

# 触屏版二次开发文档

## 目录

一、硬件设计 .....	3
(一) 基本参数 .....	3
(二) 电路原理图 .....	3
(三) GPIO 分配 .....	4
(四) 基本 IO.....	4
1. 电压检测及充电信号 .....	4
2. 按键及上电信号 .....	5
3. 蜂鸣器 .....	5
(五) LCD 模块 .....	6
(六) SD 卡模块.....	7
二、软件开发指南 .....	1
1. Arduino（推荐用于快速原型） .....	1
2. ESP-IDF（推荐用于产品级/高性能项目） .....	2
3. PlatformIO（推荐作为 Arduino/IDF 的 IDE 替代） .....	2

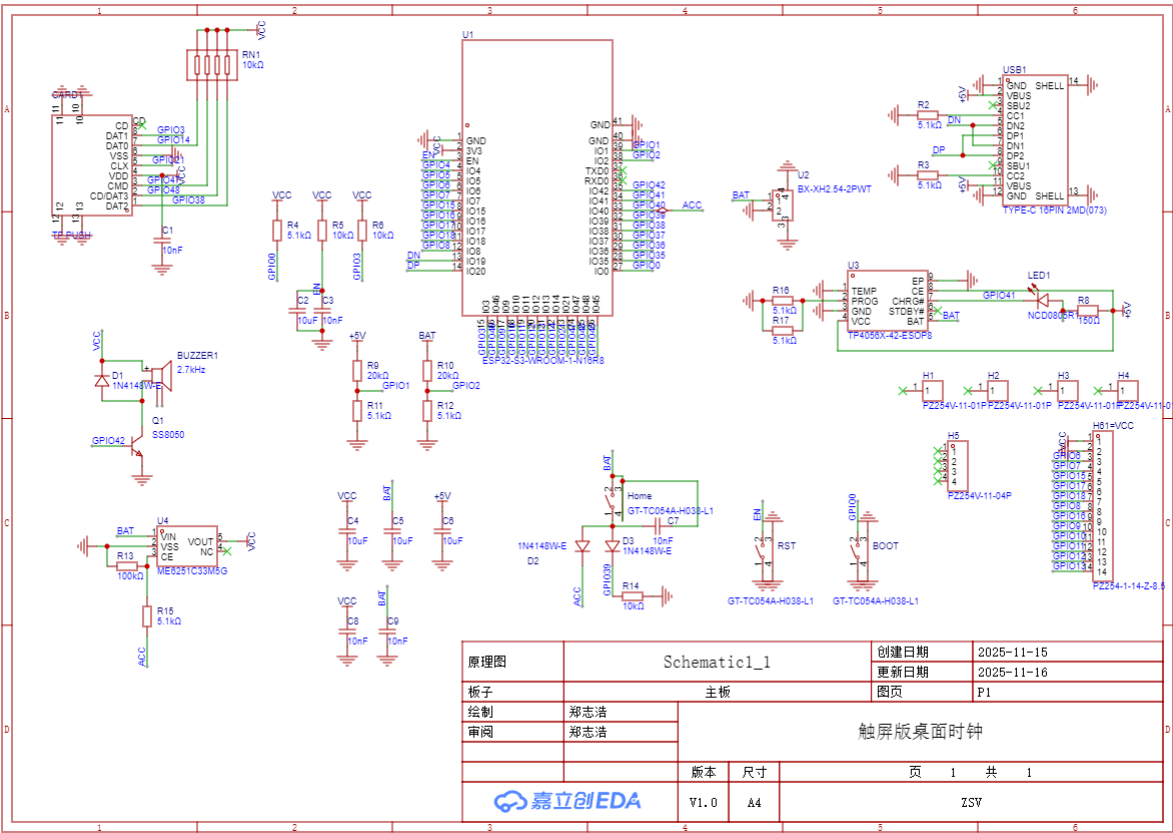
4.ESPHome（适合 IoT 自动化，非 UI 场景） ...	3
5.LuatOS (上手简单、拓展困难).....	4
6.图形化方案 .....	5

一、硬件设计

(一) 基本参数

- 主芯片：ESP32S3 N16R8 (WROOM)
- 电源管理：ME6251C33M5G 3.3V 线性稳压器
- 充电芯片：TP4056X [电池已含充放电保护板]
- 充电电流：300mA 电池容量：1500mAh
- 显示屏：ST7789v 2.4 寸 240H\*320V
- 触摸驱动芯片：FT6336U
- 外部功能：蜂鸣器、SD 卡、JTAG

