

# ESP32 技术规格书



**Espressif Systems**

2017 年 6 月 20 日

# 关于本手册

本文档为用户提供 ESP32 硬件技术规格简介。

文档结构如下：

章	标题	内容
第 1 章	概述	概括描述 ESP32，包括专用解决方案、基本特性和高级特性、应用和开发支持。
第 2 章	管脚定义	介绍管脚的布局 and 描述。
第 3 章	功能描述	介绍主要功能模块。
第 4 章	外设接口	描述集成在 ESP32 上的外设接口。
第 5 章	电气特性	提供 ESP32 电气特性和数据。
第 6 章	封装信息	提供 ESP32 的封装细节。
第 7 章	产品型号及订购信息	提供 ESP32 系列的产品型号和订购信息。
第 8 章	学习资源	提供 ESP32 相关必读文档和必备资源。
附录 A	触摸传感器	提供触摸传感器的设计和布局指导。
附录 B	示例代码	提供输入和输出示例代码。
附录 C	ESP32 管脚清单	ESP32 的 GPIO_Matrix, Ethernet_MAC 和 IO_MUX 管脚清单。

## 发布说明

日期	版本	发布说明
2016.08	V1.0	首次发布。
2017.02	V1.1	增加章节 <a href="#">产品型号和订购信息</a> ； 更新章节 <a href="#">MCU 和高级特性</a> ； 更新章节 <a href="#">功能框图</a> ； 更新章节 <a href="#">管脚定义</a> ； 更新章节 <a href="#">CPU 和存储</a> ； 更新章节 <a href="#">音频 PLL 时钟</a> ； 更新章节 <a href="#">极限参数</a> ； 更新章节 <a href="#">封装信息</a> ； 更新章节 <a href="#">学习资源</a> 。
2017.03	V1.2	增加表格 <a href="#">管脚描述</a> 的说明； 更新章节 <a href="#">片上存储</a> 的说明。
2017.04	V1.3	增加附录 <a href="#">ESP32 管脚清单</a> ； 更新表格 <a href="#">Wi-Fi 射频特性</a> ； 更新图 <a href="#">ESP32 管脚布局（封装为 QFN 5*5）</a> 。
2017.05	V1.4	在章节 <a href="#">1.3.2 时钟和定时器</a> 中增加对于外置晶振频率的说明； 增加章节 <a href="#">2.4 Strapping 管脚</a> 中的说明； 将表 <a href="#">8 极限参数</a> 中最大驱动能力由 12 mA 改为 80 mA； 将表 <a href="#">10 Wi-Fi 射频</a> 中输入阻抗值 50Ω 改为输出阻抗值 30+j10 Ω； 更新章节 <a href="#">3.7 RTC 和低功耗管理</a> ； 在表 <a href="#">18 管脚清单</a> 说明第 8 条中增加一条描述； 删除表 <a href="#">IO_MUX</a> 中 GPIO20。

日期	版本	发布说明
2017.06	V1.5	更新章节 <a href="#">1.3.1</a> CPU 和存储中供电电压范围； 更新章节 <a href="#">2.3</a> 电源管理中的说明； 更新表 <a href="#">8</a> 极限参数； 更新表 <a href="#">18</a> 管脚清单说明第 8 条中数字输出管脚的驱动强度； 增加文档变更通知。
2017.06	V1.6	更改两处描述错误：将章节 <a href="#">1.1.2</a> 中的外部元器件个数改为 20；将章节 <a href="#">4.1</a> 中的 GPIO 管脚个数改为 34。

## 文档变更通知

用户可以通过[乐鑫官网](#)订阅技术文档变更的电子邮件通知。

## 免责声明和版权公告

本文中的信息，包括参考的 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。Wi-Fi 联盟成员标志归 Wi-Fi 联盟所有。蓝牙标志是 Bluetooth SIG 的注册商标。

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。

版权归 © 2017 乐鑫所有。保留所有权利。

# 目录

<b>1 概述</b>	<b>1</b>
1.1 专用解决方案	1
1.1.1 超低功耗	1
1.1.2 高集成度	1
1.2 基础协议	1
1.2.1 Wi-Fi	1
1.2.2 蓝牙	2
1.3 MCU 和高级特性	2
1.3.1 CPU 和存储	2
1.3.2 时钟和定时器	3
1.3.3 高级外设接口	3
1.3.4 安全机制	3
1.3.5 开发支持	4
1.4 应用	4
1.5 功能框图	5
<b>2 管脚定义</b>	<b>6</b>
2.1 管脚布局	6
2.2 管脚描述	7
2.3 电源管理	9
2.4 Strapping 管脚	10
<b>3 功能描述</b>	<b>11</b>
3.1 CPU 和存储	11
3.1.1 CPU	11
3.1.2 片上存储	11
3.1.3 外部 Flash 和 SRAM	12
3.1.4 存储器映射	12
3.2 定时器和看门狗	14
3.2.1 64-bit 通用定时器	14
3.2.2 看门狗定时器	14
3.3 系统时钟	15
3.3.1 CPU 时钟	15
3.3.2 RTC 时钟	15
3.3.3 音频 PLL 时钟	15
3.4 射频	15
3.4.1 2.4 GHz 接收器	16
3.4.2 2.4 GHz 发射器	16
3.4.3 时钟生成器	16
3.5 Wi-Fi	16
3.5.1 Wi-Fi 射频和基带	16
3.5.2 Wi-Fi MAC	17
3.5.3 Wi-Fi 固件	17
3.5.4 通信包仲裁 (PTA)	17

3.6	蓝牙	18
3.6.1	蓝牙射频和基带	18
3.6.2	蓝牙接口	18
3.6.3	蓝牙协议栈	18
3.6.4	蓝牙链路控制器	18
3.7	RTC 和低功耗管理	19
<b>4</b>	<b>外设接口和传感器</b>	<b>21</b>
4.1	通用输入/输出接口 (GPIO)	21
4.2	模/数转换器 (ADC)	21
4.3	超低噪声前置模拟放大器	21
4.4	霍尔传感器	21
4.5	数/模转换器 (DAC)	21
4.6	温度传感器	21
4.7	触摸传感器	22
4.8	超低功耗协处理器 (ULP)	22
4.9	以太网 MAC 接口	22
4.10	SD/SDIO/MMC 主机控制器	23
4.11	SDIO/SPI 从机控制器	23
4.12	通用异步收发器 (UART)	23
4.13	I2C 接口	24
4.14	I2S 接口	24
4.15	红外遥控器	24
4.16	脉冲计数器	24
4.17	脉冲宽度调制 (PWM)	24
4.18	LED PWM	24
4.19	串行外设接口 (SPI)	25
4.20	硬件加速器	25
<b>5</b>	<b>电气特性</b>	<b>26</b>
5.1	极限参数	26
5.2	射频功耗参数	26
5.3	Wi-Fi 射频	27
5.4	蓝牙射频	27
5.4.1	接收器 - 基础数据率 (BR)	27
5.4.2	发射器 - 基础数据率 (BR)	28
5.4.3	接收器 - 增强数据率 (EDR)	28
5.4.4	发射器 - 增强数据率 (EDR)	29
5.5	低功耗蓝牙射频	29
5.5.1	接收器	29
5.5.2	发射器	30
<b>6</b>	<b>封装信息</b>	<b>31</b>
<b>7</b>	<b>产品型号和订购信息</b>	<b>32</b>
<b>8</b>	<b>学习资源</b>	<b>33</b>

8.1 必读资料	33
8.2 必备资源	33
<b>附录 A - 触摸传感器</b>	34
A.1. 电极图形	34
A.2. PCB 布局	34
<b>附录 B - 示例代码</b>	36
B.1. 输入	36
B.2. 输出	36
<b>附录 C - ESP32 管脚清单</b>	37
C.1. 管脚清单说明	37
C.2. GPIO_Matrix	39
C.3. Ethernet_MAC	44
C.4. IO_MUX	44

# 表格

2	管脚描述	7
3	Strapping 管脚	10
4	存储器和外设地址映射	13
5	不同省电模式下的功能	20
6	不同省电模式下的功耗	20
7	ESP32 上的电容式传感 GPIO	22
8	极限参数	26
9	射频功耗参数	26
10	Wi-Fi 射频特性	27
11	接收器特性 - 基础数据率 (BR)	27
12	发射器特性 - 基础数据率 (BR)	28
13	接收器特性 - 增强数据率 (EDR)	28
14	发射器特性 - 增强数据率 (EDR)	29
15	低功耗蓝牙接收器特性	29
16	低功耗蓝牙发射器特性	30
17	订购信息	32
18	管脚清单说明	37
19	GPIO_Matrix	39
20	Ethernet_MAC	44

# 插图

1	功能框图	5
2	ESP32 管脚布局 (封装为 QFN 6*6)	6
3	ESP32 管脚布局 (封装为 QFN 5*5)	7
4	地址映射结构	12
5	QFN48 (6x6 mm) 封装	31
6	QFN48 (5x5 mm) 封装	31
7	ESP32 产品型号	32
8	典型的触摸传感器应用	34
9	电极图形要求	34
10	传感器布局布线	35



## 1. 概述

ESP32 是集成 2.4 GHz Wi-Fi 和蓝牙双模的单芯片方案，采用台积电（TSMC）超低功耗的 40 纳米工艺，拥有最佳的功耗性能、射频性能、稳定性、通用性和可靠性，适用于各种应用和不同功耗需求。

目前 ESP32 系列的产品型号包括 ESP32-D0WWDQ6，ESP32-D0WWD，ESP32-D2WD，和 ESP32-S0WWD。产品型号说明和订购信息请参照章节[产品型号和订购信息](#)。

### 1.1 专用解决方案

#### 1.1.1 超低功耗

ESP32 专为移动设备、可穿戴电子产品和物联网 (IoT) 应用而设计。作为业内领先的低功耗芯片，ESP32 具有精细分辨时钟门控、省电模式和动态电压调整等特性。

例如，在低功耗 IoT 传感器 Hub 应用场景中，ESP32 只有在特定条件下才会被周期性地唤醒。低占空比可以令 ESP32 芯片的能耗达到最小。射频功率放大器的输出功率也可调节，以实现通信距离、数据率和功耗之间的最佳平衡。

**说明：**

更多信息请参阅第 3.7 节：RTC 和低功耗管理。

#### 1.1.2 高集成度

ESP32 是业内领先的高度集成的 Wi-Fi+ 蓝牙解决方案，外部元器件只需大约 20 个。ESP32 集成了天线开关、射频 Balun、功率放大器、低噪放大器、过滤器以及电源管理模块，极大减少了印刷电路板 (PCB) 的面积。

ESP32 采用 CMOS 工艺实现单芯片集成射频和基带，还集成了先进的自校准电路，实现了动态自动调整，可以消除外部电路的缺陷，更好地适应外部环境的变化。因此，ESP32 的批量生产不需要昂贵的专用 Wi-Fi 测试设备。

## 1.2 基础协议

### 1.2.1 Wi-Fi

- 802.11 b/g/n/e/i
- 802.11 n (2.4 GHz)，速度高达 150 Mbps
- 802.11 e: QoS 机制实现无线多媒体技术
- WMM-PS, UAPSD
- A-MPDU 和 A-MSDU 帧聚合技术
- 块回复
- 分片和重组
- Beacon 自动监测/扫描
- 802.11 i 安全特性：预认证和 TSN
- 支持 WPA/WPA2/WPA2-Enterprise/WPS 加密

- 基础结构型网络 (Infrastructure BSS) Station 模式/SoftAP 模式
- Wi-Fi Direct (P2P)、P2P 发现、P2P GO 模式和 P2P 电源管理
- UMA 兼容和认证
- 天线分集与选择

### 说明:

更多信息, 请参阅第 3.5 节: Wi-Fi。

## 1.2.2 蓝牙

- 蓝牙 v4.2 完整标准, 包含传统蓝牙 (BR/EDR) 和低功耗蓝牙 (BLE)
- 支持标准 Class-1、Class-2 和 Class-3, 且无需外部功率放大器
- 加强的精准功率控制
- 输出功率高达 +10 dBm
- NZIF 接收器具有 -98 dBm 的 BLE 接收灵敏度
- 自适应跳频 (AFH)
- 基于 SDIO/SPI/UART 接口的标准 HCI
- 速度高达 4 Mbps 的高速 UART HCI
- 支持 BT 4.2 controller 和 host 协议栈
- 服务发现协议 (SDP)
- 通用访问应用 (GAP)
- 安全管理协议 (SMP)
- 低功耗蓝牙
- ATT/GATT
- HID
- 可支持所有基于 GATT 的低功耗蓝牙应用
- SPP-Like 低功耗蓝牙数据透传协议
- BLE Beacon
- A2DP/AVRCP/SPP, HSP/HFP, RFCOMM
- CVSD 和 SBC 音频编解码算法
- 蓝牙微微网 (Piconet) 和散射网 (Scatternet)

## 1.3 MCU 和高级特性

### 1.3.1 CPU 和存储

- Xtensa® 32-bit LX6 单/双核处理器, 运算能力高达 600 DMIPS
- 448 KB ROM

- 520 KB SRAM
- RTC 中 16 KB SRAM
- QSPI 最多可连接 4 个 Flash/SRAM，每个 Flash 最大为 16 MB
- 供电电压：2.3V 到 3.6V

### 1.3.2 时钟和定时器

- 内置 8 MHz 振荡器，支持自校准
- 内置 RC 振荡器，支持自校准
- 支持外置 2 MHz 至 60 MHz 的晶振（如果使用 Wi-Fi/蓝牙功能，则目前仅支持 40 MHz 晶振）
- 支持外置 32 kHz 晶振，用于 RTC，支持自校准
- 2 个定时器群组，每组包括 2 个 64-bit 通用定时器和 1 个主系统看门狗
- 具有次秒级精度的 RTC 定时器
- RTC 看门狗

### 1.3.3 高级外设接口

- 12-bit SAR ADC，多达 18 个通道
- 2 个 8-bit D/A 转换器
- 10 个触摸传感器
- 温度传感器
- 4 个 SPI
- 2 个 I2S
- 2 个 I2C
- 3 个 UART
- 1 个 Host SD/eMMC/SDIO
- 1 个 Slave SDIO/SPI
- 带有专用 DMA 的以太网 MAC 接口，支持 IEEE 1588
- CAN 2.0
- IR (TX/RX)
- 电机 PWM
- LED PWM，多达 16 个通道
- 霍尔传感器
- 超低噪声前置模拟放大器

### 1.3.4 安全机制

- 支持所有 IEEE 802.11 的安全特性，包括 WPA、WPA/WPA2 和 WAPI
- 安全启动

- Flash 加密
- 1024-bit OTP, 用户可用的高达 768 bit
- 加密硬件加速器:
  - AES
  - HASH (SHA-2) 库
  - RSA
  - ECC
  - 随机数生成器 (RNG)

### 1.3.5 开发支持

- 支持快速线上编程的 SDK 固件
- 基于 GCC 的开源工具链

#### 说明:

更多信息请参考第 8 章: 学习资源。

## 1.4 应用

- 通用低功耗 IoT 传感器 Hub
- 通用低功耗 IoT 记录器
- 相机的视频流传输
- OTT 电视盒/机顶盒设备
- 音乐播放器
  - 网络音乐播放器
  - 音频流媒体设备
- Wi-Fi 玩具
  - 计数器
  - 玩具防丢器
- Wi-Fi 语音识别设备
- 耳麦
- 智能插座
- 家庭自动化
- Mesh 网络
- 工业无线控制
- 婴儿监控器
- 可穿戴电子产品

