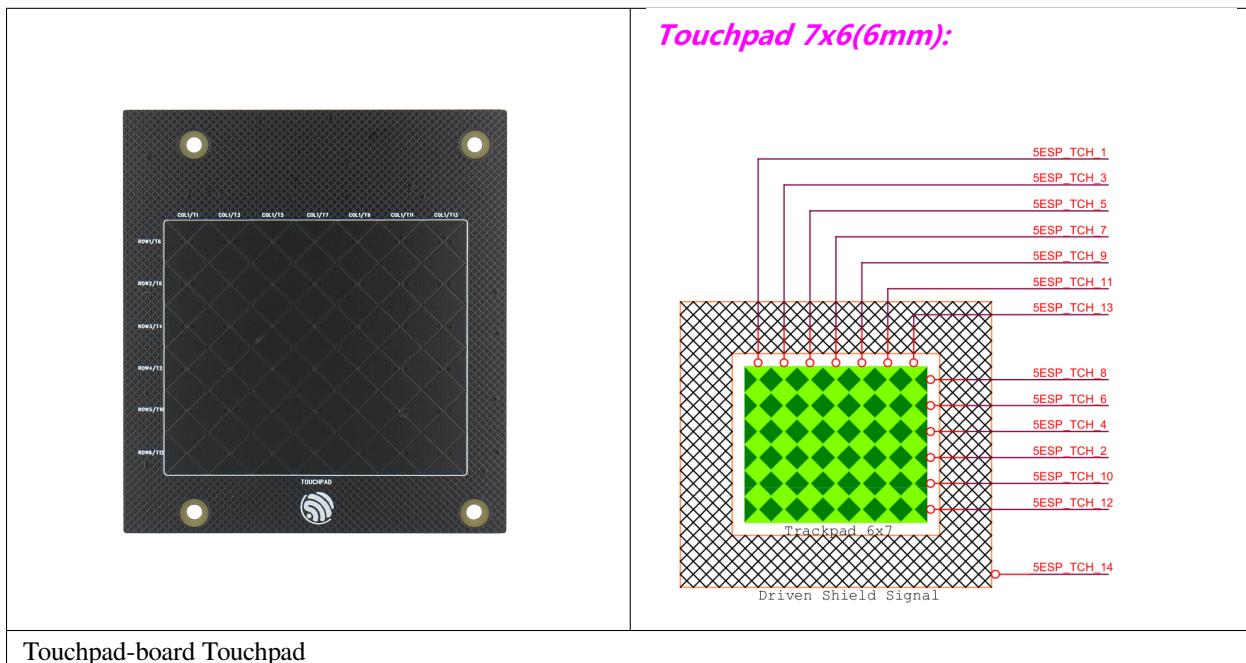


Slider-board Slider

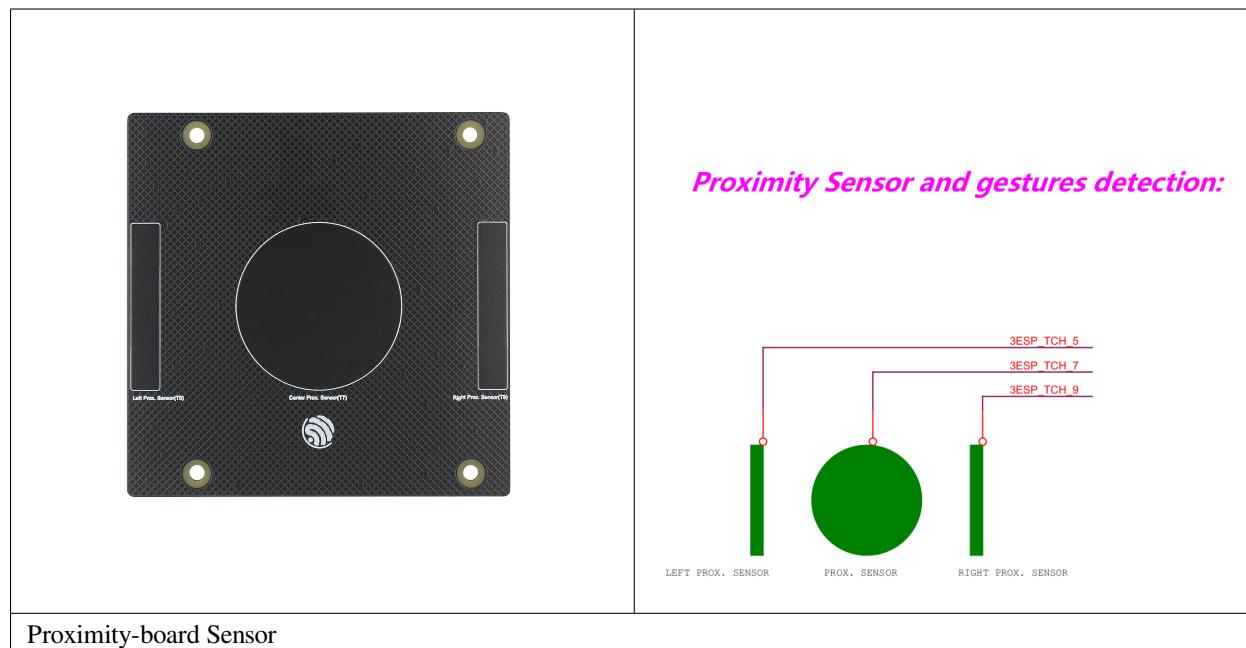
- Matrix-board: 3 x 4 capacitive touch matrix button, 7 channels of touch sensor make up of 12 capacitive touch buttons.



- Touchpad-board: 7 x 6 two-dimension capacitive touchpad.



- Proximity-board: Three capacitive touch proximity sensors.



Get Started

Install SDK and it's dependencies:

1. Install ESP-IDF, please refer to [ESP-IDF Programming Guide](#).
2. Use the release-4.3 version of ESP-IDF.

```
cd esp-idf #Enter esp-idf folder  
git checkout release/v4.3 #Checkout release-v4.3 version of ESP-IDF  
git submodule update #Maybe is needed
```

Get ESP32-S2-Touch-Devkit-1 project:

1. Clone esp-dev-kits repo.

```
git clone https://github.com/espressif/esp-dev-kits.git
```

2. Enter ESP32-S2-Touch-Devkit-1 project folder.

```
cd esp-dev-kits/esp32-s2-touch-devkit-1
```

The project's structure is shown as followed:

```
|--- CMakeLists.txt  
|--- components #components' driver  
|   |--- board_detect #subboard detector
```

(下页继续)

(续上页)

```

|   └── buzzer           #buzzer driver
|   └── digital_tube     #digital tube controller driver
|   └── rgb_light         #RGB light driver(ws2812)
|   └── subboards         #Subboards' application source file
|       └── touch_element #Touch element library
└── main                #Main application demo logic
    ├── CMakeLists.txt
    └── main.c

```

Build & Flash ESP32-S2-Touch-Devkit-1 demo project:

1. Enable ESP-IDF environment variable.

```

cd esp-idf
./export.sh

```

2. Build & Flash.

```

cd esp-dev-kits/esp32-s2-touch-devkit-1
idf.py set-target esp32s2 #Enable esp32s2 platform
idf.py build flash

