# WT9932S31-TINY
# 使用指南

2026-10-21

# 目录

**免责申明** **2**

**注意** **2**

**1. 开发板简介** **4**
1.1 开发板介绍 . . . . . 4
1.2 产品特点 . . . . . 4
1.3 开发板图片 . . . . . 5

**2. 硬件参考** **6**
2.1 功能框图 . . . . . 6
2.2 硬件外设 . . . . . 7
2.3 组件介绍 . . . . . 8
2.4 接口封装 . . . . . 10
2.5 接口说明 . . . . . 10
[H1] 排针接口 . . . . . 10
[H2] 排针接口 . . . . . 12
[J1][J2]USB 供电接口 . . . . . 13
[CARD1]TF 卡接口 . . . . . 13
[FPC2]DVP FPC 座 接口 . . . . . 13
[U5] DVP 排针接口 . . . . . 14

**3. 原理图** **16**

**4. 结构支持** **17**
4.1 PCBA 尺寸图 . . . . . 17

**5. 使用指引** **18**
5.1 准备工作 . . . . . 18
5.2 硬件接线 . . . . . 19
5.3 使用指南 . . . . . 19
5.3.1 从简单的示例开始验证 . . . . . 20

**6. 相关文档** **23**

**7. 联系我们** **24**

wireless-tag 深圳市启明云端科技有限公司

## 免责申明

和版权公告本文中的信息，包括供参考的 URL 地址，如有变更，恕不另行通知。文档 “按现状” 提供，不负任何担保责任，包括对适销性、适用于特定用途或非侵权性的任何担保，和任何提案、规格或样品在他处提到的任何担保。本文档不负任何责任，包括使用本文档内信息产生的侵犯任何专利权行为的责任。本文档在此未以禁止反言或其他方式授予任何知识产权使用许可，不管是明示许可还是暗示许可。文中提到的所有商标名称、商标和注册商标均属其各自所有者的财产，特此声明。

## 注意

由于产品版本升级或其他原因，本手册内容有可能变更。深圳市启明云端科技有限公司保留在没有任何通知或者提示的情况下对本手册的内容进行修改的权利。本手册仅作为使用指导，深圳市启明云端科技有限公司尽全力在本手册中提供准确的信息，但是深圳市启明云端科技有限公司并不确保手册内容完全没有错误，本手册中的所有陈述、信息和建议也不构成任何明示或暗示的担保。

修改记录

| 版本 | 时间 | 变更人 | 变更项 |
| --- | --- | --- | --- |
| V1.0 | 2026/10/08 | Kirto | 创建文档 |
| V1.1 | 2026/10/21 | Iris | 修正文档元数据、资源路径和操作说明；删除封面副标题，更新文档日期，修正开发板图片大小；调整相关文档顺序，新增乐鑫官方快速入门 |

wireless_tag 深圳市启明云端科技有限公司

# 1. 开发板简介

## 1.1 开发板介绍

WT9932S31-TINY 是一款基于 WT0132S31-S1 模组的小尺寸开发板，以乐鑫 ESP32-S31 为核心，采用四层 PCB 设计，尺寸仅为 54 × 28 mm，适合空间受限的嵌入式应用与原型开发。ESP32-S31 搭载主频最高 320 MHz 的双核 RISC-V 处理器，支持 2.4 GHz Wi-Fi 6、低功耗蓝牙、经典蓝牙，以及 Zigbee 和 Thread，为设备控制、图像处理和多协议物联网应用提供支持。开发板板载 DVP 摄像头 FPC 接口，便于连接摄像头开展图像采集与视觉应用开发。GPIO 通过排针引出，支持面包板连接，方便扩展传感器及其他外设；同时配备 RGB 指示灯，以及 FUSB、HUSB 两个 USB Type-C 接口，适用于物联网终端、图像采集及视觉原型开发。

## 1.2 产品特点

1. **高性能、小尺寸**：基于 WT0132S31-S1 模组，搭载乐鑫 ESP32-S31 双核 RISC-V 处理器，主频最高 320 MHz，支持 Wi-Fi 6 等多种无线通信协议。采用四层 PCB 设计，尺寸仅 **54 × 28 mm**。
2. **DVP 与 GPIO 灵活扩展**：板载 DVP 摄像头 FPC 接口，与乐鑫官方 S31 开发板对应接口兼容。接口下方预留 **2×9、2.54 mm 间距双排贴片排针焊盘，默认不焊接**；不使用 DVP 时，可自行焊接排针，将相关信号引脚作为普通 GPIO 使用。板上引出排针支持面包板连接，方便原型搭建。
3. **丰富的板载外设**：配备支持 **4 线模式的插拔式 SD 卡槽**，方便数据记录与文件存储；板载 **BOOT、EN 双按钮**，便于下载与复位操作；配备 **RGB 指示灯**，支持多彩状态指示与交互效果。
4. **完善的供电设计**：FUSB、HUSB 双 USB Type-C 接口采用电源合路与防反灌设计，同时接入时优先使用 FUSB 供电。排针引出 **5 V、3.3 V（DC-DC）**，DVP 扩展排针焊盘另引出 **1.5 V、2.8 V（LDO）**，方便为不同外设供电。
5. **软硬件开源**：硬件设计在立创平台开源，软件代码在 GitHub 开源，方便学习参考、修改与二次开发。
6. **结构固定接口**：板边固定结构便于配合螺钉、压片或定制底座使用，适合进一步制作外壳与完整设备。