- Wi-Fi 位置感知设备
- 安全 ID 标签
- 健康医疗
  - 运动监测和防丢报警器
  - 温度记录仪

## 1.5 功能框图

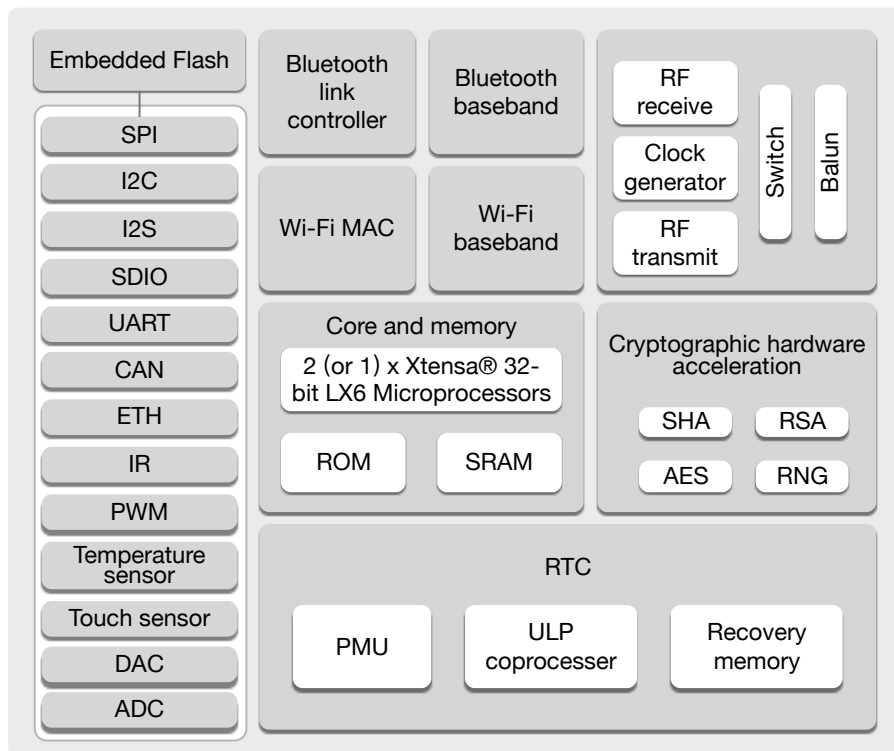


图 1: 功能框图

**说明:**

Embedded Flash 和 CPU 数量在不同型号的芯片之间有差异，详见章节[产品型号和订购信息](#)。

## 2. 管脚定义

### 2.1 管脚布局

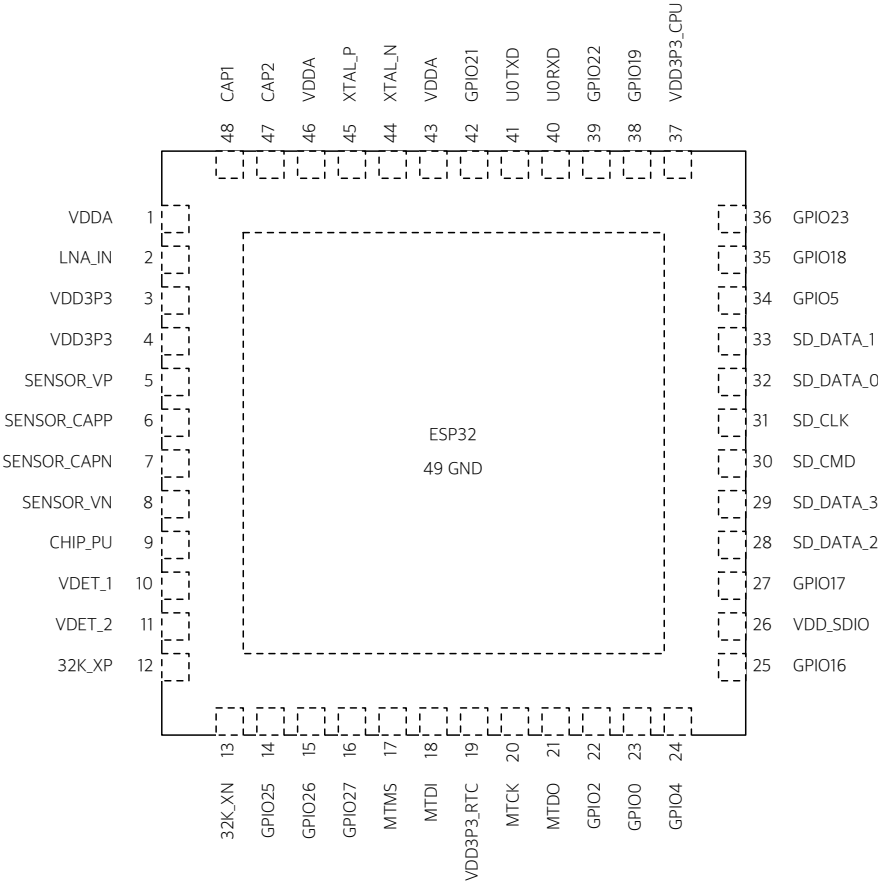


图 2: ESP32 管脚布局 (封装为 QFN 6\*6)

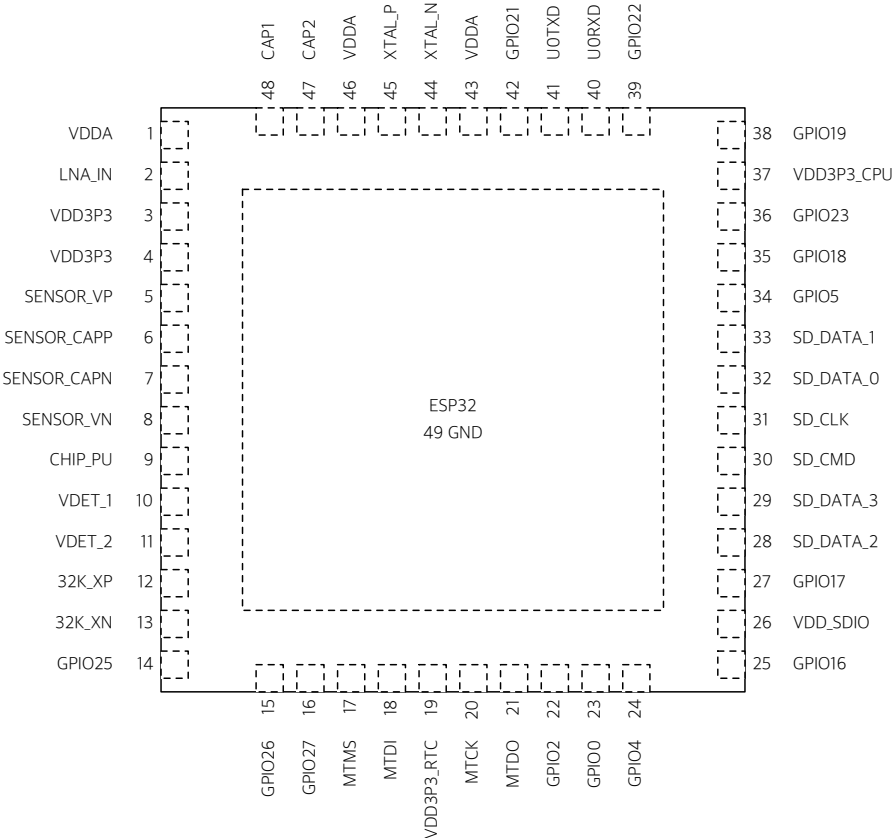


图 3: ESP32 管脚布局 (封装为 QFN 5\*5)

**说明:**  
关于 ESP32 芯片型号对应的封装规格, 请参考章节[产品型号和订购信息](#)。

## 2.2 管脚描述

表 2: 管脚描述

名称	编号	类型	功能
模拟			
VDDA	1	P	模拟电源 (2.3V ~ 3.6V)
LNA_IN	2	I/O	射频输入和输出
VDD3P3	3	P	放大器电源 (2.3V ~ 3.6V)
VDD3P3	4	P	放大器电源 (2.3V ~ 3.6V)
VDD3P3_RTC			
SENSOR_VP	5	I	GPIO36、ADC_PRE_AMP、ADC1_CH0、RTC_GPIO0 注意: 作为 ADC_PRE_AMP 使用时, 将 270 pF 电容从 SENSOR_VP 连接到 SENSOR_CAPP 上。
SENSOR_CAPP	6	I	GPIO37、ADC_PRE_AMP、ADC1_CH1、RTC_GPIO1 注意: 作为 ADC_PRE_AMP 使用时, 将 270 pF 电容从 SENSOR_VP 连接到 SENSOR_CAPP 上。

名称	编号	类型	功能
SENSOR_CAPN	7	I	GPIO38、ADC1_CH2、ADC_PRE_AMP、RTC_GPIO2 注意：作为 ADC_PRE_AMP 使用时，将 270 pF 电容从 SENSOR_VN 连接到 SENSOR_CAPN 上。
SENSOR_VN	8	I	GPIO39、ADC1_CH3、ADC_PRE_AMP、RTC_GPIO3 注意：作为 ADC_PRE_AMP 使用时，将 270 pF 电容从 SENSOR_VN 连接到 SENSOR_CAPN 上。
CHIP_PU	9	I	芯片使能（高电平有效） 高电平：上电，芯片正常工作； 低电平：断电，芯片以最小功率工作； 注意：不能让 CHIP_PU 管脚悬浮。
VDET_1	10	I	GPIO34、ADC1_CH6、RTC_GPIO4
VDET_2	11	I	GPIO35、ADC1_CH7、RTC_GPIO5
32K_XP	12	I/O	GPIO32、32K_XP（32.768 kHz 晶振输入），ADC1_CH4、TOUCH9、RTC_GPIO9
32K_XN	13	I/O	GPIO33、32K_XN（32.768 kHz 晶振输出），ADC1_CH5、TOUCH8、RTC_GPIO8
GPIO25	14	I/O	GPIO25、DAC_1、ADC2_CH8、RTC_GPIO6、EMAC_RXD0
GPIO26	15	I/O	GPIO26、DAC_2、ADC2_CH9、RTC_GPIO7、EMAC_RXD1
GPIO27	16	I/O	GPIO27、ADC2_CH7、TOUCH7、RTC_GPIO17、EMAC_RX_DV
MTMS	17	I/O	GPIO14、ADC2_CH6、TOUCH6、RTC_GPIO16、MTMS、HSPI-CLK、HS2_CLK、SD_CLK、EMAC_TXD2
MTDI	18	I/O	GPIO12、ADC2_CH5、TOUCH5、RTC_GPIO15、MTDI、HSPIQ、HS2_DATA2、SD_DATA2、EMAC_TXD3
VDD3P3_RTC	19	P	RTC IO 电源输入（1.8V ~ 3.6V）
MTCK	20	I/O	GPIO13、ADC2_CH4、TOUCH4、RTC_GPIO14、MTCK、HSPID、HS2_DATA3、SD_DATA3、EMAC_RX_ER
MTDO	21	I/O	GPIO15、ADC2_CH3、TOUCH3、RTC_GPIO13、MTDO、HSPICS0、HS2_CMD、SD_CMD、EMAC_RXD3
GPIO2	22	I/O	GPIO2、ADC2_CH2、TOUCH2、RTC_GPIO12、HSPIWP、HS2_DATA0、SD_DATA0
GPIO0	23	I/O	GPIO0、ADC2_CH1、TOUCH1、RTC_GPIO11、CLK_OUT1、EMAC_TX_CLK
GPIO4	24	I/O	GPIO4、ADC2_CH0、TOUCH0、RTC_GPIO10、HSPIHD、HS2_DATA1、SD_DATA1、EMAC_TX_ER
VDD_SDIO			
GPIO16	25	I/O	GPIO16、HS1_DATA4、U2RXD、EMAC_CLK_OUT
VDD_SDIO	26	P	1.8V 或 VDD3P3_RTC 电源输出
GPIO17	27	I/O	GPIO17、HS1_DATA5、U2TXD、EMAC_CLK_OUT_180
SD_DATA_2	28	I/O	GPIO9、SD_DATA2、SPIHD、HS1_DATA2、U1RXD
SD_DATA_3	29	I/O	GPIO10、SD_DATA3、SPIWP、HS1_DATA3、U1TXD
SD_CMD	30	I/O	GPIO11、SD_CMD、SPICS0、HS1_CMD、U1RTS
SD_CLK	31	I/O	GPIO6、SD_CLK、SPICLK、HS1_CLK、U1CTS
SD_DATA_0	32	I/O	GPIO7、SD_DATA0、SPIQ、HS1_DATA0、U2RTS
SD_DATA_1	33	I/O	GPIO8、SD_DATA1、SPID、HS1_DATA1、U2CTS
VDD3P3_CPU			



名称	编号	类型	功能
GPIO5	34	I/O	GPIO5、VSPICS0、HS1_DATA6、EMAC_RX_CLK
GPIO18	35	I/O	GPIO18、VSPICLK、HS1_DATA7
GPIO23	36	I/O	GPIO23、VSPID、HS1_STROBE
VDD3P3_CPU	37	P	CPU IO 电源输入 (1.8V ~ 3.6V)
GPIO19	38	I/O	GPIO19、VSPIQ、U0CTS、EMAC_TXD0
GPIO22	39	I/O	GPIO22、VSPIWP、U0RTS、EMAC_TXD1
U0RXD	40	I/O	GPIO3、U0RXD、CLK_OUT2
U0TXD	41	I/O	GPIO1、U0TXD、CLK_OUT3、EMAC_RXD2
GPIO21	42	I/O	GPIO21、VSPIHD、EMAC_TX_EN
模拟			
VDDA	43	P	模拟电源 (2.3V ~ 3.6V)
XTAL_N	44	O	外部晶振输出
XTAL_P	45	I	外部晶振输入
VDDA	46	P	PLL 数字电源 (2.3V ~ 3.6V)
CAP2	47	I	并联 3 nF 电容和 20 k $\Omega$ 电阻到 CAP1
CAP1	48	I	串联 10 nF 电容到地
GND	49	P	接地

**说明:**

- ESP32-D2WD 的管脚 GPIO16, GPIO17, SD\_CMD, SD\_CLK, SD\_DATA\_0 和 SD\_DATA\_1 用于连接嵌入式 Flash, 不建议用于其他功能。
- 关于 ESP32 的 IO MUX、Ethernet MAC、GPIO Matrix 和 Strapping 管脚清单, 请查看 [ESP32 管脚清单](#)。

## 2.3 电源管理

ESP32 的数字管脚可分为 3 种不同的电源域:

- VDD3P3\_RTC
- VDD3P3\_CPU
- VDD\_SDIO

VDD3P3\_RTC 同时是 RTC 和 CPU 的输入电源。VDD3P3\_CPU 是 CPU 的输入电源。

VDD\_SDIO 与一个内置 LDO 的输出相连, 该内置 LDO 的输入是 VDD3P3\_RTC。当 VDD\_SDIO 与 VDD3P3\_RTC 连接在相同的电源上, 内置 LDO 会自动关闭。

内置 LDO 可被配置成 1.8V 或与 VDD3P3\_RTC 相同的电压。在 Deep-sleep 模式下, 为了使 Flash 电流降到最低, 可以通过软件关闭内置 LDO。

**说明:**

- ESP32 的 Pin9 CHIP\_PU 使能管脚上电要晚于系统电源 3.3V 上电, 延时时间 (T) 可由 RC 参数计算而得。RC 电路设计可参考 [《ESP-WROOM-32 技术规格书》](#) 中 **ESP-WROOM-32 外设原理图**。
- CHIP\_PU 用于芯片复位, 输入电平要求低于 0.6V, 并且持续 200  $\mu$ s 以上。
- ESP32 工作电压范围为 2.3V ~ 3.6V; 使用单电源供电时, 建议电源电压为 3.3V, 最大输出电流可达 500 mA 及以上。

## 2.4 Strapping 管脚

ESP32 共有 5 个 Strapping 管脚。

- MTDI/GPIO12: 内部下拉
- GPIO0: 内部上拉
- GPIO2: 内部下拉
- MTD0/GPIO15: 内部上拉
- GPIO5: 内部上拉

软件可以读取寄存器“GPIO\_STRAPPING”中这 5 个位的值。

在芯片上电复位过程中, Strapping 管脚对电平采样并存储到锁存器中, 锁存为“0”或“1”, 并一直保持到芯片掉电或关闭。

每一个 Strapping 管脚都会连接内部上拉/下拉。如果一个 Strapping 管脚没有外部连接或者连接的外部线路处于高阻抗状态, 内部弱上拉/下拉将决定 Strapping 管脚输入电平的默认值。

为改变 Strapping 比特的值, 用户可以应用外部下拉/上拉电阻, 或者应用主机 MCU 的 GPIO 控制 ESP32 上电复位时的 Strapping 管脚电平。

复位后, Strapping 管脚和普通管脚功能相同。

配置 Strapping 管脚的详细启动模式请参阅表 3。

表 3: Strapping 管脚

内置 LDO (VDD_SDIO) 电压					
管脚	默认	3.3V		1.8V	
MTDI/GPIO12	下拉	0		1	
系统启动模式					
管脚	默认	SPI Flash 启动模式		下载启动模式	
GPIO0	上拉	1		0	
GPIO2	下拉	无关项		0	
系统启动过程中，U0TXD 输出 log 打印信息					
管脚	默认	U0TXD 翻转		U0TXD 静止	
MTDO/GPIO15	上拉	1		0	
SDIO 从机信号输入输出时序					
管脚	默认	下降沿输入 下降沿输出	下降沿输入 上升沿输出	上升沿输入 下降沿输出	上升沿输入 上升沿输出
MTDO/GPIO15	上拉	0	0	1	1
GPIO5	上拉	0	1	0	1