```

3. Monitor log output.

```
idf.py monitor
```

Example output:

```

I (2880) Touch Demo: Slider sub-board plug in
I (22480) Touch Demo: Slider sub-board plug out
I (22480) Touch Demo: Nothing detected
I (41540) Touch Demo: Touchpad sub-board plug in
I (47700) Touchpad Board: Touchpad pressed, position: [0, 5]
I (47710) Touchpad Board: Position: [0, 5]
I (47720) Touchpad Board: Position: [0, 5]
I (47730) Touchpad Board: Position: [0, 6]
I (47740) Touchpad Board: Position: [0, 6]
I (47750) Touchpad Board: Position: [0, 6]
I (47760) Touchpad Board: Position: [0, 6]
I (47770) Touchpad Board: Position: [0, 7]
I (47780) Touchpad Board: Position: [0, 8]
I (47790) Touchpad Board: Position: [0, 9]
I (47800) Touchpad Board: Position: [0, 9]
I (47810) Touchpad Board: Position: [1, 10]
I (47820) Touchpad Board: Position: [2, 11]

```

(下页继续)

(续上页)

```
I (47830) Touchpad Board: Position: [2, 12]
I (47840) Touchpad Board: Position: [3, 13]
I (47850) Touchpad Board: Position: [4, 14]
I (47860) Touchpad Board: Position: [5, 15]
I (47870) Touchpad Board: Position: [6, 16]
I (47880) Touchpad Board: Position: [7, 16]
I (47890) Touchpad Board: Position: [9, 17]
I (47900) Touchpad Board: Position: [10, 18]
I (47910) Touchpad Board: Position: [11, 18]
I (47920) Touchpad Board: Position: [11, 19]
I (47930) Touchpad Board: Position: [12, 20]
I (47940) Touchpad Board: Position: [13, 21]
I (47950) Touchpad Board: Position: [14, 21]
I (47960) Touchpad Board: Position: [14, 22]
I (47970) Touchpad Board: Position: [15, 23]
I (47980) Touchpad Board: Position: [15, 23]
I (47990) Touchpad Board: Position: [15, 24]
I (48000) Touchpad Board: Position: [15, 24]
I (48010) Touchpad Board: Position: [15, 24]
I (48020) Touchpad Board: Position: [16, 24]
I (48030) Touchpad Board: Position: [16, 24]
I (48040) Touchpad Board: Position: [16, 24]
I (48050) Touchpad Board: Position: [16, 23]
I (48060) Touchpad Board: Position: [16, 23]
I (48070) Touchpad Board: Position: [16, 22]
I (48080) Touchpad Board: Position: [16, 22]
I (48090) Touchpad Board: Position: [16, 21]
I (48100) Touchpad Board: Touchpad released, position: [16, 21]
```

Notes

- Some new Touch Sensor features (Touchpad, Touch proximity) are not supported in ESP-IDF Touch Element library, so we copy Touch Element from ESP-IDF components into this demo project's components' folder and add the necessary features. They will appear in the future version of ESP-IDF.

Troubleshooting

Q1: Why Proximity-board is connected with Motherboard, they don't work or work abnormally?

A1: Though all of those subboards are hot-swappable theoretically, the Proximity-board needs to startup in an ideal environment(Far away from your hands). If it goes wrong, you can reset it mandatorily by releasing the power-switch.

Related Documents

Schematic

- [ESP32-S2-Touch-Devkit-1 Motherboard Schematic](#)
- [Button Subboard Schematic](#)
- [Slider Subboard Schematic](#)
- [Matrix Button Subboard Schematic](#)
- [Touchpad Subboard Schematic](#)
- [Proximity Subboard Schematic](#)

Other Documents

- [Touch Element Library Programming Guide](#)
- [ESP32-S2-MINI-1 Datasheet](#)
- [ESP32-S2 Datasheet](#)
- [ESP32-S2 Technical Reference Manual](#)

1.1.14 ESP32-MeshKit-Sense

[[English](#)]

ESP32-MeshKit-Sense 是一款以乐鑫 ESP32 模组为核心的开发板，集成了温湿度传感器、环境亮度传感器等外设，并且可外接屏幕，主要用于检测模组在正常工作或睡眠状态下，连接各个外设时的电流情况。

ESP32-MeshKit-Sense

[English]

产品概述

ESP32-MeshKit-Sense 是一款以乐鑫 ESP32 模组为核心的开发板，集成了温湿度传感器、环境亮度传感器等外设，并且可外接屏幕，主要用于检测模组在正常工作或睡眠状态下，连接各个外设时的电流情况。