## 1.3 开发板图片

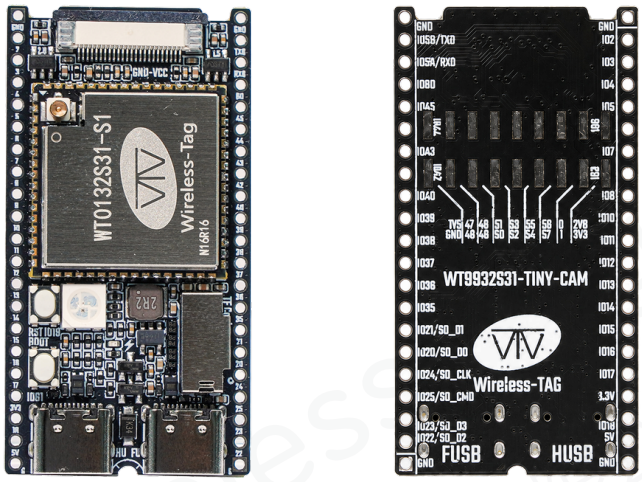



**图 1** – 开发板图片

# 2. 硬件参考

## 2.1 功能框图

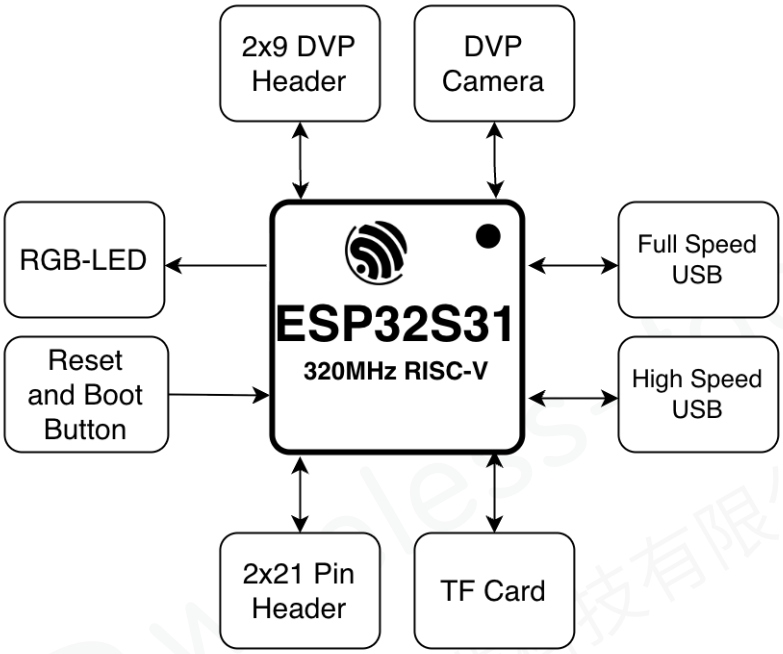



**图 2** – 功能框图

## 2.2 硬件外设

| 外设名称 | 描述 |
| --- | --- |
| WS2812 | 单总线 RGB LED 灯连接在 IO19。 |
| RESET 按键 | 用于复位 |
| BOOT 按键 | 用于手动进入烧录模式，连接在 IO61。 |
| FUSB | 全速 USB，用于供电与下载升级。 |
| HUSB | 高速 USB。 |
| DVP | 0.5 间距，24pin FPC 排线座。 |
| TF 卡 | 四线拔插式 TF 卡卡座。 |

wireless_tag 深圳市启明云端科技有限公司

## 2.3 组件介绍

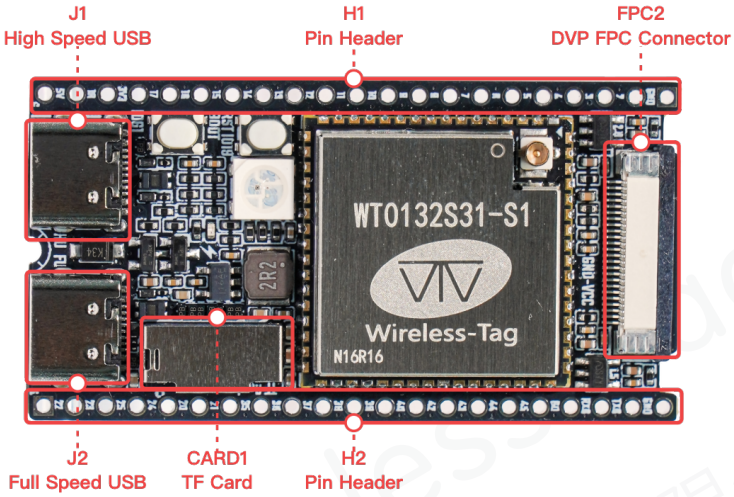



**图 3** – 正面接口组件

| 编号 | 组件名称 | 说明 |
| --- | --- | --- |
| J1 | USB 2.0 Type-C 接口 | 高速 USB 2.0 typeC 接口。无法对外提供电压。 |
| H1 | 排针接口 | 所有可用 GPIO 管脚均已引出至排针 |
| FPC2 | DVP FPC 座子 | 按照 OV3660 线序制作的 DVP FPC 座子 |
| J2 | USB 2.0 Type-C 接口 | 全速 USB 2.0 typeC 接口。可烧录固件，也可用于 JTAG 调试。 |
| CARD1 | TF Card | 4-bit 模式的拔插式 TF 卡 |
| H2 | 排针接口 | 所有可用 GPIO 管脚均已引出至排针 |