### 说明:

- 固件可以通过配置一些寄存器比特位, 在启动后改变“内置 LDO (VDD\_SDIO) 电压”和“SDIO 从机信号输入输出时序”的设定。
- 内置的 Flash 工作电压是 1.8V, 带有内置 Flash 的型号芯片上电时需要将 MTDI/GPIO12 拉高。

## 3. 功能描述

本章描述 ESP32 的具体功能。

### 3.1 CPU 和存储

#### 3.1.1 CPU

ESP32 搭载低功耗 Xtensa® LX6 32-bit 单/双核处理器，具有以下特性：

- 7 级流水线架构，支持高达 240 MHz 的时钟频率
- 16-bit/24-bit 指令集提高代码密度
- 支持浮点单元 (FPU)
- 支持 DSP 指令，例如 32-bit 放大器、32-bit 除法器 and 40-bit 累加乘法器 (MAC)
- 支持来自约 70 个中断源的 32 个中断向量

单/双核处理器接口包括：

- Xtensa RAM/ROM 指令和数据接口
- 用于快速访问外部寄存器的 Xtensa 本地存储接口
- 具有内外中断源的中断
- 用于调试的 JTAG 接口

#### 3.1.2 片上存储

ESP32 片上存储包括：

- 448 KB 的 ROM，用于程序启动和内核功能调用
- 用于数据和指令存储的 520 KB 片上 SRAM
- RTC 中 8 KB 的 SRAM，即 RTC 慢速存储器，可以在 Deep-sleep 模式下被协处理器访问
- RTC 中 8 KB 的 SRAM，即 RTC 快速存储器，可以在 Deep-sleep 模式下 RTC 启动时用于数据存储以及被主 CPU 访问
- 1 kbit 的 eFuse，其中 256 bit 为系统专用（MAC 地址和芯片设置）；其余 768 bit 保留给用户应用，这些应用包括 Flash 加密和芯片 ID
- 嵌入式 Flash

##### 说明：

- ESP32 芯片是否有嵌入式 Flash，以及嵌入式 Flash 的大小与芯片型号有关。详见章节[产品型号和订购信息](#)。
- 在本手册中提到的 ESP32 芯片型号中，ESP32-D2WD 带有 16 Mbit 的嵌入式 Flash，通过管脚 GPIO16, GPIO17, SD\_CMD, SD\_CLK, SD\_DATA\_0 和 SD\_DATA\_1 连接。其他型号无嵌入式 Flash。

3.1.3 外部 Flash 和 SRAM

ESP32 最多支持 4 个 16 MB 的外部 QSPI Flash 和静态随机存储器 (SRAM)，具有基于 AES 的硬件加密功能，从而保护开发者的程序和数据。

ESP32 可通过高速缓存访问外部 QSPI Flash 和 SRAM：

- 高达 16 MB 的外部 Flash 映射到 CPU 代码空间，支持 8-bit、16-bit 和 32-bit 访问，并可执行代码。
- 高达 8 MB 的外部 Flash 和 SRAM 映射到 CPU 数据空间，支持 8-bit、16-bit 和 32-bit 访问。Flash 仅支持读操作，SRAM 可支持读写操作。

**说明：**  
带有嵌入式 Flash 的 ESP32 芯片型号不支持外部 Flash 映射到代码空间。

3.1.4 存储器映射

ESP32 的地址映射结构如图 4 所示。ESP32 存储器和外设地址映射如表 4 所示。

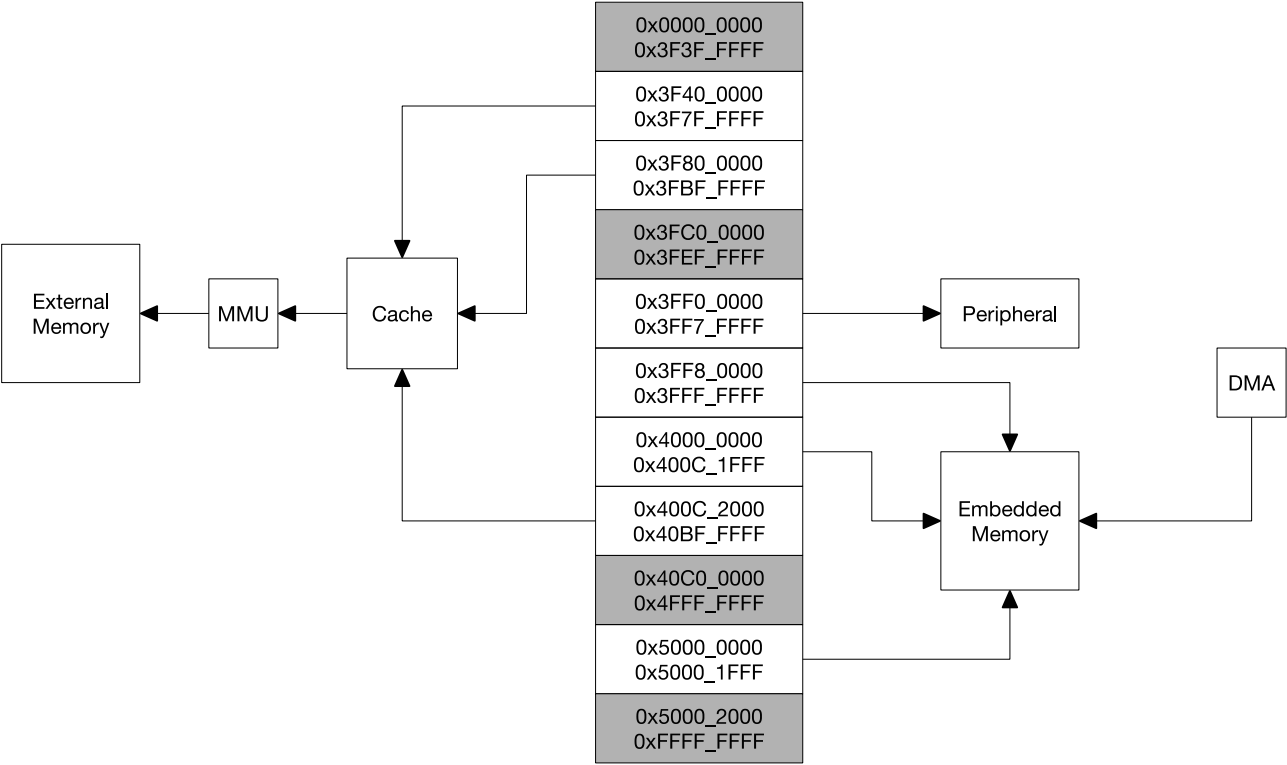


图 4: 地址映射结构

表 4: 存储器和外设地址映射

种类	目标	开始地址	结束地址	大小
片上存储器	内部 ROM 0	0x4000_0000	0x4005_FFFF	384 KB
	内部 ROM 1	0x3FF9_0000	0x3FF9_FFFF	64 KB
	内部 SRAM 0	0x4007_0000	0x4009_FFFF	192 KB
	内部 SRAM 1	0x3FFE_0000	0x3FFF_FFFF	128 KB
		0x400A_0000	0x400B_FFFF	
	内部 SRAM 2	0x3FFA_E000	0x3FFD_FFFF	200 KB
	RTC 快速存储器	0x3FF8_0000	0x3FF8_1FFF	8 KB
		0x400C_0000	0x400C_1FFF	
	RTC 慢速存储器	0x5000_0000	0x5000_1FFF	8 KB
片外存储器	外部 Flash	0x3F40_0000	0x3F7F_FFFF	4 MB
		0x400C_2000	0x40BF_FFFF	11 MB
	外部 SRAM	0x3F80_0000	0x3FBF_FFFF	248 KB
外设	DPort 寄存器	0x3FF0_0000	0x3FF0_0FFF	4 KB
	AES 加速器	0x3FF0_1000	0x3FF0_1FFF	4 KB
	RSA 加速器	0x3FF0_2000	0x3FF0_2FFF	4 KB
	SHA 加速器	0x3FF0_3000	0x3FF0_3FFF	4 KB
	安全启动	0x3FF0_4000	0x3FF0_4FFF	4 KB
	Cache MMU Table	0x3FF1_0000	0x3FF1_3FFF	16 KB
	PID 控制器	0x3FF1_F000	0x3FF1_FFFF	4 KB
	UART0	0x3FF4_0000	0x3FF4_0FFF	4 KB
	SPI1	0x3FF4_2000	0x3FF4_2FFF	4 KB
	SPI0	0x3FF4_3000	0x3FF4_3FFF	4 KB
	GPIO	0x3FF4_4000	0x3FF4_4FFF	4 KB
	RTC	0x3FF4_8000	0x3FF4_8FFF	4 KB
	IO MUX	0x3FF4_9000	0x3FF4_9FFF	4 KB
	SDIO Slave	0x3FF4_B000	0x3FF4_BFFF	4 KB
	UDMA1	0x3FF4_C000	0x3FF4_CFFF	4 KB
	I2S0	0x3FF4_F000	0x3FF4_FFFF	4 KB
	UART1	0x3FF5_0000	0x3FF5_0FFF	4 KB
	I2C0	0x3FF5_3000	0x3FF5_3FFF	4 KB
	UDMA0	0x3FF5_4000	0x3FF5_4FFF	4 KB
	SDIO Slave	0x3FF5_5000	0x3FF5_5FFF	4 KB
	RMT	0x3FF5_6000	0x3FF5_6FFF	4 KB
	PCNT	0x3FF5_7000	0x3FF5_7FFF	4 KB
	SDIO Slave	0x3FF5_8000	0x3FF5_8FFF	4 KB
	LED PWM	0x3FF5_9000	0x3FF5_9FFF	4 KB
	eFuse 控制器	0x3FF5_A000	0x3FF5_AFFF	4 KB
	Flash 加密	0x3FF5_B000	0x3FF5_BFFF	4 KB
	PWM0	0x3FF5_E000	0x3FF5_EFFF	4 KB
	TIMG0	0x3FF5_F000	0x3FF5_FFFF	4 KB
	TIMG1	0x3FF6_0000	0x3FF6_0FFF	4 KB

种类	目标	开始地址	结束地址	大小
外设	SPI2	0x3FF6_4000	0x3FF6_4FFF	4 KB
	SPI3	0x3FF6_5000	0x3FF6_5FFF	4 KB
	SYSCON	0x3FF6_6000	0x3FF6_6FFF	4 KB
	I2C1	0x3FF6_7000	0x3FF6_7FFF	4 KB
	SDMMC	0x3FF6_8000	0x3FF6_8FFF	4 KB
	EMAC	0x3FF6_9000	0x3FF6_AFFF	8 KB
	PWM1	0x3FF6_C000	0x3FF6_CFFF	4 KB
	I2S1	0x3FF6_D000	0x3FF6_DFFF	4 KB
	UART2	0x3FF6_E000	0x3FF6_EFFF	4 KB
	PWM2	0x3FF6_F000	0x3FF6_FFFF	4 KB
	PWM3	0x3FF7_0000	0x3FF7_0FFF	4 KB
	RNG	0x3FF7_5000	0x3FF7_5FFF	4 KB

## 3.2 定时器和看门狗

### 3.2.1 64-bit 通用定时器

ESP32 内置 4 个 64-bit 通用定时器，具有 16-bit 分频器和 64-bit 可自动重载的向上/向下计数器。

定时器特性：

- 16-bit 时钟分频器，分频系数为 2 至 65536
- 64-bit 时基计数器
- 时基计数器方向可配置：递增或递减
- 软件控制计数暂停和继续
- 定时器超时自动重载
- 软件控制的即时重载
- 电平触发中断和边沿触发中断

### 3.2.2 看门狗定时器

ESP32 中有 3 个看门狗定时器：2 个定时器模块中各有 1 个（称作主看门狗定时器，即 MWDT），RTC 模块中也有 1 个（称作 RTC 看门狗定时器，即 RWDT）。意外的软件或硬件问题会导致应用程序工作失常，而看门狗定时器可以帮助系统从中恢复。看门狗定时器有 4 个阶段。如果当前阶段超过预定时间，但没有喂狗或关闭看门狗定时器，可能引发以下 3 到 4 种动作中的 1 种。这些动作是：中断、CPU 复位、内核复位和系统复位。其中，只有 RWDT 能够触发系统复位，将复位包括 RTC 在内的整个芯片。每个阶段的超时时间长度均可单独设置。

在 Flash 启动期间，RWDT 和第一个 MWDT 会自动开启，以便检测和修复启动问题。

ESP32 看门狗具有以下特性：

- 4 个阶段，每一阶段都可被单独配置或关闭
- 各阶段时间段可被编程
- 如阶段超时，会采取 3 到 4 种可能动作中的 1 种（中断、CPU 复位、内核复位和系统复位）
- 32-bit 超时计数器

- 写保护，防止 RWDT 和 MWDT 配置被不小心改变
- SPI Flash 启动保护；如果在预定时间内，来自 SPI Flash 的启动过程没有完成，看门狗会重启整个系统。

### 3.3 系统时钟

#### 3.3.1 CPU 时钟

一旦重启，一个外置晶振时钟源会被选为默认的 CPU 时钟。这个外置晶振时钟源也会与 PLL 连接产生一个高频时钟（通常为 160 MHz）。

另外，ESP32 内置了一个 8 MHz 的振荡器，保证其在工作温度范围内都能保持精度稳定（误差小于 1%）。因此，应用程序可以从外置晶振时钟源、PLL 时钟和内置 8 MHz 时钟当中选择。根据应用程序不同，被选择的时钟源直接或在分频之后驱动 CPU 时钟。

#### 3.3.2 RTC 时钟

RTC 时钟有 5 种可能的时钟源：

- 外置低速 (32 kHz) 晶振时钟
- 4 分频外置晶振时钟
- 内置 RC 振荡器（通常为 150 kHz，频率可调节）
- 内置 8 MHz 振荡器时钟
- 内置 31.25 kHz 时钟（由内置 8 MHz 振荡器时钟经 256 分频生成）

当芯片处于正常功耗模式且需要更快速的 CPU 访问时，应用程序可选择 4 分频外部高速晶振时钟或者内置 8 MHz 振荡器时钟。当芯片在低功耗模式下运行时，应用程序可选择外部低速 (32 kHz) 晶振时钟、内置 RC 振荡器时钟或内置 31.25 kHz 时钟。

#### 3.3.3 音频 PLL 时钟

音频时钟由超低噪声 fractional-N PLL 生成。根据以下公式，音频 PLL 的输出频率可配置范围是 16 MHz 至 128 MHz：

$$f_{\text{out}} = \frac{f_{\text{xtal}}(\text{sdm2} + \frac{\text{sdm1}}{2^8} + \frac{\text{sdm0}}{2^{16}} + 4)}{2(\text{odiv} + 2)}$$

$f_{\text{out}}$  其中， $f_{\text{out}}$  表示输出频率， $f_{\text{xtal}}$  表示晶振的频率，sdm2、sdm1、sdm0 和 odiv 都是整数值，由寄存器配置。

### 3.4 射频

ESP32 射频包含以下主要模块：

- 2.4 GHz 接收器
- 2.4 GHz 发射器
- 偏置和线性稳压器
- Balun 和收发切换器

- 时钟生成器

#### 3.4.1 2.4 GHz 接收器

2.4 GHz 接收器将 2.4 GHz 射频信号降频为正交基带信号，并用 2 个高精度、高速的 ADC 将后者转为数字信号。为了适应不同的信道情况，ESP32 集成了 RF 滤波器、自动增益控制 (AGC)、DC 偏移补偿电路和基带滤波器。

#### 3.4.2 2.4 GHz 发射器

2.4 GHz 发射器将正交基带信号升频到 2.4 GHz 射频信号，使用大功率互补金属氧化物半导体 (CMOS) 功率放大器驱动天线。数字校准进一步改善了功率放大器的线性，使得 802.11b 无线传输达到 +20.5 dBm 的平均功率，802.11n 无线传输达到 +17 dBm 的平均功率，功能超强。

为了抵消射频接收器的瑕疵，ESP32 还另增了校准措施，例如：

- 载波泄露
- I/Q 相位匹配
- 基带非线性
- 射频非线性
- 天线匹配

这些内置校准措施缩短了产品的测试时间，不再需要测试设备。

#### 3.4.3 时钟生成器

时钟生成器为接收器和发射器生成 2.4 GHz 正交基带时钟信号，所有部件均集成于芯片上，包括电感、变容二极管、闭环滤波器、线性稳压器和分频器。

时钟生成器含有内置校准电路和自测电路。运用拥有自主知识产权的校准算法，正交时钟相噪特性在片上经过算法优化处理（该算法拥有自主知识产权），以确保接收器和发射器达到最佳性能。