关于 ESP32 详细信息，请参考文档《[ESP32 技术规格书](#)》。

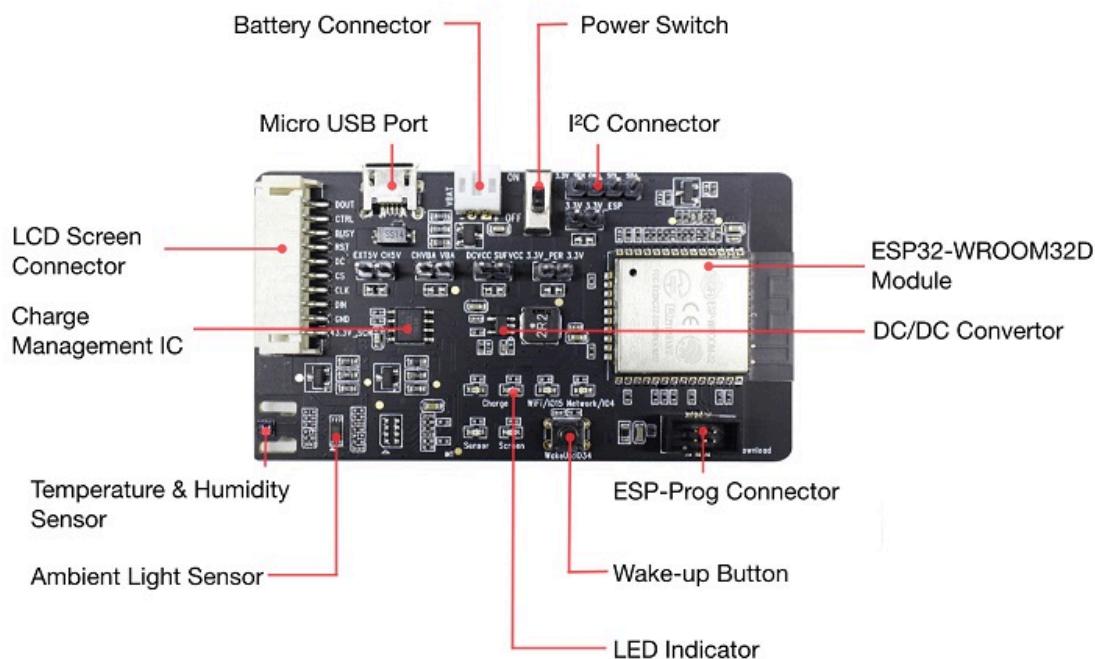


图 113: ESP32-MeshKit-Sense

电路设计说明

系统框图

ESP32 的系统框图如下图所示。

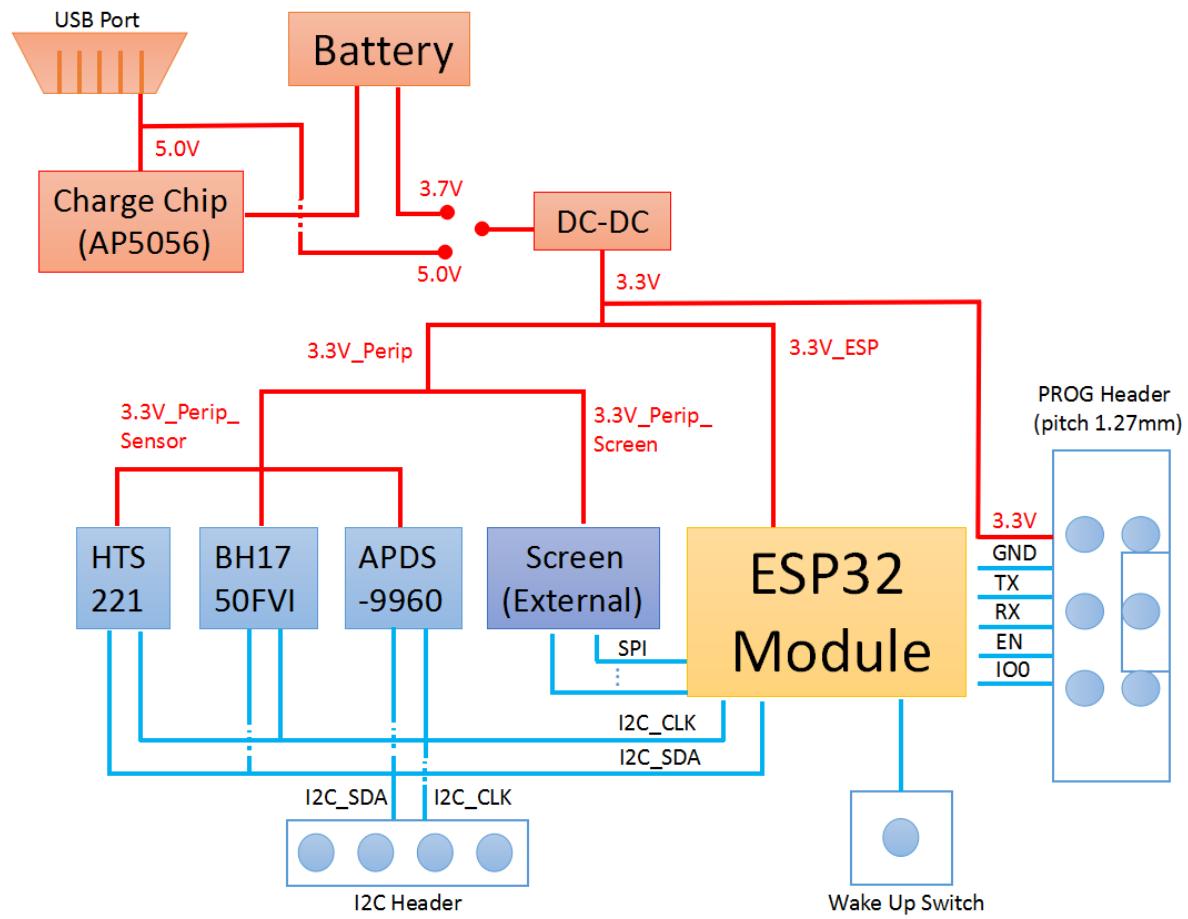


图 114: ESP32 系统框图

PCB 布局

PCB 布局如下图所示。

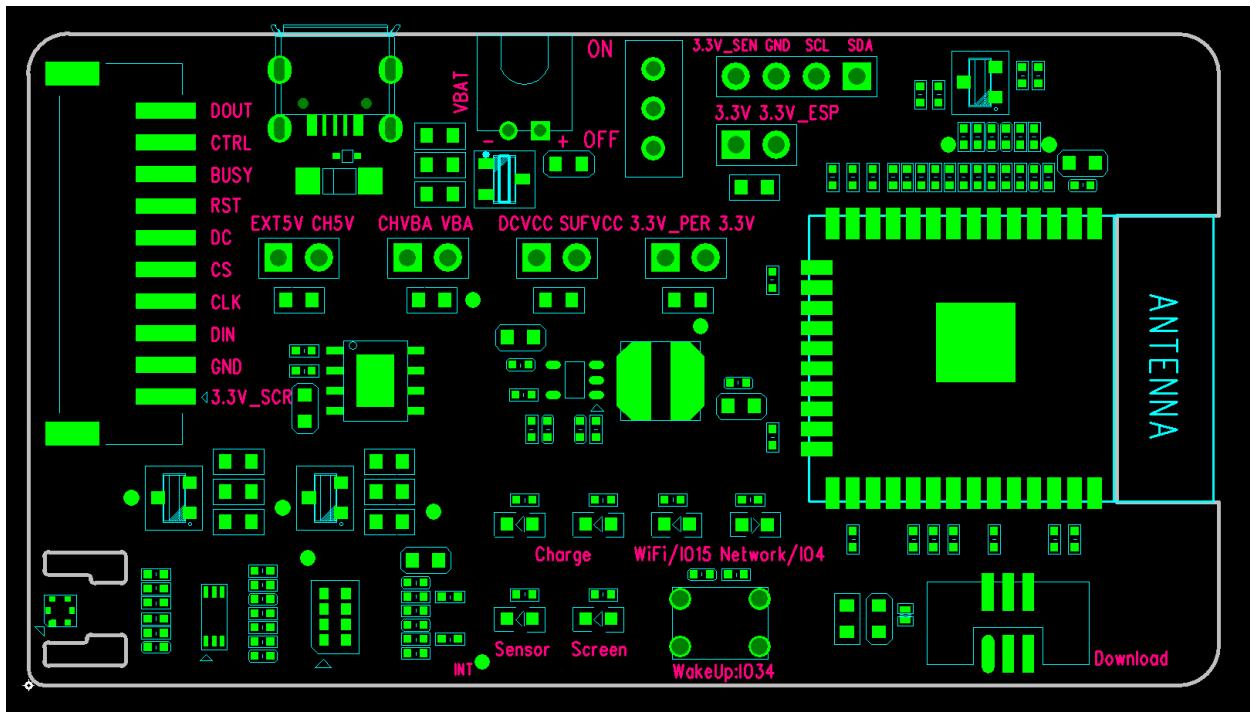


图 115: PCB 布局

PCB 部件功能说明如下表所示：

| PCB 部件 | 说明 |
|---------------|--|
| EXT5V | USB 输入的 5 V |
| CH5V | 充电管理芯片的输入 |
| CHVBA | 充电管理芯片的输出 |
| VBA | 接至电源的正极 |
| SUFVCC | 当开关处于“ON”档位时，与电源输入接通，当开关处于“OFF”档位时，与电源输入断开 |
| DCVCC | 电源管理芯片 DC-DC 的输入 |
| 3.3V | 电源管理芯片的输出，即总路 3.3 V |
| 3.3V_PER | 为所有外设供电的 3.3V_Perip |
| 3.3V_ESP | 为 ESP32 模组模块供电的 3.3V_ESP |
| 3.3V_SEN | 为三款传感器供电的 3.3V_Perip_Sensor |
| 3.3V_SCR | 为外接屏幕供电的 3.3V_Perip_Screen |
| Charge | 电池充电指示灯，D5 为红灯，表示正在充电；D6 为绿灯，表示充电完成 |
| Sensor | 电源指示灯，表示 3.3V_Perip_Sensor 已接通 |
| Screen | 电源指示灯，表示 3.3V_Perip_Screen 已接通 |
| WiFi / IO15 | 信号指示灯，表示 Wi-Fi 正常工作 |
| Network / IO4 | 信号指示灯，表示与服务器连接正常 |