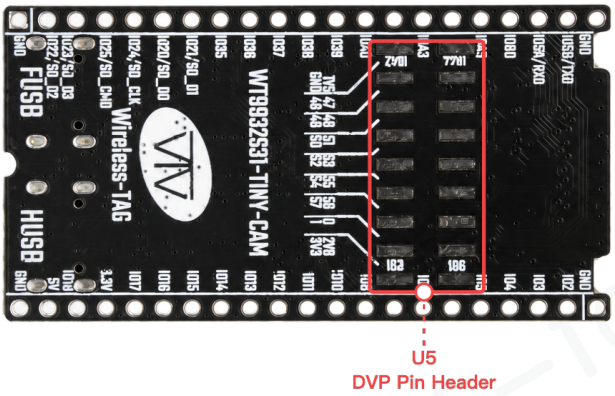



**图 4** – 反面接口组件

| 编号 | 组件名称 | 说明 |
| --- | --- | --- |
| U5 | 2.54mm 间距 2x9 卧贴片排针（未焊接） | 将 DVP 引脚和电源引出，不使用 DVP 情况下，可以使用排针作为普通 IO |

## 2.4 接口封装

| 接口描述 | 接口封装 | 备注 |
| --- | --- | --- |
| 排针接口 | HDR-TH_21P-P2.54-V-M | 间距支持洞洞板/面包板 |
| DVP FPC 座子接口 | FPC-SMD_AFC07-S24FCA-00 | 请注意 FPC 线序，谨防 GND 和 VCC 短接 |
| TF 卡接口 | TF-SMD_TF-021B-H265 | 正面拔插式 TF 卡座 |
| DVP 排针接口 | HDR-SMD_18P-P2.54-H-M-R2-C18-LS17.0 | 出厂未焊接 |

## 2.5 接口说明

### [H1] 排针接口

排针为 2.54mm 21 pin 直插式焊接排针。

具体引脚功能请参考官方手册

引脚线序为：

| 序号 | 引脚 | 描述 | 电压范围 | 备注 |
| --- | --- | --- | --- | --- |
| 1 | GND | 电源地 | 0V | |
| 2 | IO2 | 通用输入输出引脚 | 0V/3.3V | |
| 3 | IO3 | 通用输入输出引脚 | 0V/3.3V | |
| 4 | IO4 | 通用输入输出引脚 | 0V/3.3V | |
| 5 | IO5 | 通用输入输出引脚 | 0V/3.3V | |
| 6 | IO6 | 通用输入输出引脚 | 0V/3.3V | |
| 7 | IO7 | 通用输入输出引脚 | 0V/3.3V | |
| 8 | IO8 | 通用输入输出引脚 | 0V/3.3V | |
| 9 | IO9 | 通用输入输出引脚 | 0V/3.3V | |
| 10 | IO10 | 通用输入输出引脚 | 0V/3.3V | |
| 11 | IO11 | 通用输入输出引脚 | 0V/3.3V | |
| 12 | IO12 | 通用输入输出引脚 | 0V/3.3V | |

| 序号 | 引脚 | 描述 | 电压范围 | 备注 |
| --- | --- | --- | --- | --- |
| 13 | IO13 | 通用输入输出引脚 | 0V/3.3V | |
| 14 | IO14 | 通用输入输出引脚 | 0V/3.3V | |
| 15 | IO15 | 通用输入输出引脚 | 0V/3.3V | |
| 16 | IO16 | 通用输入输出引脚 | 0V/3.3V | |
| 17 | IO17 | 通用输入输出引脚 | 0V/3.3V | |
| 18 | 3.3V | 3.3V 电源输出 | 3.3V | 禁止电源输入 |
| 19 | IO18 | 通用输入输出引脚 | 0V/3.3V | |
| 20 | 5V | 5V 电源输出 | 5V | 禁止电源输入 |
| 21 | GND | 电源地 | 0V | |

wireless-tag 深圳市启明云端科技有限公司

**[H2] 排针接口**

排针为 2.54mm 21 pin 直插式焊接排针。

具体引脚功能请参考官方手册

引脚线序为：

| 序号 | 引脚 | 描述 | 电压范围 | 备注 |
|---|---|---|---|---|
| 1 | GND | 电源地 | 0V | |
| 2 | IO58 | 通用输入输出引脚 | 0V/3.3V | 默认为 UART0 的 TX 引脚 |
| 3 | IO59 | 通用输入输出引脚 | 0V/3.3V | 默认为 UART0 的 RX 引脚 |
| 4 | IO60 | 通用输入输出引脚 | 0V/3.3V | |
| 5 | IO45 | 通用输入输出引脚 | 0V/3.3V | |
| 6 | IO44 | 通用输入输出引脚 | 0V/3.3V | |
| 7 | IO43 | 通用输入输出引脚 | 0V/3.3V | |
| 8 | IO42 | 通用输入输出引脚 | 0V/3.3V | |
| 9 | IO40 | 通用输入输出引脚 | 0V/3.3V | |
| 10 | IO39 | 通用输入输出引脚 | 0V/3.3V | |
| 11 | IO38 | 通用输入输出引脚 | 0V/3.3V | |
| 12 | IO37 | 通用输入输出引脚 | 0V/3.3V | |
| 13 | IO36 | 通用输入输出引脚 | 0V/3.3V | |
| 14 | IO35 | 通用输入输出引脚 | 0V/3.3V | |
| 15 | IO21 | 通用输入输出引脚 | 0V/3.3V | SD 卡 D1 引脚 |
| 16 | IO20 | 通用输入输出引脚 | 0V/3.3V | SD 卡 D0 引脚 |
| 17 | IO24 | 通用输入输出引脚 | 0V/3.3V | SD 卡 CLK 引脚 |
| 18 | IO25 | 通用输入输出引脚 | 0V/3.3V | SD 卡 CMD 引脚 |
| 19 | IO23 | 通用输入输出引脚 | 0V/3.3V | SD 卡 D3 引脚 |
| 20 | IO22 | 通用输入输出引脚 | 0V/3.3V | SD 卡 D2 引脚 |
| 21 | GND | 电源地 | 0V/3.3V | |