### 3.5 Wi-Fi

ESP32 支持 TCP/IP 协议，完全遵循 802.11 b/g/n/e/i WLAN MAC 协议和 Wi-Fi Direct 标准，支持分布式控制功能 (DCF) 下的基本服务集 (BSS) STA 和 SoftAP 操作，也支持最新 Wi-Fi P2P 协议的 P2P GO 模式。

通过适当的命令，ESP32 会自动实施主动扫描、被动扫描以及 P2P 发现。通过最小化主机交互来优化有效工作周期，以实现功耗管理。

#### 3.5.1 Wi-Fi 射频和基带

ESP32 Wi-Fi 射频和基带支持以下特性：

- 802.11b 和 802.11g 数据率
- 802.11n MCS0-7 同时支持 20 MHz 和 40 MHz 带宽
- 802.11n MCS32
- 802.11n 0.4  $\mu$ s 保护间隔
- 数据率高达 150 Mbps



- 接收 STBC 2x1
- 发射功率高达 21 dBm
- 可调节的发射功率
- 天线分集与选择（软件控制硬件）

#### 3.5.2 Wi-Fi MAC

ESP32 Wi-Fi MAC 自行支持的底层协议功能如下：

- RTS/CTS 和 ACK/BA
- 分片和重组
- 帧聚合算法 AMPDU 和 AMSDU
- WMM, U-APSD
- 802.11 e: QoS 机制实现无线多媒体技术
- CCMP (CBC-MAC, 计数器模式)、TKIP (MIC, RC4)、WAPI (SMS4)、WEP (RC4) 和 CRC
- 帧封装 (802.11h/RFC 1042)
- 自动 Beacon 监测/扫描

#### 3.5.3 Wi-Fi 固件

ESP32 Wi-Fi 固件具有以下功能：

- 基础结构型网络 (Infrastructure BSS) 工作站 (Station) 模式/P2P 模式/SoftAP 模式
- P2P 发现、P2P GO、P2P GC 和 P2P 电源管理
- WPA/WPA2-Enterprise 和 WPS 驱动
- 附加 802.11i 安全特性，例如预认证和 TSN
- 针对企业级平台的认证开放接口，例如 TLS、PEAP、LEAP、SIM、AKA 或者用户自定义接口
- 时钟/电源门控与符合 802.11 标准的电源管理一起动态地适应当前连接条件，实现最小功耗
- 自适应速率回退算法基于实际信噪比 (SNR) 和丢包信息来控制最佳传输速率和发射功率
- MAC 层的自动重传和应答，以防止在慢速主模式环境中的数据包丢弃

#### 3.5.4 通信包仲裁 (PTA)

ESP32 拥有一个可配置的通信包仲裁器，支持灵活而精确的 Wi-Fi 和蓝牙共存模式。它采用频分多路复用 (FDM) 和时分复用 (TDM) 的组合机制，自动协调 2 个协议栈，主要特性如下：

- Wi-Fi 最好在 20 MHz 带宽模式下工作，以减少与蓝牙的干扰
- BT 采用 AFH（自适应跳频）来避免在 Wi-Fi 带宽内使用信道
- Wi-Fi MAC 限制 Wi-Fi 数据包的持续时间，也不以最低数据率发射长的 Wi-Fi 数据包
- 一般来讲，BT 数据包的优先级高于正常的 Wi-Fi 数据包
- 保护关键的 Wi-Fi 数据包，包括 Beacon 的收发以及 ACK/BA 的收发
- 保护最高优先级的 BT 数据包，包括查询应答、页面响应、LMP 数据和应答、Park Beacon、上一次轮询周期、SCO/eSCO 时段以及 BLE 事件序列

- Wi-Fi MAC 采用 CTS-to-self 数据包来保护蓝牙传输的持续时间
- 在 P2P GO 模式下，Wi-Fi MAC 采用缺席通知 (NoA) 数据包来关闭 Wi-Fi 传输，从而为蓝牙预留时间
- 在 STA 模式下，Wi-Fi MAC 通过带有 Power-Save 比特位的 NULL 数据包来关闭 Wi-Fi 传输，从而为蓝牙预留时间

## 3.6 蓝牙

ESP32 集成了蓝牙链路控制器和蓝牙基带，支持基带协议和其他低层链路协议，例如调制/解调、包处理、比特流处理和跳频等。

### 3.6.1 蓝牙射频和基带

ESP32 蓝牙射频和基带支持以下特性：

- Class-1、Class-2 和 Class-3 发射输出功率，动态控制范围超过 30 dB
- $\pi/4$  DQPSK 和 8 DPSK 调制
- NZIF 接收器灵敏度高，动态范围超过 98 dB
- 无需外部 PA 即可支持 Class-1 操作
- 内置 SRAM 支持全速数据传送、混合语音和数据以及完整的微微网 (Piconet) 运行
- 用于前向纠错、包头错误控制、接入码相关、CRC、解调、加密比特流生成、白化和发送脉冲成形的逻辑电路
- ACL、SCO、eSCO 和 AFH
- PCM 接口中的 A-law、 $\mu$ -law 和 CVSD 数字音频编解码
- SBC 音频编解码
- 低功耗应用的电源管理
- 带有 128-bit AES 的 SMP

### 3.6.2 蓝牙接口

- 提供 UART HCI 接口，速度高达 4 Mbps
- 提供 SDIO/SPI HCI 接口
- 为主机提供 I2C 接口以进行配置
- 提供 PCM/I2S 音频接口

### 3.6.3 蓝牙协议栈

ESP32 的蓝牙协议栈符合蓝牙 v4.2 BR/EDR 和 BLE 标准。

### 3.6.4 蓝牙链路控制器

链路控制器主要可实现 3 种操作：Standby、Connection 和 Sniff。可实现多路连接以及查询、寻呼和安全简易配对 (SSP) 等操作，因此能够组建微微网 (Piconet) 和散射网 (Scatternet)。以下为链路控制器的主要特性：

- 传统蓝牙

- 设备发现（查询和查询扫描）
- 建立连接（寻呼和寻呼扫描）
- 多连接
- 支持异步数据收发
- 同步连接 (SCO/eSCO)
- 主从切换
- 自适应跳频 (AFH) 和信道选择
- 加密广播
- 授权和加密
- 安全简易配对 (SSP)
- 多点连接和散射网管理
- Sniff（呼吸）模式
- 非连接的从模式广播（收发）
- 增强型功率控制
- Ping
- 低功耗蓝牙
  - 广播
  - 扫描
  - 多连接
  - 异步数据收发
  - 自适应跳频和信道选择
  - 连接参数更新
  - 支持扩展的数据包长度
  - 链路层加密
  - LE Ping

### 3.7 RTC 和低功耗管理

ESP32 采用了先进的电源管理技术，可以在不同的省电模式之间切换（见表 5）。

- 低功耗模式
  - Active 模式：芯片射频处于工作状态。芯片可以接收、发射和侦听信号。
  - Modem-sleep 模式：CPU 可运行，时钟可被配置。Wi-Fi/蓝牙基带和射频关闭。
  - Light-sleep 模式：CPU 暂停运行。RTC 存储器和外设以及 ULP 协处理器运行。任何唤醒事件（MAC、主机、RTC 定时器或外部中断）都会唤醒芯片。
  - Deep-sleep 模式：只有 RTC 存储器和外设处于工作状态。Wi-Fi 和蓝牙连接数据存储在 RTC 中。ULP 协处理器可以工作。

- Hibernation 模式：内置的 8 MHz 振荡器和 ULP 协处理器均被禁用。RTC 内存恢复电源被切断。只有 1 个位于慢时钟上的 RTC 时钟定时器和某些 RTC GPIO 在工作。RTC 时钟定时器或 RTC GPIO 可以将芯片从 Hibernation 模式中唤醒。

- 睡眠方式

- 关联睡眠方式：省电模式在 Active、Modem-、Light-sleep 模式之间切换。CPU、Wi-Fi、蓝牙和射频按照预设的时间间隔被唤醒，以保证 Wi-Fi/蓝牙的连接。
- 超低功耗传感器监测方式：主系统处于 Deep-sleep 模式，ULP 协处理器定期被开启或关闭来测量传感器数据。根据传感器测量到的数据，ULP 协处理器决定是否唤醒主系统。

表 5: 不同省电模式下的功能

省电模式	Active	Modem-sleep	Light-sleep	Deep-sleep	Hibernation
睡眠方式	关联睡眠方式			超低功耗 传感器监测方式	-
CPU	开启	开启	暂停	关闭	关闭
Wi-Fi/蓝牙基带和射频	开启	关闭	关闭	关闭	关闭
RTC 存储器和外设	开启	开启	开启	开启	关闭
ULP 协处理器	开启	开启	开启	开启/关闭	关闭

功耗随省电模式/睡眠方式以及功能模块的工作状态而改变（见表 6）。

表 6: 不同省电模式下的功耗

省电模式	描述	功耗
Active（射频工作）	Wi-Fi Tx packet 13 dBm ~ 21 dBm	160 ~ 260 mA
	Wi-Fi/BT Tx packet 0 dBm	120 mA
	Wi-Fi/BT Rx 和侦听	80 ~ 90 mA
	关联睡眠方式（与 Light-sleep 模式关联）	0.9 mA@DTIM3, 1.2 mA@DTIM1
Modem-sleep	CPU 处于工作状态	最大速度：20 mA
		正常速度：5 ~ 10 mA
		慢速：3 mA
Light-sleep	-	0.8 mA
Deep-sleep	ULP 协处理器处于工作状态	0.15 mA
	超低功耗传感器监测方式	25 $\mu$ A @1% duty
	RTC 定时器 +RTC 存储器	10 $\mu$ A
Hibernation	仅有 RTC 定时器处于工作状态	5 $\mu$ A
关闭	CHIP_PU 脚拉低，芯片处于关闭状态	<0.1 $\mu$ A

## 说明：

更多有关射频功耗的内容，参见第 5.2 节：射频功耗参数。

## 4. 外设接口和传感器

### 4.1 通用输入/输出接口 (GPIO)

ESP32 共有 34 个 GPIO 管脚，通过配置对应的寄存器，可以为这些管脚分配不同的功能，包括如下几类 GPIO：只有数字功能的 GPIO、带模拟功能的 GPIO 以及带电容触摸功能的 GPIO 等等。带模拟功能的 GPIO 可以被配置为数字 GPIO。带电容触摸功能的 GPIO 可被配置为数字 GPIO。

每个带数字功能的 GPIO 都可以被配置为内部上拉 / 下拉，或者被设置为高阻。当被配置为输入时，可通过读取寄存器获取输入值。输入管脚也可以被设置为通过边缘触发或电平触发来产生 CPU 中断。简言之，数字 IO 管脚都是双向、非反相和三态的，包括带有三态控制的输入和输出缓冲器。这些管脚可以复用作其他功能，例如 SDIO 接口、UART、SI 等。用于低功耗运行时，GPIO 可被设定为保持状态。

### 4.2 模/数转换器 (ADC)

ESP32 集成了 12-bit SAR ADC，共支持 18 个模拟通道输入。其中一些管脚可以通过配置可编程增益放大器，用作测量弱模拟信号。为了实现更低功耗，ESP32 的 ULP 协处理器也可以在睡眠方式下测量电压，此时，可通过设置阈值或其他触发方式唤醒 CPU。

通过适当的设置，最多可配置 18 个管脚的 ADC 和放大器，用于电压模数转换。

### 4.3 超低噪声前置模拟放大器

ESP32 集成了一个超低噪声前置模拟放大器输出给 ADC。增益率由一对片外采样电容器的大小决定。选用一个更大的电容，采样噪声会减小，但是稳定时间会延长。增益率还受到放大器的限制，放大器的最大电压增益约为 60 dB。

### 4.4 霍尔传感器

ESP32 集成的霍尔传感器是基于空穴 (N-carrier) 电阻设计的。当芯片置于电磁场中时，霍尔传感器会在电阻上横向产生一个小电压。这个小电压可由 ADC 直接测量，也可经过超低噪声前置模拟放大器放大后再由 ADC 测量。

### 4.5 数/模转换器 (DAC)

ESP32 有 2 个 8-bit DAC 通道，将 2 个数字信号分别转化为 2 个模拟电压信号输出。DAC 电路由内置电阻串和 1 个缓冲器组成。这 2 个 DAC 可以作为参考电压使用，也可以作为其它电路的电源使用。这是 2 个独立的 DAC。

### 4.6 温度传感器

温度传感器生成一个随温度变化的电压。内部 ADC 将传感器电压转化为一个数字量。

温度传感器的测量范围为 -40°C 到 125°C。由于制程漂移，每颗芯片的温度传感器补偿各不相同，而且 Wi-Fi 电路本身也会产生热量（这影响了测量结果），因此内置温度传感器只适用于那些探测温度变化而不是绝对温度的应用，也适用于校准。

不过，如果用户校准了温度传感器，并且在最低功耗的应用上使用该传感器，结果就会足够精确。

## 4.7 触摸传感器

ESP32 提供了多达 10 个电容式传感 GPIO，能够探测由手指或其他物品直接接触或接近而产生的电容差异。这种设计的低噪声特性和电路的高灵敏度支持使用相对较小的触摸板。也可以使用触摸板阵列以探测更大区域或更多点。表 7 列出了 10 个电容式传感 GPIO。

表 7: ESP32 上的电容式传感 GPIO

电容式传感信号名称	管脚名称
T0	GPIO4
T1	GPIO0
T2	GPIO2
T3	MTDO
T4	MTCK
T5	MTD1
T6	MTMS
T7	GPIO27
T8	32K_XN
T9	32K_XP

**说明:**

关于触摸传感器的设计与布局的更多内容，参见附录 A: 触摸传感器。

## 4.8 超低功耗协处理器 (ULP)

ULP 处理器和 RTC 存储器在 Deep-sleep 模式下仍保持工作状态。因此，开发者可以将 ULP 协处理器的程序存放在 RTC 存储器中，使其能够在 Deep-sleep 模式下访问外设、内置定时器和内置传感器。在 CPU 需要由外部事件、定时器或者这些事件的组合来唤醒的应用中，可用于保持最低的功耗。

## 4.9 以太网 MAC 接口

ESP32 为以太网通信提供了一个符合 IEEE-802.3-2008 标准的媒体访问控制器 (MAC) 接口。ESP32 需要一个外部物理接口 (PHY) 来连接实体 LAN 总线（双绞线、光纤等）。物理接口通过 17 个 MII 信号或 9 个 RMII 信号与 ESP32 连接。以太网 MAC 接口 (EMAC) 支持以下特性：

- 10 Mbps 和 100 Mbps 的速率
- 专用的 DMA 控制器实现以太网 MAC 接口与专用 SRAM 之间的高速传输
- 带标记的 MAC 帧（支持 VLAN）
- 半双工 (CSMA/CD) 和全双工操作
- MAC 控制子层（控制帧）
- 32-bit CRC 自动生成和消除
- 用于单播和组播地址（广播和组地址）的多种地址过滤模式
- 记录每个收发帧的 32-bit 状态码

- 内部 FIFO 用于缓冲发射和接收帧。发送 FIFO 和接收 FIFO 均为 512 字 (32-bit)
- 符合 IEEE 1588 2008 (PTP V2) 标准的硬件 PTP（精确时间协议）
- 25 MHz/50 MHz 的时钟输出

### 4.10 SD/SDIO/MMC 主机控制器

ESP32 拥有一个 SD/SDIO/MMC 主机控制器，支持以下特性：

- SD 卡 3.0 和 3.01 版本
- SDIO 3.0 版本
- CE-ATA 1.1 版本
- 多媒体卡（MMC 4.41 版本、eMMC 4.5 版本和 4.51 版本）

控制器实现了高达 80 MHz 的时钟输出，并且支持 3 种数据总线模式：1 bit、4 bit 和 8 bit。在 4-bit 数据总线模式中，可以支持 2 个 SD/SDIO/MMC4.41 卡，还支持 1 个以 1.8V 电压工作的 SD 卡。

### 4.11 SDIO/SPI 从机控制器

ESP32 集成了符合工业标准 SDIO 全速卡规格的 SD 设备接口，并允许主机控制器使用 SDIO 总线协议访问 SoC 设备。ESP32 用作 SDIO 总线上的从机。主机可以直接访问 SDIO 接口的寄存器并通过使用 DMA 引擎访问设备中的共享内存，从而不需要处理器内核即可使性能最优化。

SDIO/SPI 从机控制器具有以下特性：

- 时钟范围为 0 至 50 MHz，支持 SPI、1-bit SDIO 和 4-bit SDIO 的传输模式
- 采样和驱动的时钟边沿可配置
- 主机可直接访问的专用寄存器
- 可中断主机，启动数据传输
- 允许卡中断主机
- 支持自动填充 SDIO 总线上的发送数据，同样支持自动丢弃 SDIO 总线上的填充数据
- 字节块大小可达 512 字节
- 主机与从机间有中断向量可以相互使对方产生中断
- 用于数据传输的 DMA 链表