硬件模块

本章主要介绍各个功能模块（接口）的硬件实现，以及对这些模块的描述。

电源管理

USB/BAT 供电管理模块

开发板支持电池供电，电源管理芯片 AP5056 可对电池进行充电。AP5056 是一款单片锂离子电池恒流/恒压线性电源管理芯片。高达 1 A 的可编程充电电流，预设充电电压为 4.2 V。

而当 USB 供电与电池供电同时存在时，系统的选择会如下图所示：VBUS 为高，Q4 处于截止状态，VBAT（电池电源）自动与系统电源切断，USB 成为了系统的供电之选。

USB/电池供电管理模块电路图如下图所示。

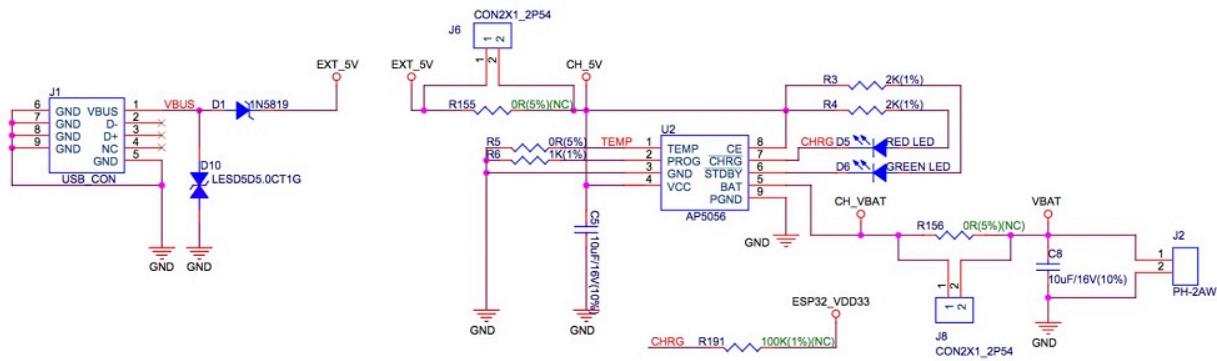
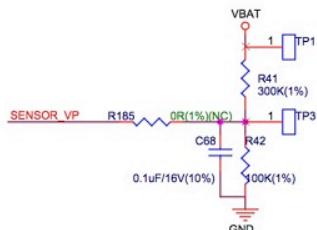
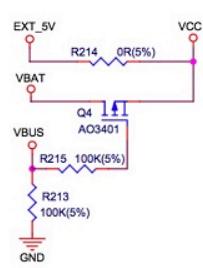
Power:**Power Switch:**

图 116: USB/电池供电管理模块

外设电源管理模块

首先，源自 USB 或 BAT 的输入需要通过电源管理芯片生成电路所需的 3.3 V。开发板采用了 ETA3425，其输出电压为 3.3 V，最大输出电流为 600 mA。

外设电池电路如下图所示。

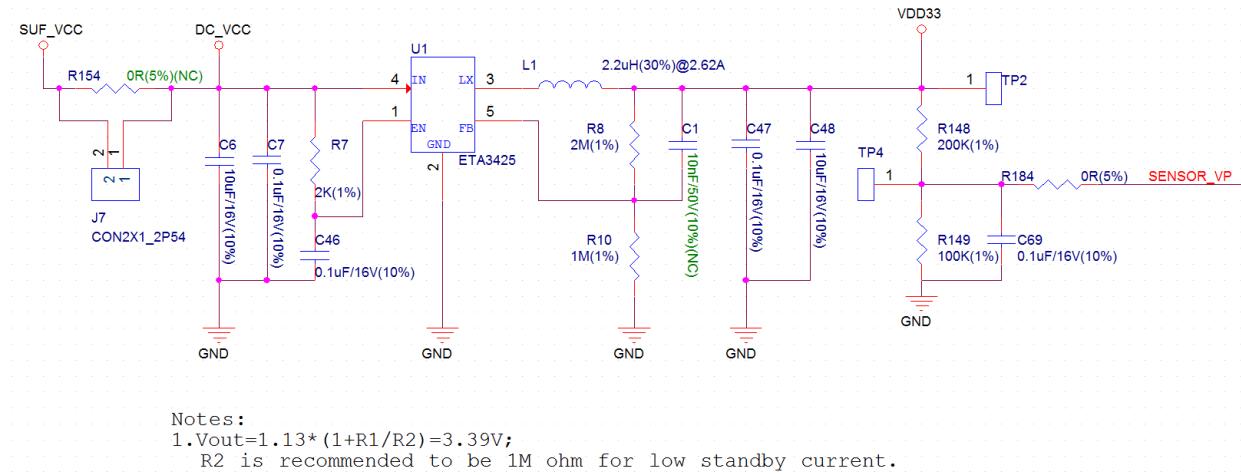


图 117: 外设电池电路

总路 VDD33 有两路分支：

- 专为 ESP32 模组模块供电的 ESP32_VDD33
- 专为所有外设供电的 VDD33_Perip

二者的连接与否都可以通过排针及跳线帽进行控制，ESP32_VDD33 原理图如下图所示。

其中，VDD33_Perip 也有两路分支：

- 专为外接屏供电的 VDD33_Perip_Screen
- 专为三款传感器供电的 VDD33_Perip_Sensor

二者的连接与否都可以通过模组 GPIO+MOS 管进行控制。VDD33_Perip 原理图如下图所示。

Boot & UART 功能

开发板采用一款插座 PROG Header，可通过排线连接至另一款 ESP-PROG 开发板上，然后再将 ESP-PROG 开发板中的 micro USB 接口与 PC 机相连，即可利用 PC 机对此开发板进行下载及调试。