(二) 电路原理图



### (三) GPIO 分配

LCD	ESP32S3	
CS	6	SPI2
RESET	7	
D/C	15	
MOSI	17	
SCK	18	
BL	8	
MISO	16	

TOUCH_SPI	TOUCH_I2C	S3
CLK	SCL	9
CS	RST	10
SDI	SDA	11
SDO	NC	12
IRQ	INT	13

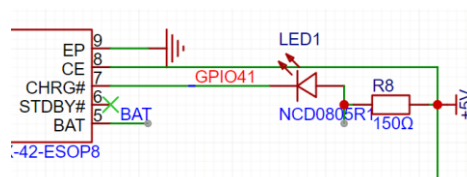
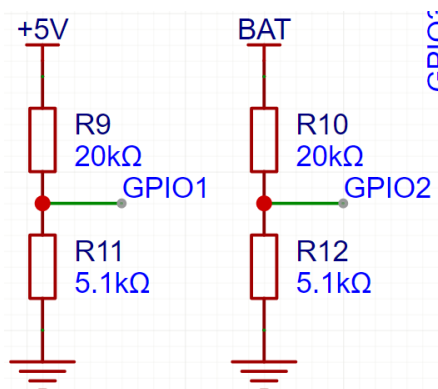
USBIN	1	USB电压检测
BAT	2	电池电压检测
BOOT	0	左上方按键
RST	EN	左下方按键
HOME	39	右上方按键
ACC	40	上电信号
BEEP	42	蜂鸣器
CHG	41	充电信号

SDCARD	S3
D1	3
D0	14
CLK	21
CMD	47
D3	48
D2	38

### (四) 基本 IO

#### 1. 电压检测及充电信号

USB 及电池电压检测电路如左图所示，采用 1/5 分压可使用 ADC 采集对应引脚电压检测状态。



充电信号如右图所示，低电平代表进入充电状态。

## 2. 按键及上电信号

板载三个按键，除 **RST** 外均为高电平有效（即按下后是高电平）。

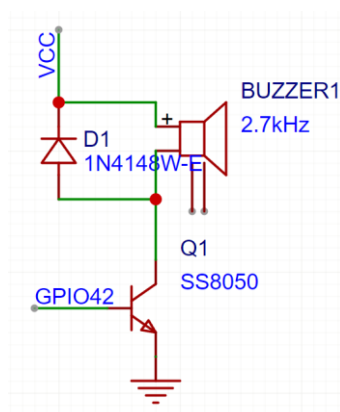
其中 **RST** 为复位按键，位于左下方，不可复用。

**BOOT** 键在开机完成后可复用为普通按键，对应引脚号为 **GPIO0**。

**HOME** 键同时作为开机键使用，按下同时触发上电信号，对应输入引脚为 **GPIO39**。

开机后需要将 **ACC** 信号（**GPIO40**）设置为高电平输出，否则松开按键后将会掉电。

## 3. 蜂鸣器



蜂鸣器对应引脚为 **GPIO42**。

不使用时需要输入低电平，否则将会短路。

只能输入频率 **100hz**，占空比 **50%**以下的方波。

## （五）LCD 模块

屏幕详情及购买链接：

<https://item.taobao.com/item.htm?id=748230294619>

所有 SPI 及 I2C 信号线已做等长处理，最大通信频率 80Mhz。

主驱动芯片为 ST7789v

横屏时 高度 240 宽度 320 像素

RGB565 色域 IPS 面板

电容触摸面板，触摸驱动芯片为 FT6336U,采用 I2C 通信

触摸芯片	GPIO	备注
SCL	9	IIC 时钟
RST	10	复位
SDA	11	IIC 数据
INT	13	中断，触摸时，输入低电平到主控
LCD 屏幕	GPIO	
CS	6	液晶屏片选控制信号，低电平有效
RESET	7	液晶屏复位控制信号，低电平复位
D/C	15	液晶屏命令/数据选择控制信号
MOSI	17	SPI 总线写数据信号
SCK	18	SPI 总线时钟信号
BL	8	液晶屏背光控制信号，默认开启
MISO	16	SPI 总线读数据信号

## (六) SD 卡模块

连接的引脚如下：

SD 卡	GPIO
D1	3
D0	14
CLK	21
CMD	47
D3	48
D2	38

该模块的可用性尚未验证。





## 二、软件开发指南

各开发方式建议

### 1. Arduino（推荐用于快速原型）

优点：

- 库丰富（TFT\_eSPI、FT6336U、SD 等支持良好）
- 开发门槛低，适合学生项目、毕业设计
- 支持 LVGL（通过 lvgl\_arduino 示例）