### 4.12 通用异步收发器 (UART)

ESP32 有 3 个 UART 接口，即 UART0、UART1 和 UART2，支持异步通信（RS232 和 RS485）和 IrDA，通信速率可达到 5 Mbps。UART 支持 CTS 和 RTS 信号的硬件管理以及软件流控（XON 和 XOFF）。3 个接口均可被 DMA 访问或者 CPU 直接访问。



### 4.13 I2C 接口

ESP32 有 2 个 I2C 总线接口，根据用户的配置，总线接口可以用作 I2C 主机或从机模式。I2C 接口支持：

- 标准模式 (100 kbit/s)
- 高速模式 (400 kbit/s)
- 速度最大可达 5 MHz，但受制于 SDA 上拉强度
- 7-bit/10-bit 寻址模式
- 双寻址模式

用户可以配置指令寄存器来控制 I2C 接口，从而实现更多灵活的应用。

### 4.14 I2S 接口

ESP32 拥有 2 个标准 I2S 接口。这 2 个接口可以以主机或从机模式，在全双工或半双工模式下工作，并且可被配置为 8/16/32/40/48-bit 的输入输出通道，支持频率从 10 kHz 到 40 MHz 的 BCK 时钟。当 1 个或 2 个 I2S 接口被配置为主机模式时，主机时钟可以输出到外部 DAC/CODEC。

2 个 I2S 接口都有专用的 DMA 控制器。支持 PDM 和 BT PCM 接口。

### 4.15 红外遥控器

红外遥控器支持 8 通道的红外发射和接收。通过程序控制脉冲波形，遥控器可以支持多种红外协议。8 个通道共用 1 个 512 x 32-bit 的存储模块来存放收发的波形。

### 4.16 脉冲计数器

脉冲计数器通过 7 种模式捕捉脉冲并对脉冲边沿计数。内部有 8 个通道，每个通道一次可同时捕捉 4 个信号。每组 4 个输入包括 2 个脉冲信号和 2 个控制信号。当计数器达到了设定的阈值，就会产生 1 个中断。

### 4.17 脉冲宽度调制 (PWM)

PWM 控制器可以用于驱动数字马达和智能灯。该控制器包含 PWM 定时器、PWM 执行器和 1 个专用的捕捉子模组。定时器可以同步定时，也可以独立运行。每个 PWM 执行器为 1 个 PWM 通道生成波形。专用的捕捉子模组可以精确捕捉外部定时事件。

### 4.18 LED PWM

LED PWM 控制器可以生成 16 路独立的数字波形，波形的周期和占空比可配置。

16 路信号在 80 MHz APB 总线时钟下工作，其中 8 路信号可以选择使用芯片内置的 8 MHz 振荡器时钟，可在 Light-sleep 模式下工作。每路信号可选择 1 个 20-bit 定时器，定时器的计数范围可配置，在输出信号周期为 1 ms 时，占空比的精确度可以高达 16 bit。

通过软件可以实时改变占空比。另外，每路 LED PWM 支持自动步进式地增加或减少占空比，可以用于 LED RGB 彩色梯度发生器。



## 4.19 串行外设接口 (SPI)

ESP32 共有 3 组 SPI (SPI、HSPI 和 VSPI) 接口，可以在主机或从机模式，在 1-line 全双工或 1/2/4-line 单工通信模式下工作，作为通用 SPI 支持以下特性：

- 4 种定时模式的 SPI 格式传输，模式取决于极性 (POL) 和相位 (PHA)
- 最高支持 80 MHz，也支持 80 MHz 的分频时钟
- 最高支持 64 Bytes 的 FIFO

所有 SPI 接口也可以用来连接外部 Flash/SRAM 和 LCD。每一个 SPI 控制器可连接到 DMA 通道。

## 4.20 硬件加速器

ESP32 配备硬件加速器，支持一些通用加密算法，比如 AES (FIPS PUB 197)、SHA (FIPS PUB 180-4)、RSA 和 ECC 等，还支持大数乘法、大数模乘等独立运算。硬件加速器支持的 RSA、ECC、大数乘法和大数模乘运算最大长度可达 4096 bits。

硬件加速器极大提高了运算速度，显著减小了软件的复杂度。硬件加速器还支持对 Flash 的加密与动态解密，确保 Flash 中的代码不被窃取。

## 5. 电气特性

**说明:**

如无特别说明，本章参数测试条件如下：VDD = 3.3V，T<sub>A</sub> = 27°C。

### 5.1 极限参数

表 8: 极限参数

参数	名称	最小值	典型值	最大值	单位
供电电压	VDD	2.3	3.3	3.6	V
输入逻辑电平低	V <sub>IL</sub>	-0.3	-	0.25×V <sub>IO</sub>	V
输入逻辑电平高	V <sub>IH</sub>	0.75×V <sub>IO</sub>	-	V <sub>IO</sub> +0.3	V
输入漏电流	I <sub>IL</sub>	-	-	50	nA
输出逻辑电平低	V <sub>OL</sub>	-	-	0.1×V <sub>IO</sub>	V
输出逻辑电平高	V <sub>OH</sub>	0.8×V <sub>IO</sub>	-	-	V
输入引脚电容	C <sub>pad</sub>	-	-	2	pF
VDDIO	V <sub>IO</sub>	1.8	3.3	3.6	V
最大驱动能力	I <sub>MAX</sub>	-	-	40	mA
存储温度范围	T <sub>STR</sub>	-40	-	150	°C
工作温度范围 *	T <sub>OPR</sub>	-40	-	125	°C

\* 由于 ESP32-D2WD 的嵌入式 Flash 工作温度为 -40°C ~ 105°C，所以 ESP32-D2WD 芯片的整体工作温度为 -40°C ~ 105°C。ESP32 系列中其他型号的芯片无嵌入式 Flash，工作温度为 -40°C ~ 125°C。

### 5.2 射频功耗参数

下列功耗数据是基于 3.0 V 电源、25°C 环境温度，在天线接口处完成的测试结果。所有发射数据均基于 90% 的占空比，在连续发射模式下测得。

表 9: 射频功耗参数

模式	最小值	典型值	最大值	单位
发送 802.11b, DSSS 1 Mbps, POUT = +19.5 dBm	-	225	-	mA
发送 802.11b, CCK 11 Mbps, POUT = +18.5 dBm	-	205	-	mA
发送 802.11g, OFDM 54 Mbps, POUT = +16 dBm	-	160	-	mA
发送 802.11n, MCS7, POUT = +14 dBm	-	152	-	mA
接收 802.11b, 包长 1024 bytes, -80 dBm	-	85	-	mA
接收 802.11g, 包长 1024 bytes, -70 dBm	-	85	-	mA
接收 802.11n, 包长 1024 bytes, -65 dBm	-	80	-	mA
接收 802.11n HT40, 包长 1024 bytes, -65 dBm	-	80	-	mA

## 5.3 Wi-Fi 射频

表 10: Wi-Fi 射频特性

参数	最小值	典型值	最大值	单位
输入频率	2412	-	2484	MHz
输出阻抗	-	30+j10	-	$\Omega$
输入反射	-	-	-10	dB
灵敏度				
72.2 Mbps PA 输出功率	15.5	16.5	17.5	dBm
11b 模式下 PA 输出功率	19.5	20.5	21.5	dBm
DSSS, 1 Mbps	-	-98	-	dBm
CCK, 11 Mbps	-	-91	-	dBm
OFDM, 6 Mbps	-	-93	-	dBm
OFDM, 54 Mbps	-	-75	-	dBm
HT20, MCS0	-	-93	-	dBm
HT20, MCS7	-	-73	-	dBm
HT40, MCS0	-	-90	-	dBm
HT40, MCS7	-	-70	-	dBm
MCS32	-	-89	-	dBm
邻道抑制				
OFDM, 6 Mbps	-	37	-	dB
OFDM, 54 Mbps	-	21	-	dB
HT20, MCS0	-	37	-	dB
HT20, MCS7	-	20	-	dB

## 5.4 蓝牙射频

### 5.4.1 接收器 - 基础数据率 (BR)

表 11: 接收器特性 - 基础数据率 (BR)

参数	条件	最小值	典型值	最大值	单位
灵敏度 @0.1% BER	-	-	-98	-	dBm
最大接收信号 @0.1% BER	-	0	-	-	dBm
共信道抑制比 C/I	-	-	+7	-	dB
邻道选择性抑制比 C/I	F = F0 + 1 MHz	-	-	-6	dB
	F = F0 - 1 MHz	-	-	-6	dB
	F = F0 + 2 MHz	-	-	-25	dB
	F = F0 - 2 MHz	-	-	-33	dB
	F = F0 + 3 MHz	-	-	-25	dB
	F = F0 - 3 MHz	-	-	-45	dB
带外数据带阻	30 MHz ~ 2000 MHz	-10	-	-	dBm
	2000 MHz ~ 2400 MHz	-27	-	-	dBm
	2500 MHz ~ 3000 MHz	-27	-	-	dBm
	3000 MHz ~ 12.5 GHz	-10	-	-	dBm
互调	-	-36	-	-	dBm

## 5.4.2 发射器 - 基础数据率 (BR)

表 12: 发射器特性 - 基础数据率 (BR)

参数	条件	最小值	典型值	最大值	单位
射频发射功率	-	-	+4	+4	dBm
射频功率控制范围	-	-	25	-	dB
20 dB 带宽	-	-	0.9	-	MHz
邻道发射功率	F = F0 + 1 MHz	-	-24	-	dBm
	F = F0 - 1 MHz	-	-16.1	-	dBm
	F = F0 + 2 MHz	-	-40.8	-	dBm
	F = F0 - 2 MHz	-	-35.6	-	dBm
	F = F0 + 3 MHz	-	-45.7	-	dBm
	F = F0 - 3 MHz	-	-40.2	-	dBm
	F = F0 + > 3 MHz	-	-45.6	-	dBm
	F = F0 - > 3 MHz	-	-44.6	-	dBm
$\Delta f_{1avg}$	-	-	-	155	kHz
$\Delta f_{2max}$	-	133.7	-	-	kHz
$\Delta f_{2avg}/\Delta f_{1avg}$	-	-	0.92	-	-
ICFT	-	-	-7	-	kHz
漂移速率	-	-	0.7	-	kHz/50 $\mu$ s
偏移 (1 槽数据包)	-	-	6	-	kHz
偏移 (5 槽数据包)	-	-	6	-	kHz

## 5.4.3 接收器 - 增强数据率 (EDR)

表 13: 接收器特性 - 增强数据率 (EDR)

参数	条件	最小值	典型值	最大值	单位
$\pi/4$ DQPSK					
灵敏度 @0.01% BER	-	-	-98	-	dBm
最大接收信号 @0.1% BER	-	-	0	-	dBm
共信道抑制比 C/I	-	-	11	-	dB
邻道选择性抑制比 C/I	F = F0 + 1 MHz	-	-7	-	dB
	F = F0 - 1 MHz	-	-7	-	dB
	F = F0 + 2 MHz	-	-25	-	dB
	F = F0 - 2 MHz	-	-35	-	dB
	F = F0 + 3 MHz	-	-25	-	dB
	F = F0 - 3 MHz	-	-45	-	dB
8DPSK					
灵敏度 @0.01% BER	-	-	-84	-	dBm
最大接收信号 @0.1% BER	-	0	-	-	dBm
共信道抑制比 C/I	-	-	18	-	dB

参数	条件	最小值	典型值	最大值	单位
邻道抑制比 C/I	$F = F_0 + 1 \text{ MHz}$	-	2	-	dB
	$F = F_0 - 1 \text{ MHz}$	-	2	-	dB
	$F = F_0 + 2 \text{ MHz}$	-	-25	-	dB
	$F = F_0 - 2 \text{ MHz}$	-	-25	-	dB
	$F = F_0 + 3 \text{ MHz}$	-	-25	-	dB
	$F = F_0 - 3 \text{ MHz}$	-	-38	-	dB

#### 5.4.4 发射器 - 增强数据率 (EDR)

表 14: 发射器特性 - 增强数据率 (EDR)

参数	条件	最小值	典型值	最大值	单位
最大射频发射功率	-	-	+2	-	dBm
相对发射控制	-	-	-1.5	-	dB
$\pi/4$ DQPSK max w0	-	-	-0.72	-	kHz
$\pi/4$ DQPSK max wi	-	-	-6	-	kHz
$\pi/4$ DQPSK max  wi + w0	-	-	-7.42	-	kHz
8DPSK max w0	-	-	0.7	-	kHz
8DPSK max wi	-	-	-9.6	-	kHz
8DPSK max  wi + w0	-	-	-10	-	kHz
$\pi/4$ DQPSK 调制精度	RMS DEVM	-	4.28	-	%
	99% DEVM	-	-	30	%
	Peak DEVM	-	13.3	-	%
8 DPSK 调制精度	RMS DEVM	-	5.8	-	%
	99% DEVM	-	-	20	%
	Peak DEVM	-	14	-	%
带内杂散发射	$F = F_0 + 1 \text{ MHz}$	-	-34	-	dBm
	$F = F_0 - 1 \text{ MHz}$	-	-40.2	-	dBm
	$F = F_0 + 2 \text{ MHz}$	-	-34	-	dBm
	$F = F_0 - 2 \text{ MHz}$	-	-36	-	dBm
	$F = F_0 + 3 \text{ MHz}$	-	-38	-	dBm
	$F = F_0 - 3 \text{ MHz}$	-	-40.3	-	dBm
	$F = F_0 \pm > 3 \text{ MHz}$	-	-	-41.5	dBm
EDR 差分相位编码	-	-	100	-	%

### 5.5 低功耗蓝牙射频

#### 5.5.1 接收器

表 15: 低功耗蓝牙接收器特性

参数	条件	最小值	典型值	最大值	单位
灵敏度 @0.1% BER	-	-	-98	-	dBm
最大接收信号 @0.1% BER	-	0	-	-	dBm
共信道抑制比 C/I	-	-	+10	-	dB

参数	条件	最小值	典型值	最大值	单位
邻道抑制比 C/I	$F = F_0 + 1 \text{ MHz}$	-	-5	-	dB
	$F = F_0 - 1 \text{ MHz}$	-	-5	-	dB
	$F = F_0 + 2 \text{ MHz}$	-	-25	-	dB
	$F = F_0 - 2 \text{ MHz}$	-	-35	-	dB
	$F = F_0 + 3 \text{ MHz}$	-	-25	-	dB
	$F = F_0 - 3 \text{ MHz}$	-	-45	-	dB
带外数据带阻	30 MHz ~ 2000 MHz	-10	-	-	dBm
	2000 MHz ~ 2400 MHz	-27	-	-	dBm
	2500 MHz ~ 3000 MHz	-27	-	-	dBm
	3000 MHz ~ 12.5 GHz	-10	-	-	dBm
互调	-	-36	-	-	dBm

## 5.5.2 发射器

表 16: 低功耗蓝牙发射器特性

参数	条件	最小值	典型值	最大值	单位
射频发射功率	-	-	+7.5	+10	dBm
射频功率控制范围	-	-	25	-	dB
邻道发射功率	$F = F_0 + 1 \text{ MHz}$	-	-14.6	-	dBm
	$F = F_0 - 1 \text{ MHz}$	-	-12.7	-	dBm
	$F = F_0 + 2 \text{ MHz}$	-	-44.3	-	dBm
	$F = F_0 - 2 \text{ MHz}$	-	-38.7	-	dBm
	$F = F_0 + 3 \text{ MHz}$	-	-49.2	-	dBm
	$F = F_0 - 3 \text{ MHz}$	-	-44.7	-	dBm
	$F = F_0 + > 3 \text{ MHz}$ $F = F_0 - > 3 \text{ MHz}$	-	-50 -50	-	dBm dBm
$\Delta f_{1\text{avg}}$	-	-	-	265	kHz
$\Delta f_{2\text{max}}$	-	247	-	-	kHz
$\Delta f_{2\text{avg}}/\Delta f_{1\text{avg}}$	-	-	-0.92	-	-
ICFT	-	-	-10	-	kHz
漂移速率	-	-	0.7	-	kHz/50 $\mu\text{s}$
偏移	-	-	2	-	kHz