Boot & UART 电路原理图如下图所示。

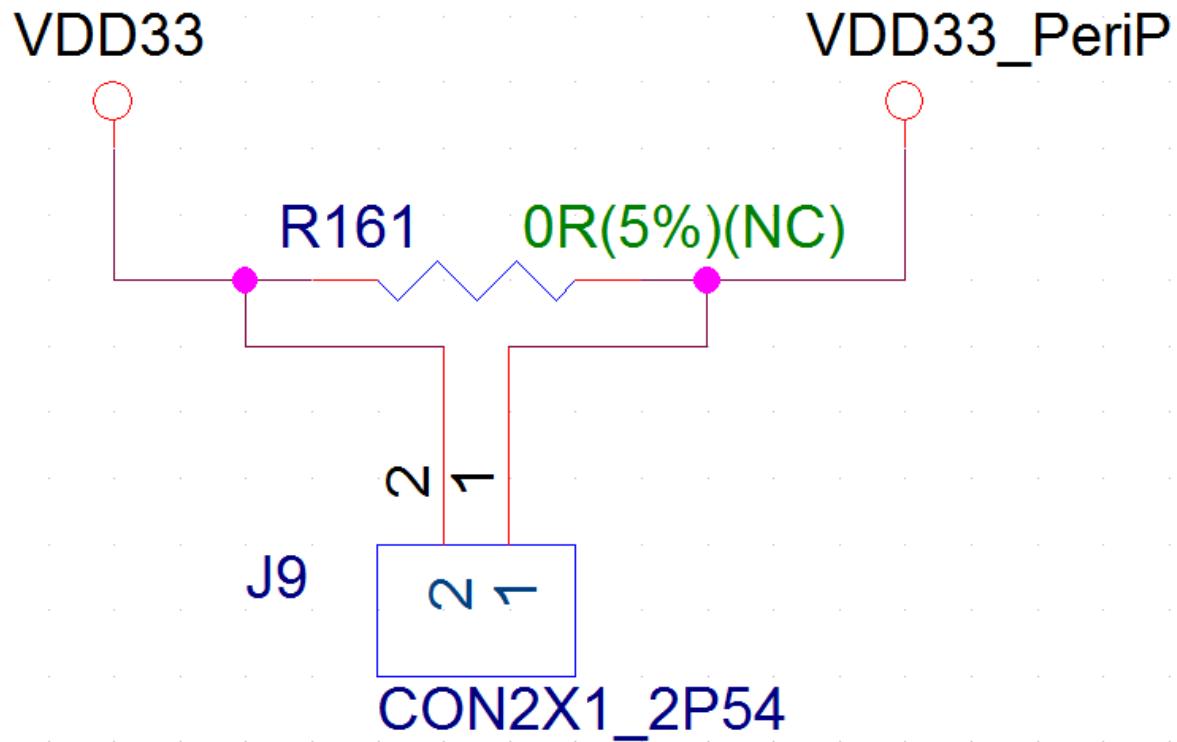


图 118: ESP32_VDD33 原理图

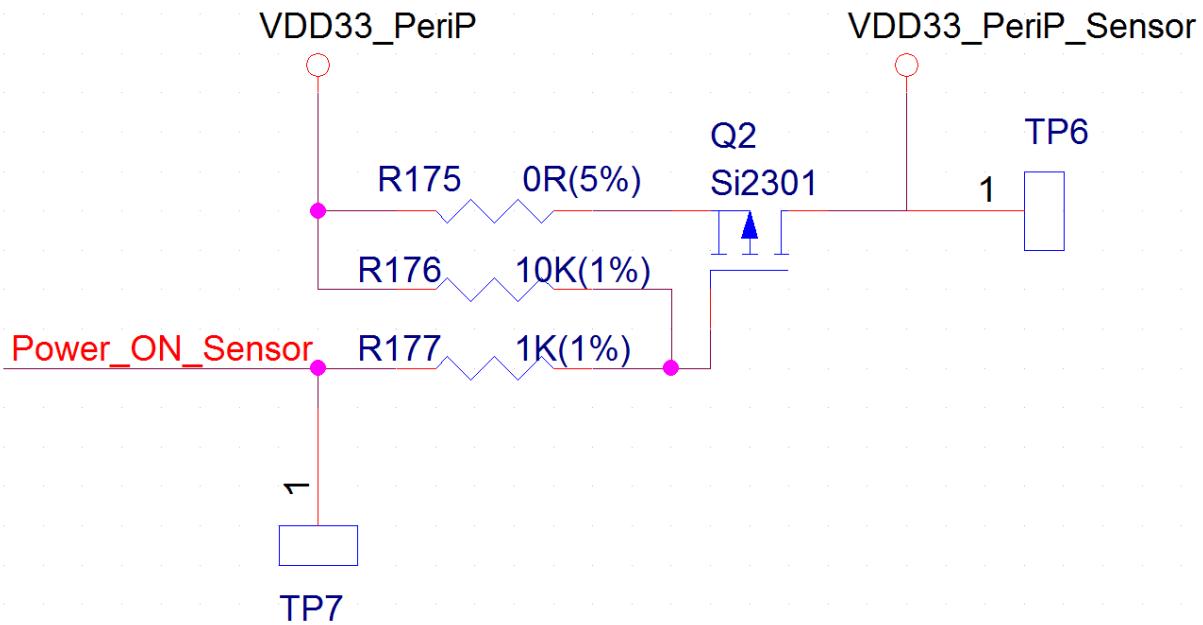


图 119: VDD33_Perip 原理图

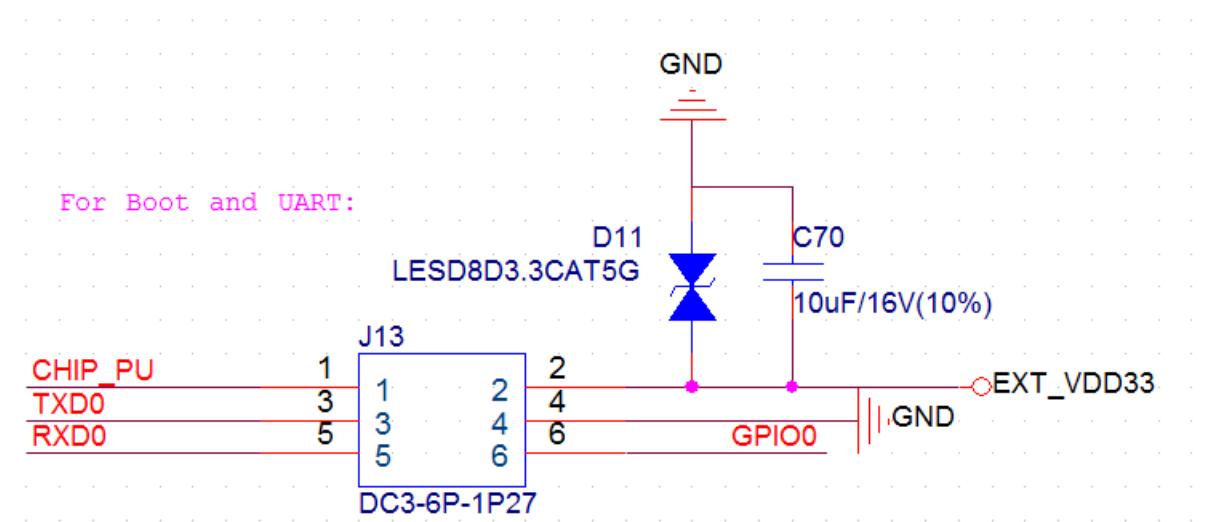


图 120: Boot & UART 电路原理图

睡眠唤醒模块

开发板采用了一个实体按键，IO34 是一个 RTC 域中的管脚。当芯片处于睡眠时，可以利用此按键的操作来实现芯片的唤醒。

睡眠唤醒模块电路原理图如下图所示。

外接屏幕

开发板采用一款可以外接屏幕的连接插座，利用排线可以将不同屏幕接至开发板上，以实现 ESP32 模组对屏幕的操作。

外接屏幕电路原理图如下图所示。

传感器

湿温度传感器

HTS221 是一种超小型相对湿度和温度传感器。开发板采用 3.3 V 供电，以及 I2C 的接口方式。

温湿度传感器电路原理图如下图所示。

Switch(Wake up) :