**[J1][J2]USB 供电接口**

USB2.0 Type-C 接口，可用于下载/供电。其中 J2 是全速 USB2.0 最高理论速度为 **12Mbps**，J1 是高速 USB2.0 最高理论速度为 **480Mbps**。

**[CARD1]TF 卡接口**

拔插式 TF 卡接口：

| 序号 | 引脚 | 描述 | 电压范围 | 备注 |
|---|---|---|---|---|
| 1 | IO22 | 四线 SDIO 中 DATA2 | 0V/3.3V | |
| 2 | IO23 | 四线 SDIO 中 DATA3 | 0V/3.3V | |
| 3 | IO25 | 四线 SDIO 中 CMD | 0V/3.3V | |
| 4 | VCC | SD 卡供电电源 | 3.3V | |
| 5 | IO24 | 四线 SDIO 中 CLK | 0V/3.3V | |
| 6 | GND | 电源地 | 0V | |
| 7 | IO20 | 四线 SDIO 中 DATA0 | 0V/3.3V | |
| 8 | IO21 | 四线 SDIO 中 DATA1 | 0V/3.3V | |

**[FPC2]DVP FPC 座 接口**

抽屉式 下接 24Pin 0.5mm 间距 以下是引脚定义：

| 序号 | 引脚 | 描述 | 电压范围 | 备注 |
|---|---|---|---|---|
| 1 | NC | 未连接引脚，保持悬空 | / | |
| 2 | 3.3V | 3.3V 电源输出 | 3.3V | |
| 3 | Y4 | 摄像头并行图像数据 Y4 | 0V/3.3V | 主控 IO48，输入 |
| 4 | Y3 | 摄像头并行图像数据 Y3 | 0V/3.3V | 主控 IO47，输入 |
| 5 | Y5 | 摄像头并行图像数据 Y5 | 0V/3.3V | 主控 IO49，输入 |
| 6 | Y2 | 摄像头并行图像数据 Y2 | 0V/3.3V | 主控 IO46，输入 |
| 7 | Y6 | 摄像头并行图像数据 Y6 | 0V/3.3V | 主控 IO50，输入 |
| 8 | PCLK | 摄像头像素时钟，用于采样图像数据 | 0V/3.3V | 主控 IO54，输入 |

| 序号 | 引脚 | 描述 | 电压范围 | 备注 |
|---|---|---|---|---|
| 9 | Y7 | 摄像头并行图像数据 Y7 | 0V/3.3V | 主控 IO51，输入 |
| 10 | GND | 电源及信号参考地 | 0V | |
| 11 | Y8 | 摄像头并行图像数据 Y8 | 0V/3.3V | 主控 IO52，输入 |
| 12 | XMCLK | 摄像头外部主时钟输入 | 0V/3.3V | 主控 IO55，输出 |
| 13 | Y9 | 摄像头并行图像数据 Y9 | 0V/3.3V | 主控 IO53，输入 |
| 14 | 2.8V | 2.8V 摄像头电源引脚 | 2.8V | |
| 15 | 1.5V | 摄像头数字内核电源引脚 | 1.5V | |
| 16 | HREF | 摄像头行有效信号，指示当前行数据有效 | 0V/3.3V | 主控 IO57，输入 |
| 17 | PWDN | 摄像头待机控制输入，高电平有效 | 0V/3.3V | 默认下拉 |
| 18 | VSYNC | 摄像头帧同步信号，用于识别图像帧边界 | 0V/3.3V | 主控 IO56，输入 |
| 19 | RESET | 摄像头硬件复位输入，低电平有效 | 0V/3.3V | 默认上啦 |
| 20 | I2C_SCL | 摄像头 SCCB 配置总线时钟 | 0V/3.3V | 主控 IO1，输出； |
| 21 | GND | 电源及信号参考地 | 0V | |
| 22 | I2C_SDA | 摄像头 SCCB 配置总线数据 | 0V/3.3V | 主控 IO0，双向； |
| 23 | GND | 电源及信号参考地 | 0V | |
| 24 | NC | 未连接引脚，保持悬空 | / | |

**[U5] DVP 排针接口**

2.54mm 间距 2x9 卧贴片排针（未焊接）线序如下：

| 序号 | 引脚 | 描述 | 电压范围 | 备注 |
|---|---|---|---|---|
| 1 | 2.8V | 2.8V 电源引脚 | 2.8V | |
| 2 | 3.3V | 3.3V 电源引脚 | 3.3V | |
| 3 | IO0 | 通用输入输出引脚 | 0V/3.3V | 摄像头 SCCB 数据，CAM_I2C_SDA |
| 4 | IO1 | 通用输入输出引脚 | 0V/3.3V | 摄像头 SCCB 时钟，CAM_I2C_SCL |