# 6. 封装信息

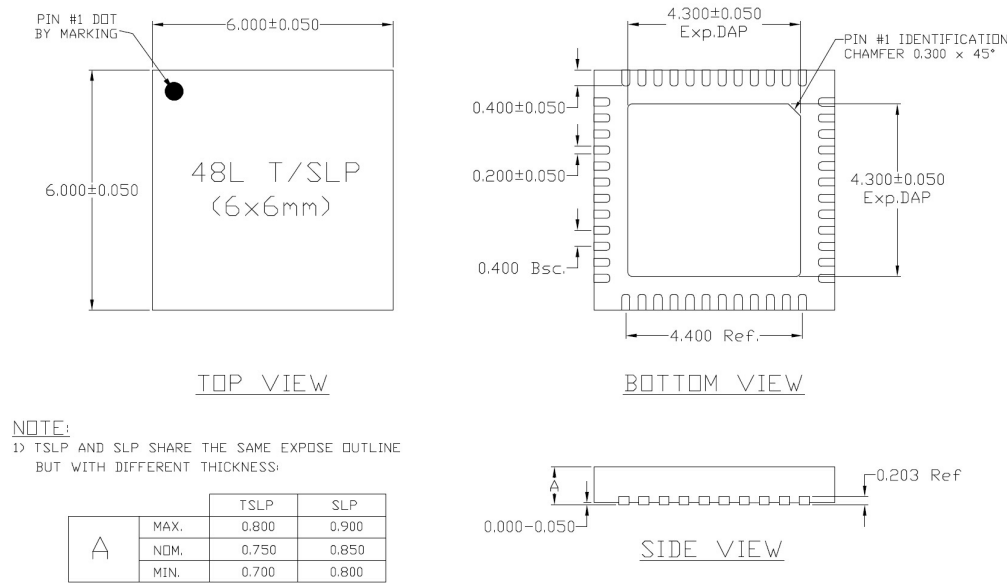


图 5: QFN48 (6x6 mm) 封装

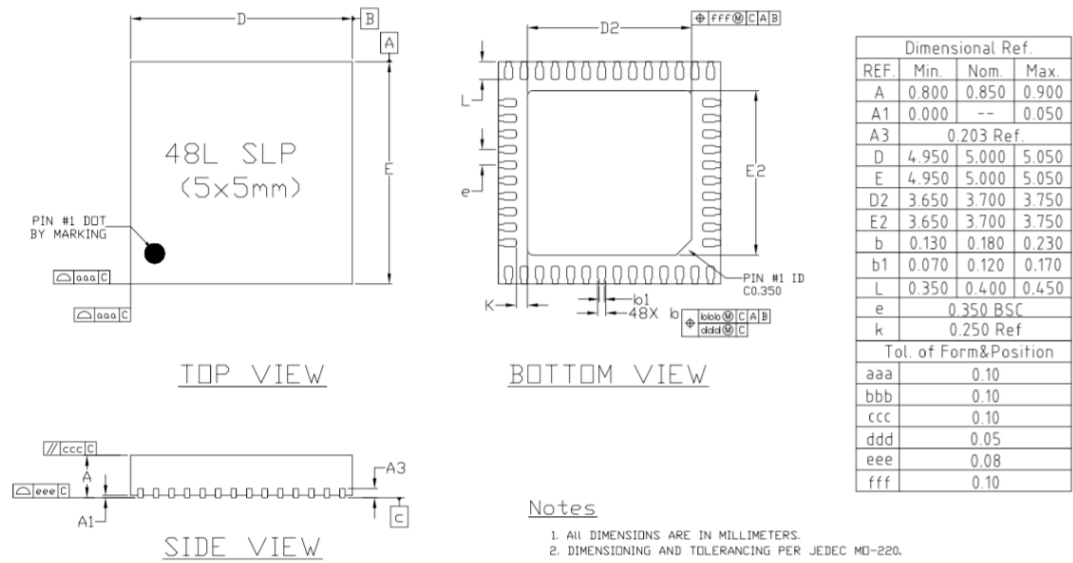


图 6: QFN48 (5x5 mm) 封装

7. 产品型号和订购信息

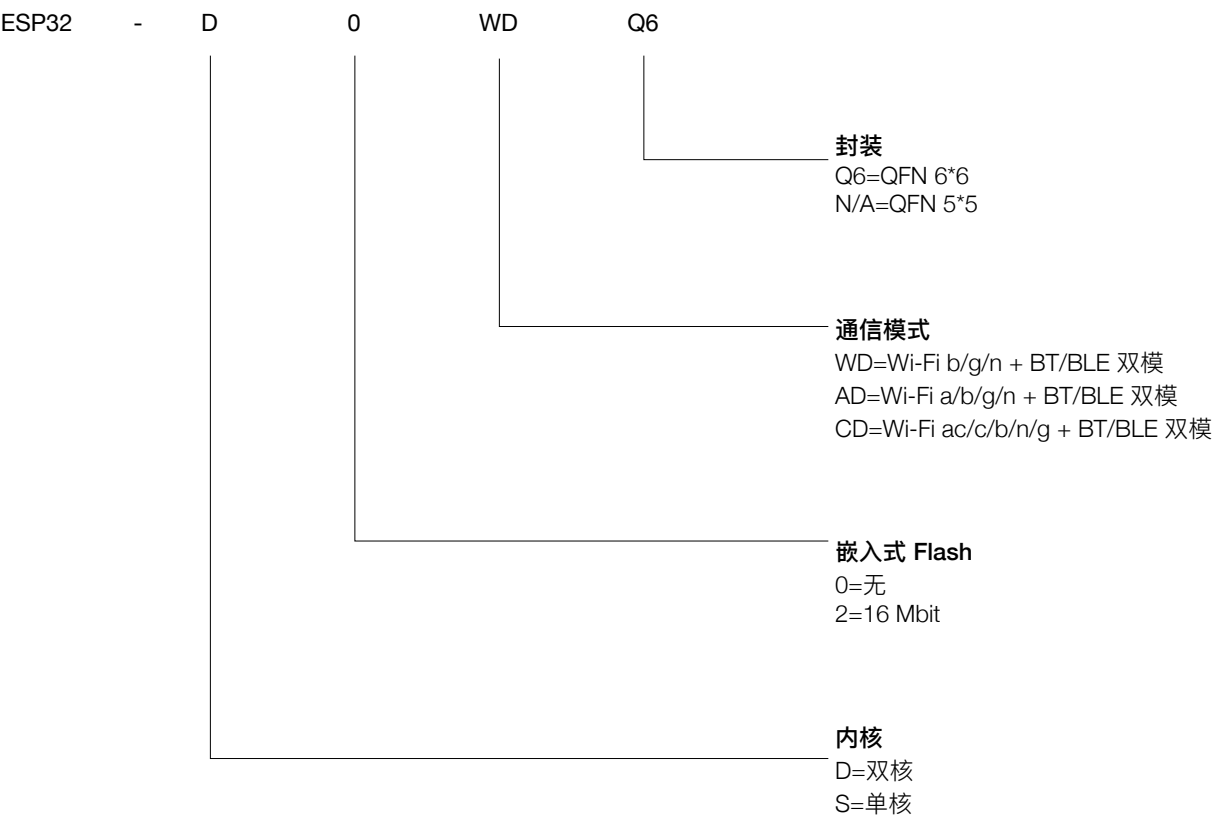


图 7: ESP32 产品型号

下表列举 ESP32 各个型号的订购信息。

表 17: 订购信息

订购型号	内核	嵌入式 Flash	通信模式	封装
ESP32-D0WDQ6	双核	无嵌入式 Flash	Wi-Fi b/g/n + BT/BLE 双模	QFN 6*6
ESP32-D0WD	双核	无嵌入式 Flash	Wi-Fi b/g/n + BT/BLE 双模	QFN 5*5
ESP32-D2WD	双核	16-Mbit 嵌入式 Flash	Wi-Fi b/g/n + BT/BLE 双模	QFN 5*5
ESP32-S0WD	单核	无嵌入式 Flash	Wi-Fi b/g/n + BT/BLE 双模	QFN 5*5



## 8. 学习资源

### 8.1 必读资料

访问以下链接可下载有关 ESP32 的文档资料。

- [ESP32 技术参考手册](#)  
该手册提供了关于 ESP32 的具体信息，包括各个功能模块的内部架构、功能描述和寄存器配置等。
- [ESP32 硬件资源](#)  
压缩包的内容主要是硬件原理图，包括 ESP32-DevKitC 的原理图，PCB 布局，制造规范和物料清单。
- [ESP32 硬件设计指南](#)  
该手册提供了 ESP32 系列产品的硬件信息，包括 ESP32 芯片，ESP-WROOM-32 模组以及 ESP32-DevKitC 开发板。
- [ESP32 AT 指令集与使用示例](#)  
该文档描述 ESP32 AT 指令集功能以及使用方法，并介绍几种常见的 AT 指令使用示例。其中 AT 指令包括基础 AT 指令，Wi-Fi 功能 AT 指令，TCP/IP 相关 AT 指令等；使用示例包括单连接 TCP 客户端，UDP 传输，透传，多连接 TCP 服务器等。

### 8.2 必备资源

以下为有关 ESP32 的必备资源。

- [ESP32 在线社区](#)  
工程师对工程师（E2E）的社区，用户可以在这里提出问题，分享知识，探索观点，并与其他工程师一起解决问题。
- [ESP32 Github](#)  
乐鑫拥有 Github 的 MIT 许可证，可以在 Github 上自由发布 ESP32 开发项目。ESP32 Github 帮助开发者了解利用 ESP32 开发的硬软件。
- [ESP32 工具](#)  
该页面提供了 ESP32 Flash 下载工具以及《ESP32 认证测试指南》。
- [ESP32 IDF](#)  
该页面提供了 ESP32 所有版本 IDF。
- [ESP32 资源合集](#)  
ESP32 的所有文档和工具资源。

## 附录 A - 触摸传感器

触摸传感系统建立在一个基底之上，这个基底带有电极并且连接一个平面保护层。当用户触摸保护层，就会触发电容变化，一个二进信号会生成并显示触摸是否有效。

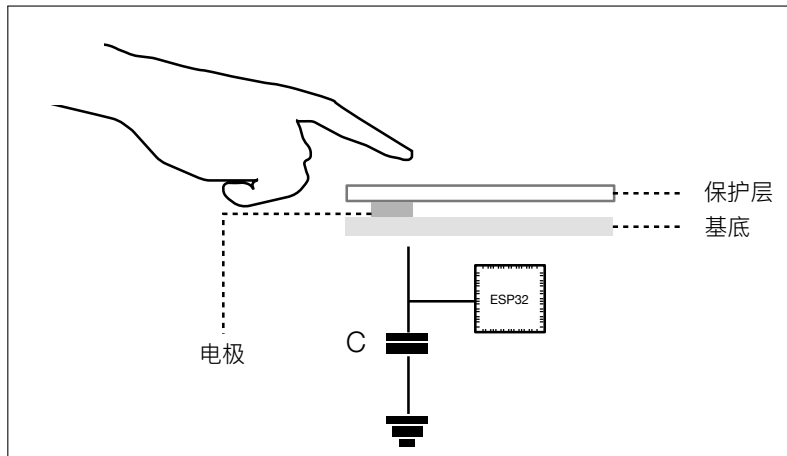


图 8: 典型的触摸传感器应用

为防止电容耦合和其他电干扰影响触摸传感系统的灵敏度，用户需要考虑以下因素：

### A.1. 电极图形

适当大小和形状的电极有助于提高系统灵敏度。常见的有圆形、椭圆形和形状类似人的指尖的电极。过大或形状不规则的电极可能导致附近电极发生错误响应。

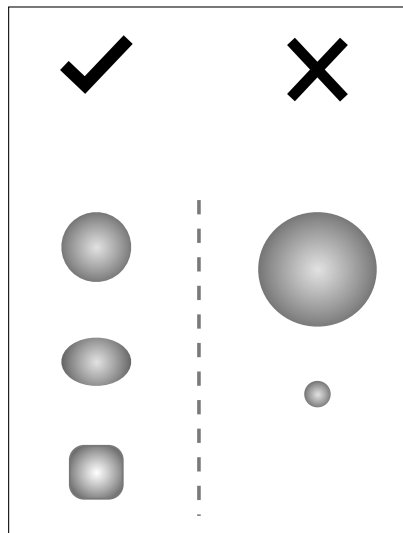


图 9: 电极图形要求

**说明：**

图 9 未按照实际比例示例。建议用户用指尖作为参考。

## A.2. PCB 布局

正确的电极感应走线布局建议如下：

- 电极之间距离过短会引起串扰和错误的触摸探测。建议电极距离至少为主板厚度的 2 倍。
- 感应走线的宽度会产生寄生电容。电容可能由于生产工艺不同存在差异。走线越细，产生的耦合电容就越少。建议保持走线尽可能细，长度不要超过 10 cm 以能容纳标准的 PCB 或 FlexPCB。
- 为避免高频信号的线间耦合，建议在同一层上平行布局传感走线并且保持走线之间的距离至少为走线宽度的 2 倍。
- 当设计一个触摸传感设备，不能有元件临近电极或在电极下面。只有电极对于触摸有敏感性。
- 不要让触摸传感设备接地。强烈建议不要在设备下面布局接地层，因为触摸传感器设备和地面之间产生的寄生电容会极大地降低灵敏度。

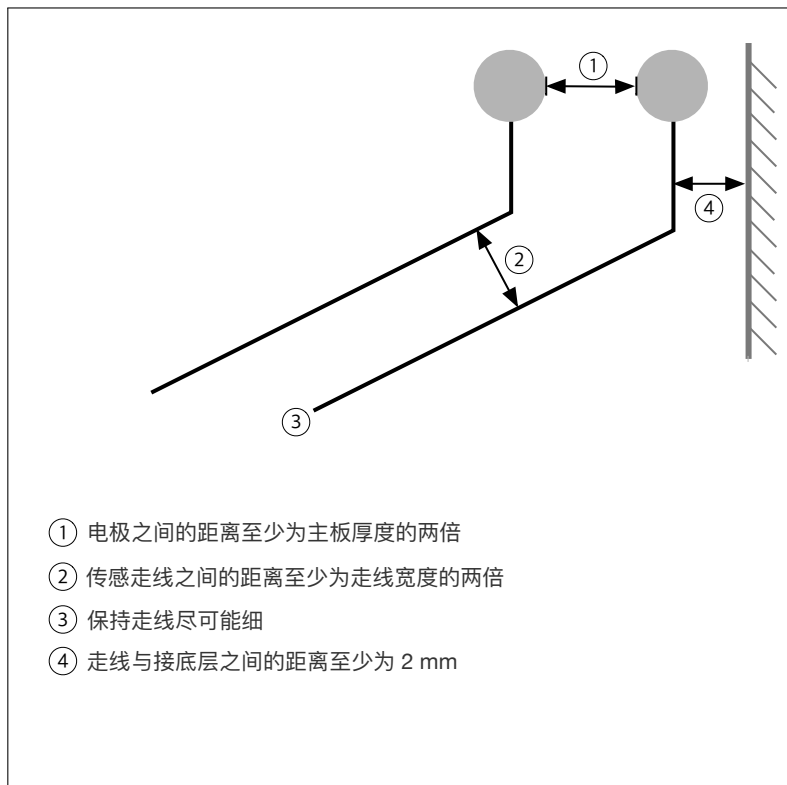


图 10: 传感器布局布线

## 附录 B - 示例代码

### B.1. 输入

```
>python esptool.py -p dev/tty8 -b 115200 write_Flash -c ESP32 -ff 40m -fm qio -fs 2MB
0x0 ~/Workspace/ESP32_BIN/boot.bin
0x04000 ~/Workspace/ESP32_BIN/drom0.bin
0x40000 ~/Workspace/ESP32_BIN/bin/irom0_Flash.bin
0xFC000 ~/Workspace/ESP32_BIN/blank.bin
0x1FC000 ~/Workspace/ESP32_BIN/esp_init_data_default.bin
```

### B.2. 输出

```
Connecting...
Erasing Flash...
Wrote 3072 bytes at 0x00000000 in 0.3 seconds (73.8 kbit/s)...
Erasing Flash...
Wrote 395264 bytes at 0x04000000 in 43.2 seconds (73.2 kbit/s)...
Erasing Flash...
Wrote 1024 bytes at 0x40000000 in 0.1 seconds (74.5 kbit/s)...
Erasing Flash...
Wrote 4096 bytes at 0xfc000000 in 0.4 seconds (73.5 kbit/s)...
Erasing Flash...
Wrote 4096 bytes at 0x1fc00000 in 0.5 seconds (73.8 kbit/s)...
Leaving...
```