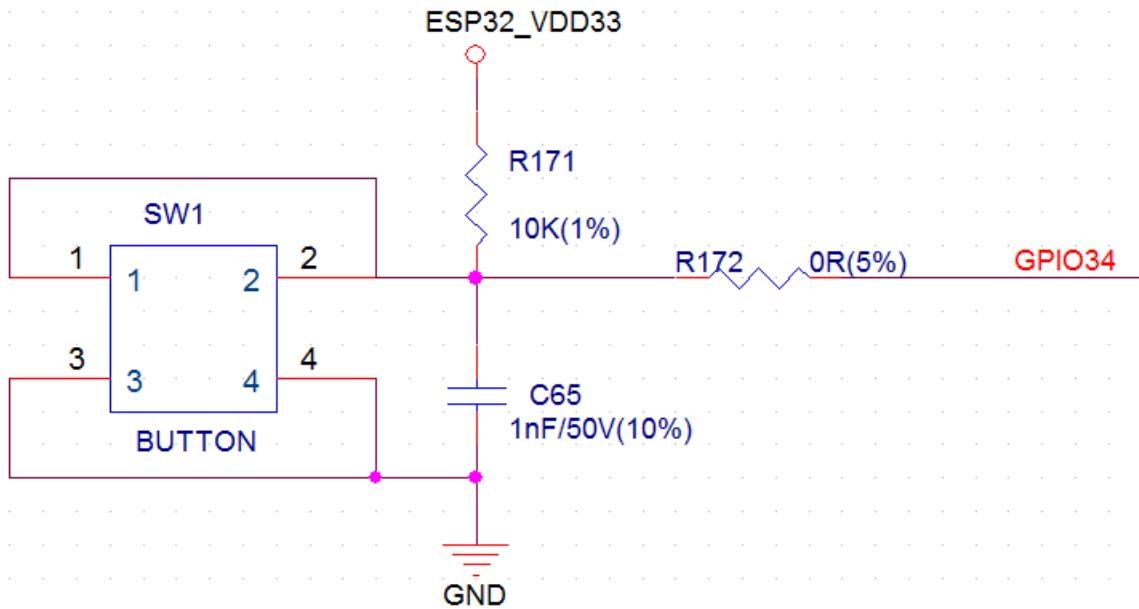


图 121: 睡眠唤醒模块电路原理图

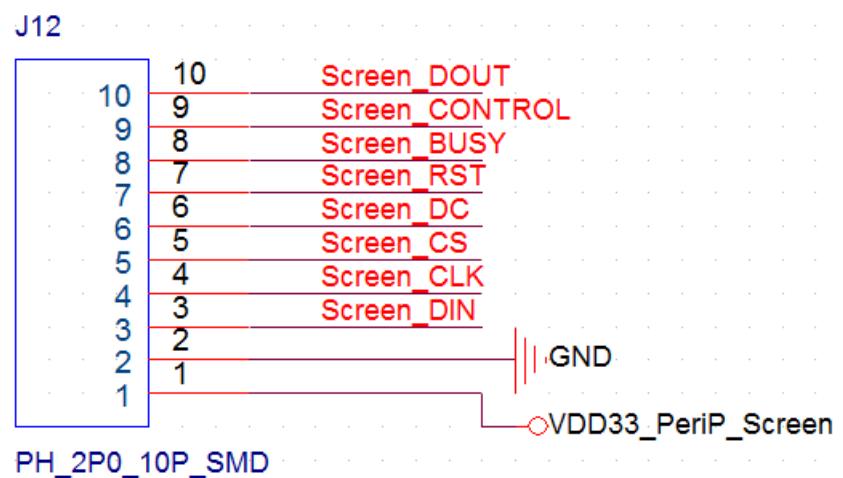
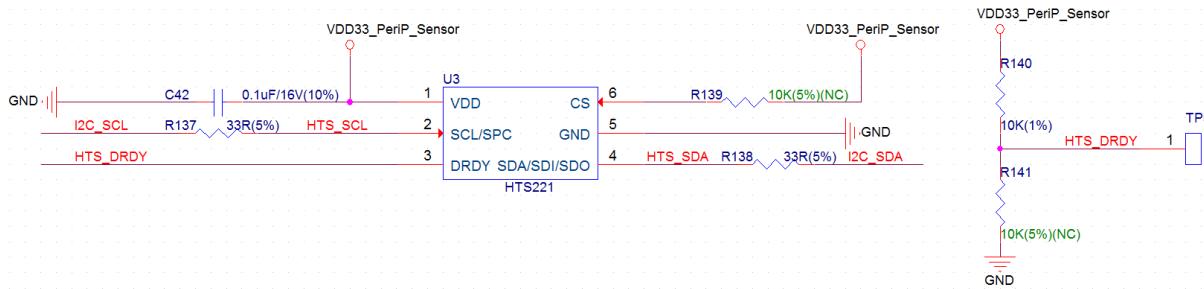


图 122: 外接屏幕电路原理图

Temperature&Humidity Sensor:



- 1.I₂C slave address:0'b 101 1111 x;
- 2.To select I₂C mode, the CS line must be tied high(i.e. connected to VDD) or left unconnected (thanks to the internal pull-up).
- 3.CTRL_REG3(22h) bit7:DRDY_H_L:data ready output signal active high,low(0:active-high,default;1:active low).

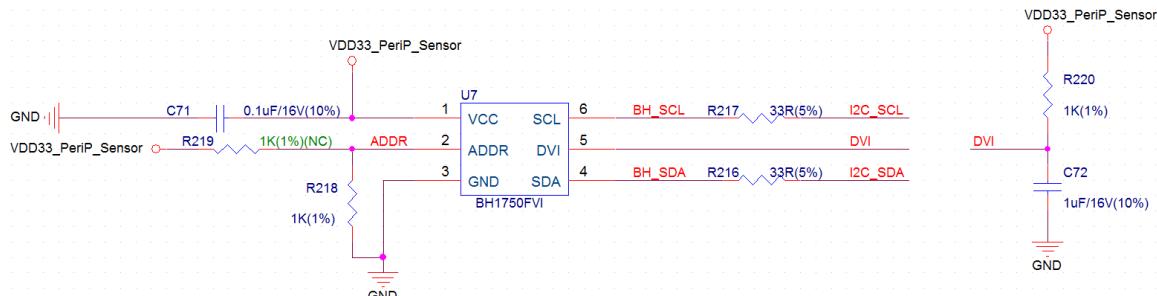
图 123: 温湿度传感器电路原理图

环境光传感器

BH1750FVI 是一款数字的环境光传感器。开发板采用 3.3 V 供电，以及 I₂C 的接口方式。

环境光传感器电路原理图如下图所示。

Ambient Light Sensor:



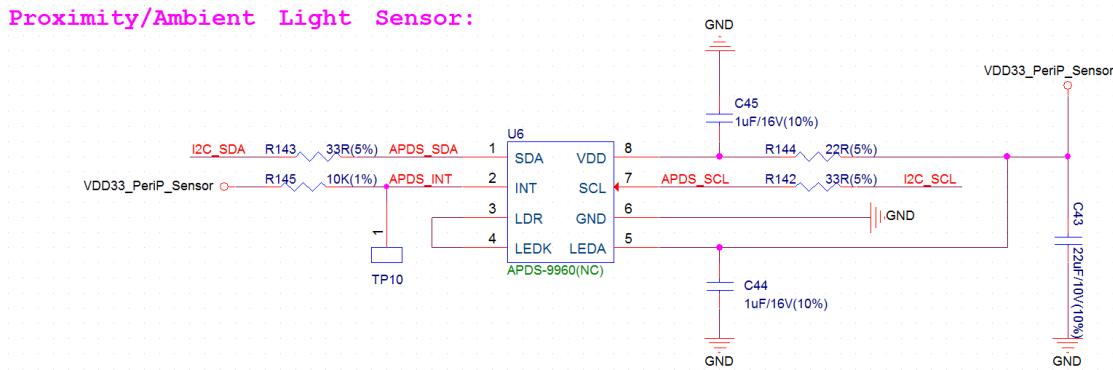
- 1.I₂C slave address: 0'b 101 1100 x (ADDR='H'); 0'b 010 0011 x (ADDR='L');
- 2.DVI is I₂C bus reference voltage terminal. And it is also asynchronous reset terminal. It is necessary to set to 'L' after VCC is supplied (At least 1us,DVI<=0.4V).