| 序号 | 引脚 | 描述 | 电压范围 | 备注 |
| --- | --- | --- | --- | --- |
| 5 | IO56 | 通用输入输出引脚 | 0V/3.3V | 摄像头帧同步信号，DVP_VSYNC |
| 6 | IO57 | 通用输入输出引脚 | 0V/3.3V | 摄像头行有效信号，DVP_HREF |
| 7 | IO55 | 通用输入输出引脚 | 待确认 | 摄像头外部主时钟输入，DVP_XMCLK |
| 8 | IO54 | 通用输入输出引脚 | 0V/3.3V | 摄像头像素时钟，DVP_PCLK |
| 9 | IO53 | 通用输入输出引脚 | 0V/3.3V | 摄像头并行数据 DVP_Y9 |
| 10 | IO52 | 通用输入输出引脚 | 0V/3.3V | 摄像头并行数据 DVP_Y8 |
| 11 | IO51 | 通用输入输出引脚 | 0V/3.3V | 摄像头并行数据 DVP_Y7 |
| 12 | IO50 | 通用输入输出引脚 | 0V/3.3V | 摄像头并行数据 DVP_Y6 |
| 13 | IO49 | 通用输入输出引脚 | 0V/3.3V | 摄像头并行数据 DVP_Y5 |
| 14 | IO48 | 通用输入输出引脚 | 0V/3.3V | 摄像头并行数据 DVP_Y4 |
| 15 | IO47 | 通用输入输出引脚 | 0V/3.3V | 摄像头并行数据 DVP_Y3 |
| 16 | IO46 | 通用输入输出引脚 | 0V/3.3V | 摄像头并行数据 DVP_Y2 |
| 17 | 1.5V | 1.5V 电源引脚 | 1.5V | |
| 18 | GND | 电源及信号参考地 | 0V | |

## 3. 原理图

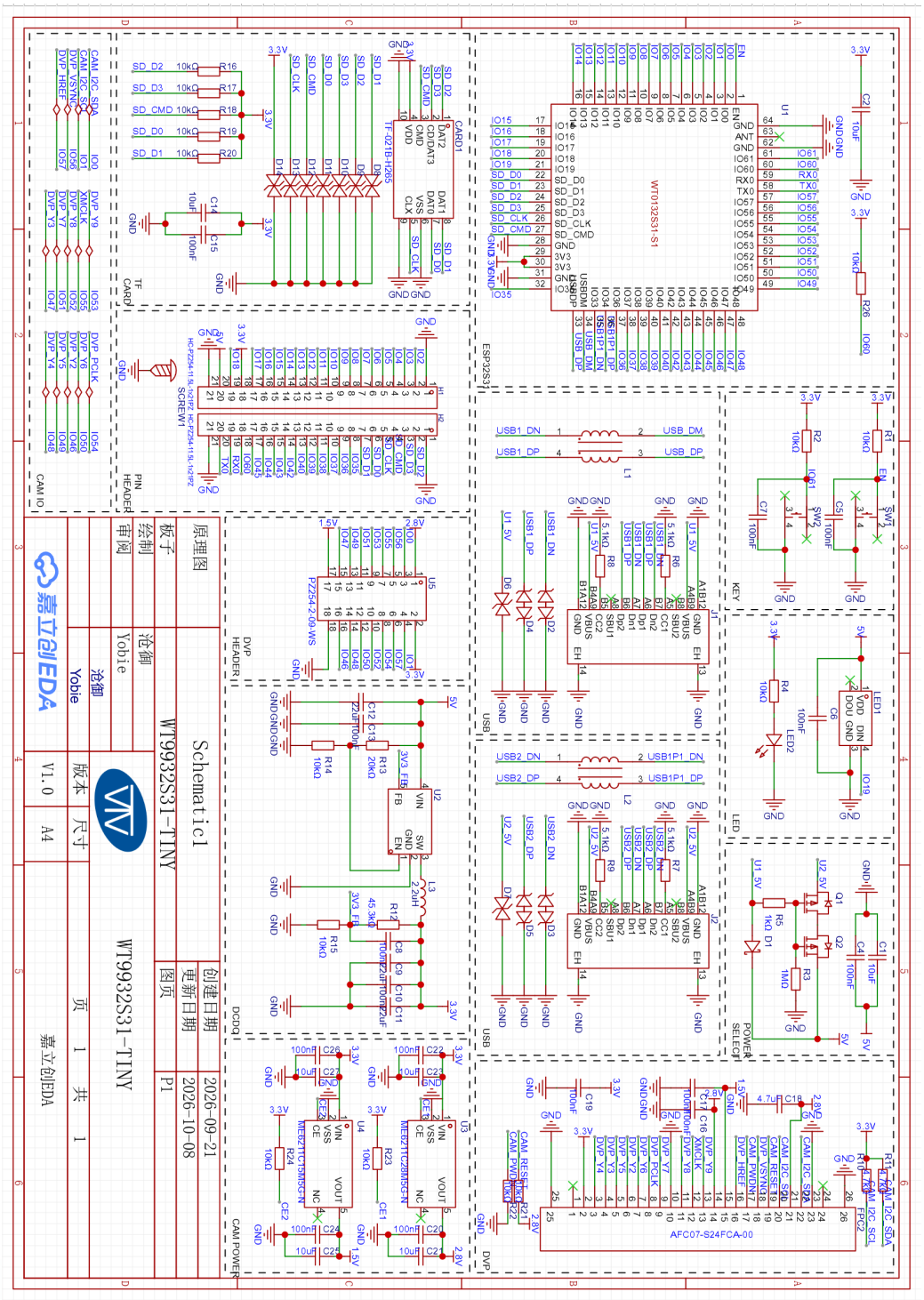

原理图
Schematic1
WT9932S31-TINY
板子
绘制
审阅
沧御
Yobie
版本
V1.0
尺寸
A4
页
1
共
1
图页
P1
创建日期
2026-09-21
更新日期
2026-10-08
嘉立创EDA
WT9932S31-TINY
ESP32S3
CAM IO
TF CARD
PIN HEADER
DVP HEADER
USB
KEY
LED
POWER SELECT
DCDC
CAM POWER
DVP