## 附录 C - ESP32 管脚清单

### C.1. 管脚清单说明

表 18: 管脚清单说明

序号	说明
1	<a href="#">IO_MUX</a> 表格中，红色单元格表示 ESP32 与 ESP31B 相比在功能上的不同之处，蓝色单元格表示 ESP32 与 ESP31B 相比新增的功能，黄色单元格表示这几个 GPIO 管脚仅为输入管脚。
2	以下管脚仅为输入管脚，不带输出驱动器或内置上拉 / 下拉电路：SENSOR_VP (GPIO36)、SENSOR_CAPP (GPIO37)、SENSOR_CAPN (GPIO38)、SENSOR_VN (GPIO39)、VDET_1 (GPIO34)、VDET_2 (GPIO35)。
3	所有管脚分为 4 个电源域：VANA (模拟电源)、VRTC (RTC 电源)、VIO (数字 IO 和 CPU 电源)、VSDIO (SDIO IO 电源)。VSDIO 是内置 SDIO-LDO 的输出源。SDIO-LDO 的配置电压为 1.8V 或与 VRTC 相同。Strapping 管脚和 eFuse bit 共同决定 SDIO-LDO 的默认电压。此外，软件还可以配置寄存器来强制改变 SDIO-LDO 电压。详见 <a href="#">IO_MUX</a> 表格中 “Power Domain” 一栏。
4	VRTC 域中的功能管脚具有模拟功能，包括 32 kHz 晶体振荡器、ADC 前置放大器、ADC、DAC 以及电容式触摸传感器。请参考 <a href="#">IO_MUX</a> 表格中 “Analog Function 1~3” 各栏。
5	这些 VRTC 管脚支持 RTC Function 功能，能够在 Deep-sleep 模式下使用。例如，RTC-GPIO 可用于 Deep-sleep 模式下的芯片。
6	GPIO 管脚最多支持 <a href="#">IO_MUX</a> 表格中 “Function 1~6” 各栏中所列的 6 个数字功能。功能选择寄存器应设置为 “ <i>N</i> -1”，其中 <i>N</i> 为功能编号。以下为常用模块的信号前缀定义： <ul style="list-style-type: none"> <li>• SD_* 是 SDIO 从机的信号。</li> <li>• HS1_* 是 SDIO 主机端口 1 的信号。</li> <li>• HS2_* 是 SDIO 主机端口 2 的信号。</li> <li>• MT* 是 JTAG 的信号。</li> <li>• U0* 是 UART0 模块的信号。</li> <li>• U1* 是 UART1 模块的信号。</li> <li>• U2* 是 UART2 模块的信号。</li> <li>• SPI* 是 SPI01 模块的信号。</li> <li>• HSPI* 是 SPI02 模块的信号。</li> <li>• VSPI* 是 SPI03 模块的信号。</li> </ul>

序号	说明
7	<p>每个数字 Function 栏均对应一列 Type。请参考下列描述来理解每个 Function 所对应 Type 的意义。对于任意功能 Function-<i>N</i> 而言，Type 所指的是：</p> <ul style="list-style-type: none"> <li>• I: 仅为输入。如果选择了 Function-<i>N</i> 以外的其他功能，则该管脚的输入信号仍会连接到 Function-<i>N</i> 的输入信号。</li> <li>• I1: 仅为输入。如果选择了</li> <li>• Function-<i>N</i> 以外的其他功能，则 Function-<i>N</i> 的输入信号恒为 1。</li> <li>• IO: 仅为输入。如果选择 Function-<i>N</i> 以外的其他功能，则 Function-<i>N</i> 的输入信号恒为 0。</li> <li>• O: 仅为输出。</li> <li>• T: 高阻抗。</li> <li>• I/O/T: 该功能信号包含输入、输出和高阻抗组合。</li> <li>• I1/O/T: 该函数信号包含输入、输出和高阻抗组合。如果选择了 Function-<i>N</i> 以外的其他功能，则 Function-<i>N</i> 的输入信号恒为 1。</li> </ul> <p>例如，30 号管脚可作为 HS1_CMD 或 SD_CMD 使用，其中 HS1_CMD 的类型为 I1/O/T。如果 30 号管脚被选作 HS1_CMD，则该管脚的输入和输出均由 SDIO 主机控制。如果 30 号管脚未被选作 HS1_CMD，则对 SDIO 主机的输入信号总为 1。</p>
8	<p>每一个数字输出管脚的驱动强度可以配置。IO_MUX 表格中的“Drive Strength”栏列出了默认值。其中，驱动强度可通过寄存器配置为以下四种：</p> <ul style="list-style-type: none"> <li>• 0: ~5 mA</li> <li>• 1: ~10 mA</li> <li>• 2: ~20 mA</li> <li>• 3: ~40 mA</li> </ul> <p>默认值为 2。</p> <p>内置上拉 (wpu) 和内置下拉 (wpd) 驱动强度为 ~75 <math>\mu</math>A。</p>
9	<p>IO_MUX 表格中“At Reset”栏列出了复位时每个管脚的状态，包括输入使能 (ie=1)、内置上拉 (wpu) 和内置下拉 (wpd)。复位时，所有管脚都处于输出禁用状态。</p>
10	<p>IO_MUX 表格中“After Reset”栏列出了复位后每个管脚的即刻状态，包括输入使能 (ie=1)、内置上拉 (wpu) 和内置下拉 (wpd)。复位后，每个管脚均设定为“Function 1”。输出使能由其数字 Function 1 控制。</p>
11	<p>表格 Ethernet_MAC 用于描述 Ethernet MAC 模块内的信号映射。Ethernet MAC 支持 MII 和 RMII 两种接口，同时支持内部 PLL 时钟和外部时钟源。对于 MII 接口来说，Ethernet MAC 可配 TX_ERR 信号，也可不配 TX_ERR 信号。MDC、MDIO、CRS 和 COL 均为慢速信号，可通过 GPIO 矩阵映射到任何 GPIO 管脚。</p>
12	<p>表格 GPIO_Matrix 用于描述 GPIO 交换矩阵。片上功能模组所发出的信号可映射到任何 GPIO 管脚上。如表格 GPIO_Matrix 中，“The same input signal from IO_MUX core”栏说明的一样，有些信号既可通过 IO-MUX，也可通过 GPIO 矩阵映射到管脚上。</p>
13	<p>* 表格 GPIO_Matrix 中，“默认值”一栏表示未分配 GPIO 时，输入信号的默认值。信号的实际值由寄存器位 GPIO_FUNC<i>m</i>_IN_INV_SEL 和寄存器域 GPIO_FUNC<i>m</i>_IN_SEL 共同决定。(m 的取值范围为 1~255。)</p>

## C.2. GPIO\_Matrix

表 19: GPIO\_Matrix

序号	输入信号	默认值 *	信号可经由 IO_MUX 输出	输出信号	输出信号的 输出使能信号
0	SPICLK_in	0	yes	SPICLK_out	SPICLK_oe
1	SPIQ_in	0	yes	SPIQ_out	SPIQ_oe
2	SPID_in	0	yes	SPID_out	SPID_oe
3	SPIHD_in	0	yes	SPIHD_out	SPIHD_oe
4	SPIWP_in	0	yes	SPIWP_out	SPIWP_oe
5	SPICS0_in	0	yes	SPICS0_out	SPICS0_oe
6	SPICS1_in	0	no	SPICS1_out	SPICS1_oe
7	SPICS2_in	0	no	SPICS2_out	SPICS2_oe
8	HSPICLK_in	0	yes	HSPICLK_out	HSPICLK_oe
9	HSPIQ_in	0	yes	HSPIQ_out	HSPIQ_oe
10	HSPID_in	0	yes	HSPID_out	HSPID_oe
11	HSPICS0_in	0	yes	HSPICS0_out	HSPICS0_oe
12	HSPIHD_in	0	yes	HSPIHD_out	HSPIHD_oe
13	HSPIWP_in	0	yes	HSPIWP_out	HSPIWP_oe
14	U0RXD_in	0	yes	U0TXD_out	1'd1
15	U0CTS_in	0	yes	U0RTS_out	1'd1
16	U0DSR_in	0	no	U0DTR_out	1'd1
17	U1RXD_in	0	yes	U1TXD_out	1'd1
18	U1CTS_in	0	yes	U1RTS_out	1'd1
23	I2S0O_BCK_in	0	no	I2S0O_BCK_out	1'd1
24	I2S1O_BCK_in	0	no	I2S1O_BCK_out	1'd1
25	I2S0O_WS_in	0	no	I2S0O_WS_out	1'd1
26	I2S1O_WS_in	0	no	I2S1O_WS_out	1'd1
27	I2S0I_BCK_in	0	no	I2S0I_BCK_out	1'd1
28	I2S0I_WS_in	0	no	I2S0I_WS_out	1'd1
29	I2CEXT0_SCL_in	1	no	I2CEXT0_SCL_out	1'd1
30	I2CEXT0_SDA_in	1	no	I2CEXT0_SDA_out	1'd1
31	pwm0_sync0_in	0	no	sdio_tohost_int_out	1'd1
32	pwm0_sync1_in	0	no	pwm0_out0a	1'd1
33	pwm0_sync2_in	0	no	pwm0_out0b	1'd1
34	pwm0_f0_in	0	no	pwm0_out1a	1'd1
35	pwm0_f1_in	0	no	pwm0_out1b	1'd1
36	pwm0_f2_in	0	no	pwm0_out2a	1'd1
37	-	0	no	pwm0_out2b	1'd1
39	pcnt_sig_ch0_in0	0	no	-	1'd1
40	pcnt_sig_ch1_in0	0	no	-	1'd1
41	pcnt_ctrl_ch0_in0	0	no	-	1'd1
42	pcnt_ctrl_ch1_in0	0	no	-	1'd1
43	pcnt_sig_ch0_in1	0	no	-	1'd1
44	pcnt_sig_ch1_in1	0	no	-	1'd1

序号	输入信号	默认值 *	信号可经由 IO_MUX 输出	输出信号	输出信号的 输出使能信号
45	pcnt_ctrl_ch0_in1	0	no	-	1'd1
46	pcnt_ctrl_ch1_in1	0	no	-	1'd1
47	pcnt_sig_ch0_in2	0	no	-	1'd1
48	pcnt_sig_ch1_in2	0	no	-	1'd1
49	pcnt_ctrl_ch0_in2	0	no	-	1'd1
50	pcnt_ctrl_ch1_in2	0	no	-	1'd1
51	pcnt_sig_ch0_in3	0	no	-	1'd1
52	pcnt_sig_ch1_in3	0	no	-	1'd1
53	pcnt_ctrl_ch0_in3	0	no	-	1'd1
54	pcnt_ctrl_ch1_in3	0	no	-	1'd1
55	pcnt_sig_ch0_in4	0	no	-	1'd1
56	pcnt_sig_ch1_in4	0	no	-	1'd1
57	pcnt_ctrl_ch0_in4	0	no	-	1'd1
58	pcnt_ctrl_ch1_in4	0	no	-	1'd1
61	HSPICS1_in	0	no	HSPICS1_out	HSPICS1_oe
62	HSPICS2_in	0	no	HSPICS2_out	HSPICS2_oe
63	VSPICLK_in	0	yes	VSPICLK_out_mux	VSPICLK_oe
64	VSPIQ_in	0	yes	VSPIQ_out	VSPIQ_oe
65	VSPID_in	0	yes	VSPID_out	VSPID_oe
66	VSPiHD_in	0	yes	VSPiHD_out	VSPiHD_oe
67	VSPiWP_in	0	yes	VSPiWP_out	VSPiWP_oe
68	VSPICS0_in	0	yes	VSPICS0_out	VSPICS0_oe
69	VSPICS1_in	0	no	VSPICS1_out	VSPICS1_oe
70	VSPICS2_in	0	no	VSPICS2_out	VSPICS2_oe
71	pcnt_sig_ch0_in5	0	no	ledc_hs_sig_out0	1'd1
72	pcnt_sig_ch1_in5	0	no	ledc_hs_sig_out1	1'd1
73	pcnt_ctrl_ch0_in5	0	no	ledc_hs_sig_out2	1'd1
74	pcnt_ctrl_ch1_in5	0	no	ledc_hs_sig_out3	1'd1
75	pcnt_sig_ch0_in6	0	no	ledc_hs_sig_out4	1'd1
76	pcnt_sig_ch1_in6	0	no	ledc_hs_sig_out5	1'd1
77	pcnt_ctrl_ch0_in6	0	no	ledc_hs_sig_out6	1'd1
78	pcnt_ctrl_ch1_in6	0	no	ledc_hs_sig_out7	1'd1
79	pcnt_sig_ch0_in7	0	no	ledc_ls_sig_out0	1'd1
80	pcnt_sig_ch1_in7	0	no	ledc_ls_sig_out1	1'd1
81	pcnt_ctrl_ch0_in7	0	no	ledc_ls_sig_out2	1'd1
82	pcnt_ctrl_ch1_in7	0	no	ledc_ls_sig_out3	1'd1
83	rmt_sig_in0	0	no	ledc_ls_sig_out4	1'd1
84	rmt_sig_in1	0	no	ledc_ls_sig_out5	1'd1
85	rmt_sig_in2	0	no	ledc_ls_sig_out6	1'd1
86	rmt_sig_in3	0	no	ledc_ls_sig_out7	1'd1
87	rmt_sig_in4	0	no	rmt_sig_out0	1'd1
88	rmt_sig_in5	0	no	rmt_sig_out1	1'd1
89	rmt_sig_in6	0	no	rmt_sig_out2	1'd1



序号	输入信号	默认值 *	信号可经由 IO_MUX 输出	输出信号	输出信号的 输出使能信号
90	rmt_sig_in7	0	no	rmt_sig_out3	1'd1
91	-	-	-	rmt_sig_out4	1'd1
92	-	-	-	rmt_sig_out6	1'd1
94	-	-	-	rmt_sig_out7	1'd1
95	I2CEXT1_SCL_in	1	no	I2CEXT1_SCL_out	1'd1
96	I2CEXT1_SDA_in	1	no	I2CEXT1_SDA_out	1'd1
97	host_card_detect_n_1	0	no	host_ccmd_od_pullup_en_n	1'd1
98	host_card_detect_n_2	0	no	host_rst_n_1	1'd1
99	host_card_write_prt_1	0	no	host_rst_n_2	1'd1
100	host_card_write_prt_2	0	no	gpio_sd0_out	1'd1
101	host_card_int_n_1	0	no	gpio_sd1_out	1'd1
102	host_card_int_n_2	0	no	gpio_sd2_out	1'd1
103	pwm1_sync0_in	0	no	gpio_sd3_out	1'd1
104	pwm1_sync1_in	0	no	gpio_sd4_out	1'd1
105	pwm1_sync2_in	0	no	gpio_sd5_out	1'd1
106	pwm1_f0_in	0	no	gpio_sd6_out	1'd1
107	pwm1_f1_in	0	no	gpio_sd7_out	1'd1
108	pwm1_f2_in	0	no	pwm1_out0a	1'd1
109	pwm0_cap0_in	0	no	pwm1_out0b	1'd1
110	pwm0_cap1_in	0	no	pwm1_out1a	1'd1
111	pwm0_cap2_in	0	no	pwm1_out1b	1'd1
112	pwm1_cap0_in	0	no	pwm1_out2a	1'd1
113	pwm1_cap1_in	0	no	pwm1_out2b	1'd1
114	pwm1_cap2_in	0	no	pwm2_out1h	1'd1
115	pwm2_flta	1	no	pwm2_out1l	1'd1
116	pwm2_fltb	1	no	pwm2_out2h	1'd1
117	pwm2_cap1_in	0	no	pwm2_out2l	1'd1
118	pwm2_cap2_in	0	no	pwm2_out3h	1'd1
119	pwm2_cap3_in	0	no	pwm2_out3l	1'd1
120	pwm3_flta	1	no	pwm2_out4h	1'd1
121	pwm3_fltb	1	no	pwm2_out4l	1'd1
122	pwm3_cap1_in	0	no	-	1'd1
123	pwm3_cap2_in	0	no	-	1'd1
124	pwm3_cap3_in	0	no	-	1'd1
140	I2S0I_DATA_in0	0	no	I2S0O_DATA_out0	1'd1
141	I2S0I_DATA_in1	0	no	I2S0O_DATA_out1	1'd1
142	I2S0I_DATA_in2	0	no	I2S0O_DATA_out2	1'd1
143	I2S0I_DATA_in3	0	no	I2S0O_DATA_out3	1'd1
144	I2S0I_DATA_in4	0	no	I2S0O_DATA_out4	1'd1
145	I2S0I_DATA_in5	0	no	I2S0O_DATA_out5	1'd1
146	I2S0I_DATA_in6	0	no	I2S0O_DATA_out6	1'd1
147	I2S0I_DATA_in7	0	no	I2S0O_DATA_out7	1'd1
148	I2S0I_DATA_in8	0	no	I2S0O_DATA_out8	1'd1