图 124: 环境光传感器电路原理图

环境亮度传感器

APDS-9960 是一款集成 ALS、红外 LED 和接近检测器的光学模块和环境亮度感测的环境亮度传感器。开发板采用 3.3 V 供电，以及 I2C 的接口方式。需说明的是，此款传感器当前设计中默认为不上件的状态。

环境亮度传感器电路原理图如下图所示。



- 1.I₂C slave address:0'b 011 1001 x;
- 2.It is recommended that LEDA pin can be connected to a seperate power supply from VDD.
But if operating from a single supply,use a 22R resistor in series with the VDD supply line and a 1uF low ESR capacitor to filter any power supply noise.

图 125: 环境亮度传感器电路原理图

示例程序

见 esp-mdf/examples/development_kit/sense。

相关文档

- [ESP32-MeshKit-Sense 原理图](#)
- [ESP32-MeshKit-Sense PCB 布局](#)

1.1.15 Contributions Guide

We welcome contributions to the esp-dev-kits project!

How to Contribute

Contributions to esp-dev-kits - fixing bugs, adding features, adding documentation - are welcome. We accept contributions via [Github Pull Requests](#).

Before Contributing

Before sending us a Pull Request, please consider this list of points:

- Is the contribution entirely your own work, or already licensed under an Apache License 2.0 compatible Open Source License? If not then we unfortunately cannot accept it.
- Does any new code conform to the esp-dev-kits : [Style Guide](#) ?
- Does the code documentation follow requirements in [Documenting-code](#) ?
- Is the code adequately commented for people to understand how it is structured?
- Are comments and documentation written in clear English, with no spelling or grammar errors?
- If the contribution contains multiple commits, are they grouped together into logical changes (one major change per pull request)? Are any commits with names like “fixed typo” [squashed into previous commits](#)?
- If you’re unsure about any of these points, please open the Pull Request anyhow and then ask us for feedback.

Pull Request Process

After you open the Pull Request, there will probably be some discussion in the comments field of the request itself.

Once the Pull Request is ready to merge, it will first be merged into our internal git system for in-house automated testing.

If this process passes, it will be merged onto the public github repository.

Legal Part

Before a contribution can be accepted, you will need to sign our [contributor-agreement](#). You will be prompted for this automatically as part of the Pull Request process.

Related Documents

esp-dev-kits 编码规范

总体原则

- 简洁明了，结构清晰
- 统一风格，易于维护

- 充分注释，易于理解
- 继承 ESP-IDF 已有规范
- 继承第三方代码已有规范

目录结构

- components: 总体上按照功能分类，大类下如果存在多级目录，应包含一个 README 做综述和索引；
- docs: rst 格式文档，包括各个组件的使用指南，API 说明；
- examples: 总体上按照与组件对应的功能分类；
- tools: CI 脚本、调试工具。

头文件

- 尽量每一 .c 对应一个同名的 .h 文件；
- 单个组件存在多个 .h，主要对外 .h 命名尽量与组件名保持一致；
- 头文件中主要放函数声明，不放函数实现；
- 尽量不在头文件定义任何形式的变量；
- 头文件应按照注释规范，对函数接口进行充分注释；
- 添加宏定义避免重复引用，宏定义名为大写的头文件名加下划线填充：

```
#ifndef _IOT_I2C_BUS_H_
#define _IOT_I2C_BUS_H_

#endif
```

- 函数声明添加 extern “C” 修饰，支持 C/C++ 混合编程：

```
#ifdef __cplusplus
extern "C"
{
#endif

//c code

#ifndef __cplusplus
}
#endif
```

注释

- 安装 VS CODE 插件 Doxygen Documentation Generator 可自动生成注释框架；
- 注释中避免使用单词缩写；
- 函数声明处注释需要描述函数功能、性能或用法，输入和输出参数、函数返回值说明。

自动生成的注释框架：

```
/***
 * @brief
 *
 * @param port
 * @param conf
 * @return i2c_bus_handle_t
 */
i2c_bus_handle_t iot_i2c_bus_create(i2c_port_t port, i2c_config_t* conf);
```

补充信息和参数方向：

```
/***
 * @brief Create an I2C bus instance then return a handle if created successfully.
 * @note Each I2C bus works in a singleton mode, which means for an i2c port only one
 *       ↪group parameter works. When
 *       ↪iot_i2c_bus_create is called more than one time for the same i2c port, following
 *       ↪parameter will override the previous one.
 *
 * @param[in] port I2C port number
 * @param[in] conf Pointer to I2C parameters
 * @return i2c_bus_handle_t Return the I2C bus handle if created successfully, return
 *         ↪NULL if failed.
 */
i2c_bus_handle_t iot_i2c_bus_create(i2c_port_t port, i2c_config_t* conf);
```

- 版权声明注释（第三方代码，请保留版权声明信息）

```
// Copyright 2019-2020 Espressif Systems (Shanghai) PTE LTD
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
//     http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
```

(下页继续)

(续上页)

```
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.
```

函数规范

- 多处重复使用的代码尽量设计为函数；
- 作用域仅限于当前文件的函数必须声明为静态 `static`；
- 设计使用静态全局变量、静态局部变量的函数时，需要考虑重入问题；
- 尽量在一个固定函数中操作静态全局变量；
- 如果函数存在重入或线程安全问题，需在注释中说明；
- 同一组件内的公有函数名，应保持同一前缀；
- 函数名统一使用 `snake_case` 格式，只使用小写字母，单词之间加 `_`；
- 函数命名指引（应保持与已有代码风格一致，不严格约束）：

| 函数名格式 | 函数示例 | 说明 |
|---|--|--|
| <code>iot_type_xxx</code> | <code>iot_sensor_xxx;</code> <code>iot_board_xxx;</code> <code>iot_storage_…</code> | 高度抽象的 <code>iot</code> 组件 |
| <code>type_xxx</code> | <code>imu_xxx;</code> <code>light_xxx;</code> <code>eeprom_xxx</code> | 对一类外设的抽象 |
| <code>name_xxx</code> | <code>mpu6050_xxx;</code> | 底层 <code>driver</code> , 由于可能来自第三方, 不约束函数名 |
| <code>xxx_creat</code> / <code>xxx_delete</code> | | 创建和销毁 |
| <code>xxx_read</code> / <code>xxx_write</code> | | 数据操作 |
| | | |

变量规范

- 避免使用全局变量，可声明为静态全局变量，使用 `get_` `set_` 等接口进行变量操作；
- 作用域仅限于当前文件的变量必须声明为静态变量 `static`；
- 静态全局变量请添加 `g_` 前缀，静态局部变量请添加 `s_` 前缀；
- 局部变量设计大小时，应考虑栈溢出的问题；
- 任何变量定义时，必须赋初值；

- 变量功能要明确，避免将单一变量做多个用途；
- 句柄类型变量，在对象销毁后，应重新赋值为 NULL；
- 变量统一使用 snake_case 格式，只使用小写字母，单词之间加 _；
- 避免不必要的缩写，例如 data 不必缩写为 dat；
- 变量应尽量使用有意义的词语，或者已经达成共识的符号或 词语缩写；
- 变量命名指引：

| 类型 | 规范 | 示例 |
|--------|-----------------------|------------------------------------|
| 全局变量 | 避免使用 | x |
| 静态全局变量 | static 标识，g_ 前缀，赋初值 | static uint32_t g_connect_num = 0; |
| 静态局部变量 | static 标识，s_ 前缀，赋初值 | static uint32_t s_connect_num= 0; |
| 迭代计数变量 | 使用通用的 i j k | |
| 常用缩写 | abbreviations-in-code | addr,buf ,cfg , cmd, , ctrl, |
| | | |