**图 5** – 原理图

# 4. 结构支持

## 4.1 PCBA 尺寸图

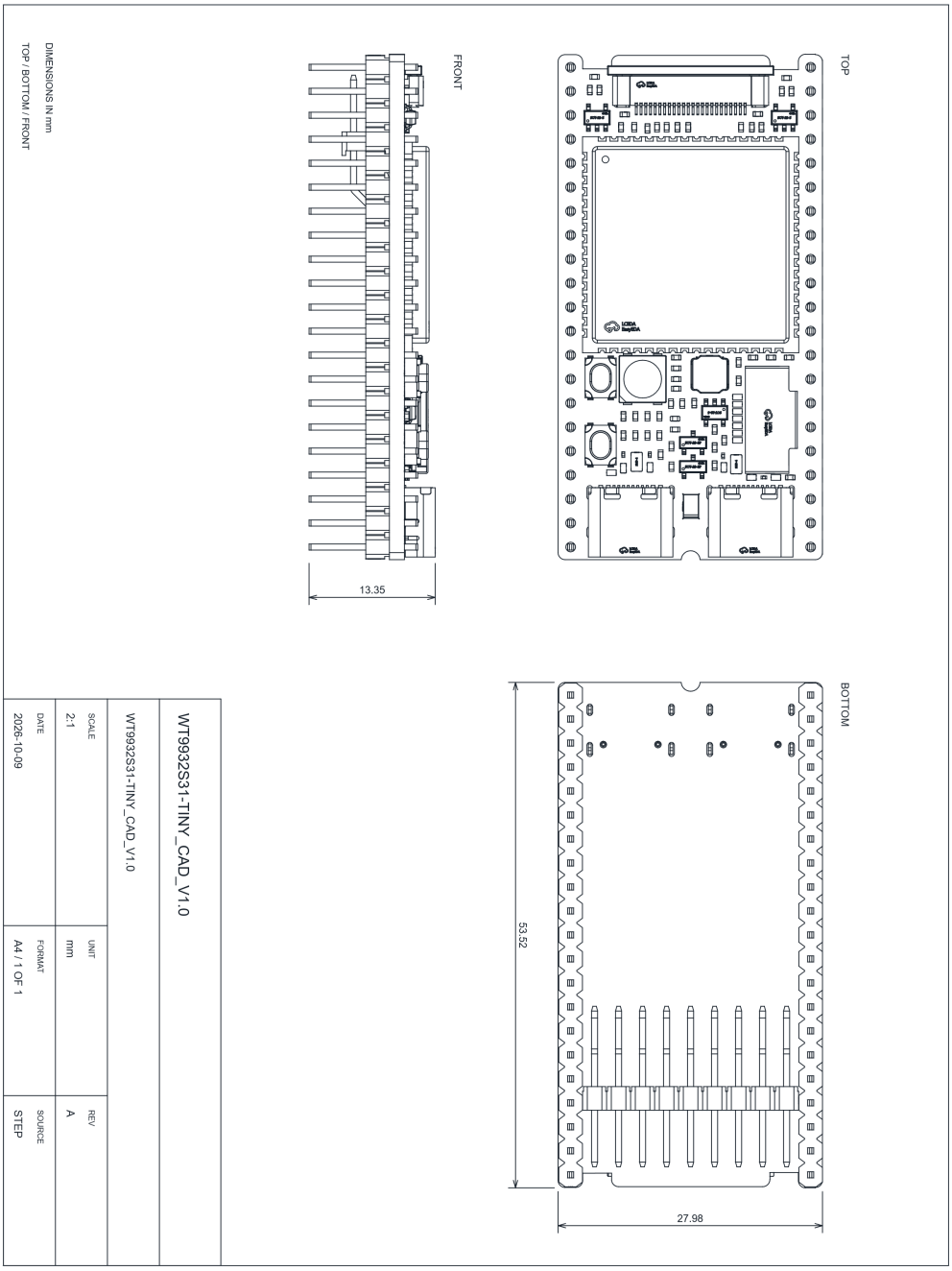



**图 6** – 主板尺寸图

# 5. 使用指引

## 5.1 准备工作

在开始使用硬件之前，请做好以下准备。

硬件准备：

1. **WT9932S31-TINY 开发板** × 1 必需，本产品的核心硬件设备。
2. **USB-C 数据线** × 1 必需，用于为开发板供电或与电脑连接，建议使用原装或质量良好的线材，保证供电稳定。
3. **FPC 天线** × 1 必需，信号增益
4. **Micro SD 内存卡** × 1 可选，用于存储
5. **WTDK-CAM-2145 摄像头** × 1 可选，用于图像的采集




**图 7** – 硬件准备

> **提示**：若使用电脑 USB 供电，建议电脑 USB 接口能提供至少 5V@1A 输出电流；如电源不足，设备可能无法正常启动。

软件准备：

1. 安装 Git。点击了解如何安装
2. 使用 ESP-IDF 安装管理器 (EIM) 来安装 ESP-IDF、构建工具和相关工具链，该管理器支持多种操作系统。点击了解如何安装。
3. 通过 **git clone** 硬件配套的 BSP 仓库：https://github.com/Wireless-TAG-Maker/WT_BSP.git

> 注意：1. 我们的开发套件使用的是 ESP32-S31 芯片，需要安装 V6.0 及其以上的 release 版本。2. 如果你使用的是 windows 系统，这里推荐使用 Windows+WSL 的开发方式，在上面安装 esp-idf 的文档中，按照【在 Linux 上安装 ESP-IDF 及工具链】部分进行阅读安装环境。