缺点：

- 实时性与内存控制不如原生 IDF
- ESP32-S3 部分高级功能（如 USB OTG）支持有限

适用场景：

快速验证屏幕/触摸/SD 功能

开发简单 UI 菜单、传感器交互等

关键库：

TFT\_eSPI（配置 ST7789）

FT6336U（或 XPT2046 替代思路，但需自改）

SD / SD\_MMC（注意引脚限制）

官方文档：

<https://docs.arduino.cc/>

ESP32 Arduino Core：

<https://github.com/espressif/arduino-esp32>

## 2. ESP-IDF（推荐用于产品级/高性能项目）

优点：

- 官方原生支持 ESP32-S3
- 完整控制外设（SPI/I2C/ADC/PWM 等）
- LVGL 官方提供 ESP-IDF 集成方案
- 支持 FreeRTOS、低功耗、安全启动等

缺点：

- 学习曲线陡峭
- 项目配置复杂（CMake、Kconfig）

适用场景：

需要稳定、长期运行的嵌入式系统

关键组件：

lvgl + lvgl\_esp32\_drivers

driver/spi\_master.h + driver/i2c.h

fatfs for SD card

官方文档：

[https://docs.espressif.com/projects/esp-idf/zh\\_CN/latest/esp32s3/](https://docs.espressif.com/projects/esp-idf/zh_CN/latest/esp32s3/)

## 3. PlatformIO（推荐作为 Arduino/IDF 的 IDE 替代）

优点：

- 支持 Arduino 框架 + ESP-IDF 框架双模式

- 跨平台、依赖自动管理、调试集成好
- 适合从 Arduino 过渡到 IDF 的开发者

缺点：

- 需要熟悉 platformio.ini 配置
- 某些最新 IDF 功能滞后

建议用法：

初期用 `framework = arduino`

后期切换为 `framework = espidf` 进行性能优化

官方文档：

<https://docs.platformio.org/en/latest/>

ESP32-S3 支持页：

<https://docs.platformio.org/en/latest/platforms/espressif32.html>

#### 4. ESPHome（适合 IoT 自动化，非 UI 场景）

优点：

- 基于 YAML 配置，无需写代码
- 与 Home Assistant 深度集成
- 自动处理 WiFi、OTA、MQTT、传感器上报

缺点：

- LVGL 等功能组件需要自行移植
- FT6336U 触摸芯片无官方驱动，需要自行移植
- SD 卡、蜂鸣器、复杂 GPIO 逻辑需通过 custom C++

组件实现

- 主要面向“传感器+执行器”，而非“人机交互终端”

适用场景：

- ◆ 物联网型项目。
- ◆ 将设备作为数据采集终端（如上报电池电压）
- ◆ 通过 Home Assistant 远程触发声光反馈（如蜂鸣器报警）

ESPHome 官方文档：

<https://esphome.io/>

ESP32-S3 支持说明：

<https://esphome.io/devices/esp32.html#esp32-s3>

## 5.LuatOS (上手简单、拓展困难)

LuatOS 是一款针对嵌入式的脚本运行框架，可以极大提升开发效率。

使用 Lua 5.3 作为主要语言，通过使用 C 语言编写的虚拟机在 ESP32S3 上运行 Lua 代码

优点：

- 上手难度低，调试方便
- 具有丰富的库支持

缺点：

- 没有触摸驱动支持，拓展复杂
- 官方已停止支持，需要自行维护底层

相关文档：

<https://zsvstudio.top/luawiki/index.html>

## 6. 图形化方案

AnyUI 是一个非常值得考虑的图形化开发路线，尤其适合在 ESP32（如 ESP32-S3）上快速开发带 LCD + 触摸屏 + 按键 交互的嵌入式 UI 应用。

### AnyUI 的优势

基于 LVGL，专为嵌入式设计

AnyUI 是一款免费、跨平台的可视化 UI 设计工具，完全基于 LVGL 框架，支持拖拽式组件布局、样式配置、事件绑定，能自动生成可集成的 C 代码 9。

### 与 ESP32 生态良好兼容

社区已有多个项目成功将 AnyUI 生成的代码用于 ESP32/ESP32-S3 + ST7789 + 触摸屏 的组合。它输出的标准 LVGL 代码可无缝集成到 ESP-IDF 或 Arduino + PlatformIO 项目中 4。

### 支持复杂 UI 与事件逻辑

可设计多页面、滑动、按钮、列表、滑块等

支持为控件绑定点击、长按、滑动等事件

生成的代码结构清晰（ui.c / ui.h），便于与您的按键（如 GPIO0/HOME）逻辑对接

持续更新，中文友好

截至 2025 年，AnyUI 已发布 v0.32.0 版本，持续优化组件编辑体验和构建速度，并提供中文文档和社区支持。