- 常用缩写列表

| 缩写 | 全称 | 缩写 | 全称 | 缩写 | 全称 | 缩写 | 全称 |
|------|---------|------|-------------|-------|-----------|------|-----------------------|
| addr | address | id | identifier | len | length | ptr | pointer |
| buf | buffer | info | information | obj | object | ret | return |
| cfg | command | hdr | header | param | parameter | temp | temporary、temperature |
| cmd | command | init | initialize | pos | position | ts | timestamp |

类型定义

- 使用加 snake_case 格式加 _t 后缀

```
typedef int signed_32_bit_t;
```

- 枚举应通过 typedef 通过以下方式定义

```
typedef enum {
    MODULE_FOO_ONE,
    MODULE_FOO_TWO,
    MODULE_FOO_THREE
} module_foo_t;
```

格式和排版规范

该部分继承 ESP-IDF 规范。

1. 缩进

每个缩进层使用 **4个空格**，不要使用制表符进行缩进，将编辑器配置为每次按 tab 键时发出 4 个空格。

2. 垂直间隔

在函数之间放置一个空行，不要以空行开始或结束函数。

```
void function1()
{
    do_one_thing();
    do_another_thing();
                                // INCORRECT, don't place empty line here
}

                                // place empty line here

void function2()
{
                                // INCORRECT, don't use an empty line here
    int var = 0;
    while (var < SOME_CONSTANT) {
        do_stuff(&var);
    }
}
```

只要不严重影响可读性，最大行长度为 120 个字符。

3. 水平间隔

总是在条件和循环关键字之后添加单个空格。

```
if (condition) {      // correct
    // ...
}

switch (n) {          // correct
    case 0:
        // ...
}
```

(下页继续)

(续上页)

```
for(int i = 0; i < CONST; ++i) {      // INCORRECT
    // ...
}
```

在二元操作符两端添加单个空格，一元运算符不需要空格，可以在乘法运算符和除法运算符之间省略空格。

```
const int y = y0 + (x - x0) * (y1 - y0) / (x1 - x0);      // correct

const int y = y0 + (x - x0)*(y1 - y0)/(x1 - x0);        // also okay

int y_cur = -y;                                         // correct
++y_cur;

const int y = y0+(x-x0)*(y1-y0)/(x1-x0);                // INCORRECT
```

. 和 -> 操作符的周围不需要任何空格。

有时，在一行中添加水平间隔有助于提高代码的可读性。如下，可以添加空格来对齐函数参数：

```
gpio_matrix_in(PIN_CAM_D6,    I2S0I_DATA_IN14_IDX,  false);
gpio_matrix_in(PIN_CAM_D7,    I2S0I_DATA_IN15_IDX,  false);
gpio_matrix_in(PIN_CAM_HREF,  I2S0I_H_ENABLE_IDX,   false);
gpio_matrix_in(PIN_CAM_PCLK,  I2S0I_DATA_IN15_IDX,  false);
```

- 但是请注意，如果有人添加了一个新行，第一个参数是一个更长的标识符（例如 PIN_CAM_VSYNC），它将不适合。因为必须重新对齐其他行，这添加了无意义的更改。因此，尽量少使用这种对齐，特别是如果您希望稍后将新行添加到这列中。
- 不要使用制表符进行水平对齐，不要在行尾添加尾随空格。

4. 括号

函数定义的大括号应该在单独的行上。

```
// This is correct:
void function(int arg)
{
}

// NOT like this:
void function(int arg) {
```

(下页继续)

(续上页)

}

在函数中，将左大括号与条件语句和循环语句放在同一行。

```
if (condition) {
    do_one();
} else if (other_condition) {
    do_two();
}
```

5. 注释

// 用于单行注释。对于多行注释，可以在每行上使用 // 或 / * * / 块注释。

虽然与格式没有直接关系，但下面是一些关于有效使用注释的注意事项。

- 不要使用一个注释来禁用某些功能。

```
void init_something()
{
    setup_dma();
    // load_resources();           // WHY is this thing commented, asks the_
    ↵reader?
    start_timer();
}
```

- 如果不再需要某些代码，则将其完全删除。如果你需要，你可以随时在 git 历史中查找这个文件。如果您因为临时原因而禁用某些调用，并打算在将来恢复它，则在相邻行上添加解释。

```
void init_something()
{
    setup_dma();
    // TODO: we should load resources here, but loader is not fully integrated yet.
    // load_resources();
    start_timer();
}
```

- #if 0 ... #endif 块也是如此。如果不使用，请完全删除代码块。否则，添加注释以解释为什么禁用该块。不要使用 #if 0 ... #endif 或注释来存储将来可能需要的代码段。
- 不要添加有关作者和更改日期的琐碎注释。您总是可以查找谁使用 git 修改了任何给定的行。例如，此注释在不添加任何有用信息的情况下，使代码混乱不堪：

```

void init_something()
{
    setup_dma();
    // XXX add 2016-09-01
    init_dma_list();
    fill_dma_item(0);
    // end XXX add
    start_timer();
}

```

6. 代码行的结束

commit 中只能包含以 LF (Unix 风格) 结尾的文件。

Windows 用户可以将 git 配置为在本地 checkout 是 CRLF (Windows 风格) 结尾，通过设置 core.autocrlf 设置来 commit 时以 LF 结尾。Github 有一个关于设置此选项的文档。但是，由于 MSYS2 使用 Unix 样式的行尾，因此在编辑 ESP-IDF 源文件时，通常更容易将文本编辑器配置为使用 LF (Unix 样式) 结尾。

如果您在分支中意外地 commit 了 LF 结尾，则可以通过在 MSYS2 或 Unix 终端中运行此命令将它们转换为 Unix (将目录更改为 IDF 工作目录，并预先检查当前是否已 checkout 正确的分支)：

```

git rebase --exec 'git diff-tree --no-commit-id --name-only -r HEAD | xargs dos2unix &
→& git commit -a --amend --no-edit --allow-empty' master

```

请注意，这行代码将在 master 上重新建立基，并在最后更改分支名称以在另一个分支上建立基。

要更新单个提交，可以运行

```
dos2unix FILENAME
```

然后运行

```
git commit --amend
```

7. 格式化代码

您可以使用 astyle 程序根据上述建议对代码进行格式化。

如果您正在从头开始编写一个文件，或者正在进行完全重写，请随意重新格式化整个文件。如果您正在更改文件的一小部分，不要重新格式化您没有更改的代码。这将帮助其他人检查您的更改。

要重新格式化文件，请运行

```
tools/format.sh components/my_component/file.c
```

CMake 代码风格

- 缩进是 4 个空格
- 最大行长为 120 个字符。分割行时，请尝试尽可能集中于可读性（例如，通过在单独的行上配对关键字/参数对）。
- 不要在 endforeach()、endif() 等后面的可选括号中放入任何内容。
- 对命令、函数和宏名使用小写 (with_underscores)。
- 对于局部作用域的变量，使用小写字母 (with_underscores)。
- 对于全局作用域的变量，使用大写 (WITH_UNDERSCORES)。
- 其他，请遵循 cmake-lint 项目的默认设置。

Bibliography

[A] flash 封装在芯片内部。