### 5.2 硬件接线

按照下图进行接线，将 USB-C 数据线的一端接入开发板，另一端接入电脑，指示灯亮起表示设备已通电。

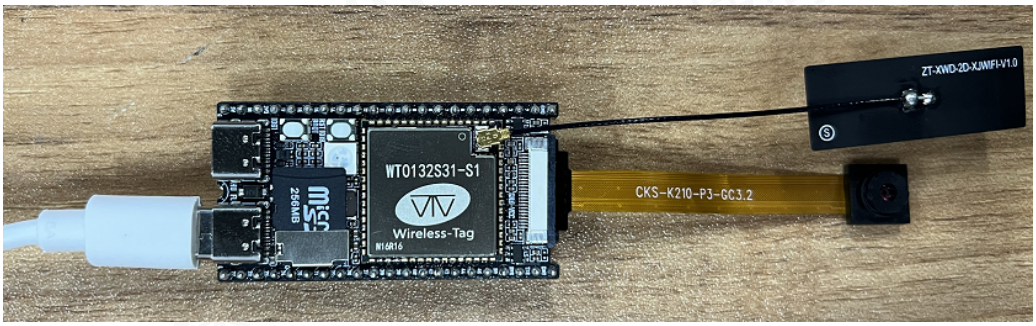



**图 8** – 硬件准备

> 如果接上了 Micro SD 内存卡，其会遮挡到电源指示灯，俯视开发板时将会看不到电源指示灯的状态，需要从侧边观察。

### 5.3 使用指南

开始之前，需要确保上述的【软件准备】部分已经就绪。在这里我们从硬件配套的 BSP 仓库开始。

选择一个用于存放 BSP 项目的目录，然后克隆该仓库：

```
mkdir -p ~/work
cd ~/work
git clone https://github.com/Wireless-TAG-Maker/WT_BSP.git
cd WT_BSP
```

正常情况下应位于 main 分支。后续示例统一假设仓库位于：

```
~/work/WT_BSP
```

> 实际开发时可以使用其他目录，只需把文档命令中的路径替换为自己的真实路径。

我们后续不要直接修改主分支（main 分支），应该创建一个新的分支进行各种修改、实验：

```
git switch -c my_branch
```

后续要切换分支拉取，请这样操作：

```
git switch main
git pull --ff-only origin main
```

> 自定义的应用尽量放在 WT_BSP 仓库外部，或保存在独立的 Git 分支中。这样后续同步官方的更新时，可以通过 main 分支进行，不容易与自己的业务代码产生冲突。团队协作时还应记录验证过的 WT_BSP commit，便于其他开发者复现相同的构建环境。

#### 5.3.1 从简单的示例开始验证

每次更新 WT_BSP 后，建议先编译最小的 blink 示例，确认 ESP-IDF、BSP 和板卡选择工具工作正常：

```
cd ~/work/WT_BSP/examples/get-started/blink

# 注意：
# 1. set-board 命令是我们定制的简化操作的命令，执行该命令之后可以正常使用 build、flash、monitor等命令。
# 2. 通过 set-board 命令会简化很多配置操作，请勿使用 idf.py set-target 命令进行配置，除非你清楚 set-board 命令进行了哪些操作。
# 3. 在这一步，按照提示选择对应的开发套件的数字编号。
~/work/WT_BSP/examples/get-started/blink$ idf.py set-board
...
Supported boards in this example:
0: WT9932C2-TINY (esp32c2)
1: WT9932C3-TINY (esp32c3)
2: WT9932C5-TINY (esp32c5)
3: WT9932C61-TINY (esp32c61)
4: WT9932P4-TINY (esp32p4)
5: WT9932P4C61-TINY (esp32p4)
6: WT9932P4X-TINY (esp32p4)
7: WT9932S31-TINY (esp32s31)

Please select the target board by entering the corresponding number.
Enter board number:  # 根据你的开发套件，输入对应的数字。
7 # 这里我们使用的是 WT9932S31-TINY ，输入 7 后按回车继续
Target changed from esp32p4 to esp32s31; running idf.py fullclean # set-board 命令还会生成当前工程使用的板级预设的配置；切换到目标芯片不同的开发板时，工具会自动执行 fullclean。
Executing action: fullclean
```

```
Executing action: remove_managed_components
Done
Generated sdkconfig.board
Generated sdkconfig.board.Kconfig
Updated build/sdkconfig
Selected WT9932S31-TINY (esp32s31)
Done

# 然后编译
~/work/WT_BSP/examples/get-started/blink$ idf.py build
```

> 后续所有的 BSP 案例都是一样的操作过程。

编译通过之后，可进行烧录，使用 USB-C 数据线，连接上开发板的 FUSB 端口与 PC 的 USB 端口：

```
idf.py -p <COM> flash
```

> **提示**：如果遇到 FUSB 接好后 如果遇到这样的情况：电脑端的 COM 口一直在进行闪烁，在串口工具软件中也会看到 UART 忽有忽无。请这样操作尝试：1. 按下开发板的 BOOT 按钮后，2. 点击 RESET 按钮后再松开 BOOT 按钮。这样手动让硬件进入烧录模式。

成功烧录之后，可以看到开发板中的 RGB 灯的颜色每隔一秒变化一次：

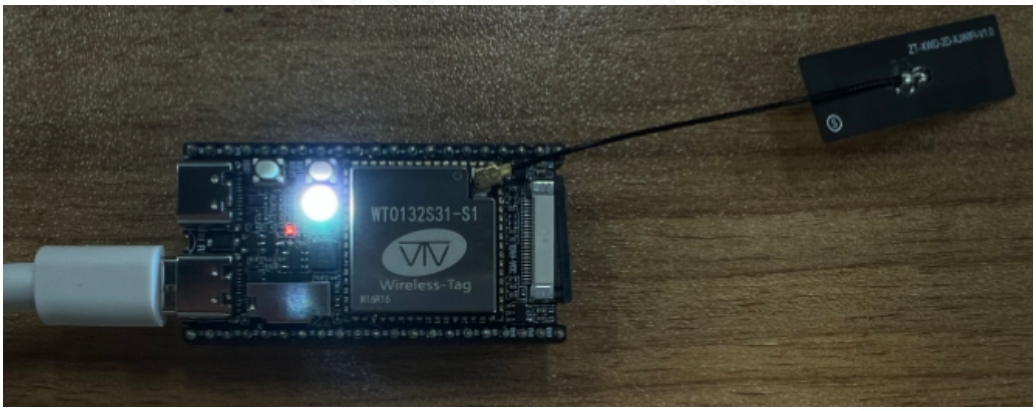



**图 9** – 硬件准备

打开串口会看到如下的 log：

```
I (27) boot: ESP-IDF v6.1 2nd stage bootloader
I (28) boot: compile time Oct 10 2026 10:33:18
I (28) boot: Multicore bootloader
I (30) boot: chip revision: v0.0
I (30) boot: efuse block revision: v0.1
I (34) boot.esp32s31: SPI Speed      : 80MHz
I (38) boot.esp32s31: SPI Mode       : DIO
I (42) boot.esp32s31: SPI Flash Size : 16MB
I (46) boot: Enabling RNG early entropy source...
I (50) boot: Partition Table:
```

```
I (53) boot: ## Label            Usage          Type ST Offset   Length
I (59) boot:  0 nvs              WiFi data        01 02 00009000 00006000
I (66) boot:  1 phy_init         RF data          01 01 0000f000 00001000
I (72) boot:  2 factory          factory app      00 00 00010000 00100000
I (80) boot: End of partition table
I (82) esp_image: segment 0: paddr=00010020 vaddr=40030020 size=0d254h (
   53844) map
I (100) esp_image: segment 1: paddr=0001d27c vaddr=2f000000 size=02d9ch (
   11676) load
I (104) esp_image: segment 2: paddr=00020020 vaddr=40000020 size=21548h
   (136520) map
I (128) esp_image: segment 3: paddr=00041570 vaddr=2f002d9c size=0ad24h (
   44324) load
I (139) esp_image: segment 4: paddr=0004c29c vaddr=2f00db00 size=02cf8h (
   11512) load
I (146) boot: Loaded app from partition at offset 0x10000
I (147) boot: Disabling RNG early entropy source...
I (156) cpu_start: Multicore app
I (165) cpu_start: GPIO 59 and 58 are used as console UART I/O pins
I (166) cpu_start: Pro cpu start user code
I (166) cpu_start: cpu freq: 320000000 Hz
I (168) app_init: Application information:
I (171) app_init: Project name:     blink
I (175) app_init: App version:      73b219b
I (179) app_init: Compile time:     Oct 10 2026 10:33:12
I (184) app_init: ELF file SHA256:  895ce1f04...
I (188) app_init: ESP-IDF:          v6.1
I (192) efuse_init: Min chip rev:     v0.0
I (196) efuse_init: Max chip rev:     v0.99
I (200) efuse_init: Chip rev:         v0.0
I (204) heap_init: Initializing. RAM available for dynamic allocation:
I (210) heap_init: At 2F012270 len 00068D40 (419 KiB): RAM
I (215) heap_init: At 2F07AFB0 len 000041C0 (16 KiB): RAM
I (220) heap_init: At 2E000000 len 00007FE8 (31 KiB): RTCRAM
I (227) spi_flash: detected chip: generic
I (229) spi_flash: flash io: dio
I (233) sleep_gpio: Configure to isolate all GPIO pins in sleep state
I (239) sleep_gpio: Enable automatic switching of GPIO sleep configuration
I (254) main_task: Started on CPU0
I (254) main_task: Calling app_main()
I (264) blink: Initializing Wireless-Tag BSP
I (264) button: IoT Button Version: 4.2.1
I (264) wt_bsp_sdmmc: Initializing on-chip LDO channel 1 for SD card
I (274) wt_bsp_sdmmc: On-chip LDO initialized successfully
```

## 6. 相关文档

| 文档名称 | 链接 | 说明 |
| --- | --- | --- |
| Wireless-tag 官方 WIKI | 官方文档中心 | 开发板文档与资料入口 |
| ESP32-S31 ESP-IDF 编程指南 | 官方中文文档 | ESP32-S31 开发环境搭建、工程开发、驱动 API 及系统功能说明 |
| 乐鑫官方快速入门（ESP32-S31） | 官方中文文档 | ESP32-S31 环境安装、首个工程的配置、编译、烧录与串口监视 |
| ESP32-S31 系列芯片技术规格书 | 官方中文 PDF | 芯片特性、引脚定义、复用功能、供电要求、电气特性及封装尺寸 |
| ESP32-S31 技术参考手册 | 官方中文 PDF | 存储器、GPIO 矩阵、LCD/Camera 等外设的工作原理与寄存器说明 |
| ESP32-S31 硬件设计指南 | 官方中文文档 | 供电、复位、晶振、射频、存储器及外设接口设计要求 |
| ESP32-S31 PCB 版图布局指南 | 官方中文文档 | 电源、晶振、射频、Flash、PSRAM、USB 等布局布线要求 |
| ESP32-S31 系列芯片勘误表 | 官方中文文档 | 芯片各版本的已知问题、影响范围及解决方法 |

# 7. 联系我们

- 官方网址：https://www.wireless-tag.com/
- 淘宝链接：https://shop538225597.taobao.com/
- 销售邮箱：sales@wireless-tag.com
- 技术支持邮箱：technical@wireless-tag.com
- 联系电话：18122057087
- 公众号二维码



**图 10** – 公众号二维码