序号	输入信号	默认值 *	信号可经由 IO_MUX 输出	输出信号	输出信号的 输出使能信号
149	I2S0I_DATA_in9	0	no	I2S0O_DATA_out9	1'd1
150	I2S0I_DATA_in10	0	no	I2S0O_DATA_out10	1'd1
151	I2S0I_DATA_in11	0	no	I2S0O_DATA_out11	1'd1
152	I2S0I_DATA_in12	0	no	I2S0O_DATA_out12	1'd1
153	I2S0I_DATA_in13	0	no	I2S0O_DATA_out13	1'd1
154	I2S0I_DATA_in14	0	no	I2S0O_DATA_out14	1'd1
155	I2S0I_DATA_in15	0	no	I2S0O_DATA_out15	1'd1
156	-	-	-	I2S0O_DATA_out16	1'd1
157	-	-	-	I2S0O_DATA_out17	1'd1
158	-	-	-	I2S0O_DATA_out18	1'd1
159	-	-	-	I2S0O_DATA_out19	1'd1
160	-	-	-	I2S0O_DATA_out20	1'd1
161	-	-	-	I2S0O_DATA_out21	1'd1
162	-	-	-	I2S0O_DATA_out22	1'd1
163	-	-	-	I2S0O_DATA_out23	1'd1
164	I2S1I_BCK_in	0	no	I2S1I_BCK_out	1'd1
165	I2S1I_WS_in	0	no	I2S1I_WS_out	1'd1
166	I2S1I_DATA_in0	0	no	I2S1O_DATA_out0	1'd1
167	I2S1I_DATA_in1	0	no	I2S1O_DATA_out1	1'd1
168	I2S1I_DATA_in2	0	no	I2S1O_DATA_out2	1'd1
169	I2S1I_DATA_in3	0	no	I2S1O_DATA_out3	1'd1
170	I2S1I_DATA_in4	0	no	I2S1O_DATA_out4	1'd1
171	I2S1I_DATA_in5	0	no	I2S1O_DATA_out5	1'd1
172	I2S1I_DATA_in6	0	no	I2S1O_DATA_out6	1'd1
173	I2S1I_DATA_in7	0	no	I2S1O_DATA_out7	1'd1
174	I2S1I_DATA_in8	0	no	I2S1O_DATA_out8	1'd1
175	I2S1I_DATA_in9	0	no	I2S1O_DATA_out9	1'd1
176	I2S1I_DATA_in10	0	no	I2S1O_DATA_out10	1'd1
177	I2S1I_DATA_in11	0	no	I2S1O_DATA_out11	1'd1
178	I2S1I_DATA_in12	0	no	I2S1O_DATA_out12	1'd1
179	I2S1I_DATA_in13	0	no	I2S1O_DATA_out13	1'd1
180	I2S1I_DATA_in14	0	no	I2S1O_DATA_out14	1'd1
181	I2S1I_DATA_in15	0	no	I2S1O_DATA_out15	1'd1
182	-	-	-	I2S1O_DATA_out16	1'd1
183	-	-	-	I2S1O_DATA_out17	1'd1
184	-	-	-	I2S1O_DATA_out18	1'd1
185	-	-	-	I2S1O_DATA_out19	1'd1
186	-	-	-	I2S1O_DATA_out20	1'd1
187	-	-	-	I2S1O_DATA_out21	1'd1
188	-	-	-	I2S1O_DATA_out22	1'd1
189	-	-	-	I2S1O_DATA_out23	1'd1
190	I2S0I_H_SYNC	0	no	pwm3_out1h	1'd1
191	I2S0I_V_SYNC	0	no	pwm3_out1l	1'd1

序号	输入信号	默认值 *	信号可经由 IO_MUX 输出	输出信号	输出信号的 输出使能信号
192	I2S0I_H_ENABLE	0	no	pwm3_out2h	1'd1
193	I2S1I_H_SYNC	0	no	pwm3_out2l	1'd1
194	I2S1I_V_SYNC	0	no	pwm3_out3h	1'd1
195	I2S1I_H_ENABLE	0	no	pwm3_out3l	1'd1
196	-	-	-	pwm3_out4h	1'd1
197	-	-	-	pwm3_out4l	1'd1
198	U2RXD_in	0	yes	U2TXD_out	1'd1
199	U2CTS_in	0	yes	U2RTS_out	1'd1
200	emac_mdc_i	0	no	emac_mdc_o	emac_mdc_oe
201	emac_mdio_i	0	no	emac_mdio_o	emac_mdio_o_e
202	emac_crs_i	0	no	emac_crs_o	emac_crs_oe
203	emac_col_i	0	no	emac_col_o	emac_col_oe
204	pcmfsync_in	0	no	bt_audio0_irq	1'd1
205	pcmclk_in	0	no	bt_audio1_irq	1'd1
206	pcmdin	0	no	bt_audio2_irq	1'd1
207	-	-	-	ble_audio0_irq	1'd1
208	-	-	-	ble_audio1_irq	1'd1
209	-	-	-	ble_audio2_irq	1'd1
210	-	-	-	pcmfsync_out	pcmfsync_en
211	-	-	-	pcmclk_out	pcmclk_en
212	-	-	-	pcmdout	pcmdout_en
213	-	-	-	ble_audio_sync0_p	1'd1
214	-	-	-	ble_audio_sync1_p	1'd1
215	-	-	-	ble_audio_sync2_p	1'd1
224	-	-	-	sig_in_func224	1'd1
225	-	-	-	sig_in_func225	1'd1
226	-	-	-	sig_in_func226	1'd1
227	-	-	-	sig_in_func227	1'd1
228	-	-	-	sig_in_func228	1'd1

## C.3. Ethernet\_MAC

表 20: Ethernet\_MAC

PIN Name	Function6	MII (int_osc)	MII (ext_osc)	RMII (int_osc)	RMII (ext_osc)
GPIO0	EMAC_TX_CLK	TX_CLK (I)	TX_CLK (I)	CLK_OUT(O)	EXT_OSC_CLK(I)
GPIO5	EMAC_RX_CLK	RX_CLK (I)	RX_CLK (I)	-	-
GPIO21	EMAC_TX_EN	TX_EN(O)	TX_EN(O)	TX_EN(O)	TX_EN(O)
GPIO19	EMAC_TXD0	TXD[0](O)	TXD[0](O)	TXD[0](O)	TXD[0](O)
GPIO22	EMAC_TXD1	TXD[1](O)	TXD[1](O)	TXD[1](O)	TXD[1](O)
MTMS	EMAC_TXD2	TXD[2](O)	TXD[2](O)	-	-
MTDI	EMAC_TXD3	TXD[3](O)	TXD[3](O)	-	-
MTCK	EMAC_RX_ER	RX_ER(I)	RX_ER(I)	-	-
GPIO27	EMAC_RX_DV	RX_DV(I)	RX_DV(I)	CRS_DV(I)	CRS_DV(I)
GPIO25	EMAC_RXD0	RXD[0](I)	RXD[0](I)	RXD[0](I)	RXD[0](I)
GPIO26	EMAC_RXD1	RXD[1](I)	RXD[1](I)	RXD[1](I)	RXD[1](I)
U0TXD	EMAC_RXD2	RXD[2](I)	RXD[2](I)	-	-
MTDO	EMAC_RXD3	RXD[3](I)	RXD[3](I)	-	-
GPIO16	EMAC_CLK_OUT	CLK_OUT(O)	-	CLK_OUT(O)	-
GPIO17	EMAC_CLK_OUT_180	CLK_OUT_180(O)	-	CLK_OUT_180(O)	-
GPIO4	EMAC_TX_ER	TX_ERR(O)*	TX_ERR(O)*	-	-
In GPIO Matrix*	-	MDC(O)	MDC(O)	MDC(O)	MDC(O)
In GPIO Matrix*	-	MDIO(IO)	MDIO(IO)	MDIO(IO)	MDIO(IO)
In GPIO Matrix*	-	CRS(I)	CRS(I)	-	-
In GPIO Matrix*	-	COL(I)	COL(I)	-	-

\* 说明：1. GPIO Matrix 可以是任意 GPIO。2. TX\_ERR (O) 非必选。

## C.4. IO\_MUX

IO\_MUX 管脚清单详见下一页。

IO\_MUX

Pin No.	Power Supply Pin	Analog Pin	Digital Pin	Power Domain	Analog Function1	Analog Function2	Analog Function3	RTC Function1	RTC Function2	Function1	Type	Function2	Type	Function3	Type	Function4	Type	Function5	Type	Function6	Type	Drive Strength (2'd2: 20 mA)	At Reset	After Reset	
1	VDDA	LNA_IN  SENSOR_VP SENSOR_CAPP SENSOR_CAPN SENSOR_VN CHIP_PU VDET_1 VDET_2 32K_XP		VANA in																					
2				VANA in																					
3	VDD3P3			VANA in																					
4	VDD3P3			VANA in																					
5					VRTC	ADC_H	ADC1_CH0		RTC_GPIO0		GPIO36	I			GPIO36	I									ie=0
6					VRTC	ADC_H	ADC1_CH1		RTC_GPIO1		GPIO37	I			GPIO37	I									ie=0
7					VRTC	ADC_H	ADC1_CH2		RTC_GPIO2		GPIO38	I			GPIO38	I									ie=0
8					VRTC	ADC_H	ADC1_CH3		RTC_GPIO3		GPIO39	I			GPIO39	I									ie=0
9					VRTC																				
10					VRTC		ADC1_CH6		RTC_GPIO4		GPIO34	I			GPIO34	I									ie=0
11					VRTC		ADC1_CH7		RTC_GPIO5		GPIO35	I			GPIO35	I									ie=0
12					VRTC	XTAL_32K_P	ADC1_CH4	TOUCH9	RTC_GPIO9		GPIO32	I/O/T			GPIO32	I/O/T							2'd2		ie=0
13		32K_XN		VRTC	XTAL_32K_N	ADC1_CH5	TOUCH8	RTC_GPIO8		GPIO33	I/O/T			GPIO33	I/O/T							2'd2		ie=0	
14			GPIO25	VRTC	DAC_1	ADC2_CH8		RTC_GPIO6		GPIO25	I/O/T			GPIO25	I/O/T					EMAC_RXD0	I	2'd2		ie=0	
15			GPIO26	VRTC	DAC_2	ADC2_CH9		RTC_GPIO7		GPIO26	I/O/T			GPIO26	I/O/T					EMAC_RXD1	I	2'd2		ie=0	
16			GPIO27	VRTC		ADC2_CH7	TOUCH7	RTC_GPIO17		GPIO27	I/O/T			GPIO27	I/O/T					EMAC_RX_DV	I	2'd2		ie=1	
17			MTMS	VRTC		ADC2_CH6	TOUCH6	RTC_GPIO16		MTMS	I0	HSPICLK	I/O/T	GPIO14	I/O/T	HS2_CLK	O	SD_CLK	I0	EMAC_TXD2	O	2'd2	wpu, ie=1	wpu, ie=1	
18			MTDI	VRTC		ADC2_CH5	TOUCH5	RTC_GPIO15		MTDI	I1	HSPIQ	I/O/T	GPIO12	I/O/T	HS2_DATA2	I1/O/T	SD_DATA2	I1/O/T	EMAC_TXD3	O	2'd2	wpd, ie=1	wpd, ie=1	
19	VDD3P3_RTC			VRTC supply in																					
20			MTCK	VRTC		ADC2_CH4	TOUCH4	RTC_GPIO14		MTCK	I1	HSPID	I/O/T	GPIO13	I/O/T	HS2_DATA3	I1/O/T	SD_DATA3	I1/O/T	EMAC_RX_ER	I	2'd2	wpu, ie=1	wpu, ie=1	
21			MTDO	VRTC		ADC2_CH3	TOUCH3	RTC_GPIO13	I2C_SDA	MTDO	O/T	HSPIC80	I/O/T	GPIO15	I/O/T	HS2_CMD	I1/O/T	SD_CMD	I1/O/T	EMAC_RXD3	I	2'd2	wpu, ie=1	wpu, ie=1	
22			GPIO2	VRTC		ADC2_CH2	TOUCH2	RTC_GPIO12	I2C_SCL	GPIO2	I/O/T	HSPIWP	I/O/T	GPIO2	I/O/T	HS2_DATA0	I1/O/T	SD_DATA0	I1/O/T			2'd2	wpd, ie=1	wpd, ie=1	
23			GPIO0	VRTC		ADC2_CH1	TOUCH1	RTC_GPIO11	I2C_SDA	GPIO0	I/O/T	CLK_OUT1	O	GPIO0	I/O/T					EMAC_TX_CLK	I	2'd2	wpu, ie=1	wpu, ie=1	
24			GPIO4	VRTC		ADC2_CH0	TOUCH0	RTC_GPIO10	I2C_SCL	GPIO4	I/O/T	HSPHID	I/O/T	GPIO4	I/O/T	HS2_DATA1	I1/O/T	SD_DATA1	I1/O/T	EMAC_TX_ER	O	2'd2	wpd, ie=1	wpd, ie=1	
25			GPIO16	VSDIO						GPIO16	I/O/T			GPIO16	I/O/T	HS1_DATA4	I1/O/T	U2RXD	I1	EMAC_CLK_OUT	O	2'd2		ie=1	
26	VDD_SDIO			VSDIO supply out/in																					
27			GPIO17	VSDIO						GPIO17	I/O/T			GPIO17	I/O/T	HS1_DATA5	I1/O/T	U2TXD	O	EMAC_CLK_OUT_180	O	2'd2		ie=1	
28		SD_DATA_2		VSDIO						SD_DATA2	I1/O/T	SPHID	I/O/T	GPIO9	I/O/T	HS1_DATA2	I1/O/T	U1RXD	I1			2'd2	wpu, ie=1	wpu, ie=1	
29		SD_DATA_3		VSDIO						SD_DATA3	I0/O/T	SPIWP	I/O/T	GPIO10	I/O/T	HS1_DATA3	I1/O/T	U1TXD	O			2'd2	wpu, ie=1	wpu, ie=1	
30		SD_CMD		VSDIO						SD_CMD	I1/O/T	SPIC80	I/O/T	GPIO11	I/O/T	HS1_CMD	I1/O/T	U1RTS	O			2'd2	wpu, ie=1	wpu, ie=1	
31		SD_CLK		VSDIO						SD_CLK	I0	SPICLK	I/O/T	GPIO6	I/O/T	HS1_CLK	O	U1CTS	I1			2'd2	wpu, ie=1	wpu, ie=1	
32		SD_DATA_0		VSDIO						SD_DATA0	I1/O/T	SPIQ	I/O/T	GPIO7	I/O/T	HS1_DATA0	I1/O/T	U2RTS	O			2'd2	wpu, ie=1	wpu, ie=1	
33		SD_DATA_1		VSDIO						SD_DATA1	I1/O/T	SPID	I/O/T	GPIO8	I/O/T	HS1_DATA1	I1/O/T	U2CTS	I1			2'd2	wpu, ie=1	wpu, ie=1	
34			GPIO5	VIO						GPIO5	I/O/T	VSPIC80	I/O/T	GPIO5	I/O/T	HS1_DATA6	I1/O/T			EMAC_RX_CLK	I	2'd2	wpu, ie=1	wpu, ie=1	
35			GPIO18	VIO						GPIO18	I/O/T	VSPICLK	I/O/T	GPIO18	I/O/T	HS1_DATA7	I1/O/T					2'd2		ie=1	
36			GPIO23	VIO						GPIO23	I/O/T	VSPID	I/O/T	GPIO23	I/O/T	HS1_STROBE	I0					2'd2		ie=1	
37	VDD3P3_CPU			VIO supply in																					
38			GPIO19	VIO						GPIO19	I/O/T	VSPIQ	I/O/T	GPIO19	I/O/T	UOCTS	I1			EMAC_TXD0	O	2'd2		ie=1	
39			GPIO22	VIO						GPIO22	I/O/T	VSPWP	I/O/T	GPIO22	I/O/T	UORTS	O			EMAC_TXD1	O	2'd2		ie=1	
40		UORXD		VIO						UORXD	I1	CLK_OUT2	O	GPIO3	I/O/T							2'd2	wpu, ie=1	wpu, ie=1	
41		UOTXD		VIO						UOTXD	O	CLK_OUT3	O	GPIO1	I/O/T					EMAC_RXD2	I	2'd2	wpu, ie=1	wpu, ie=1	
42			GPIO21	VIO						GPIO21	I/O/T	VSPHID	I/O/T	GPIO21	I/O/T					EMAC_TX_EN	O	2'd2		ie=1	
43	VDDA			VANA in																					
44		XTAL_N		VANA																					
45		XTAL_P		VANA																					
46	VDDA			VANA																					
47		CAP2		VANA																					
48		CAP1		VANA																					
Total Number	8	14	26																						
Note: Please see Table: Notes on ESP32 Pin Lists. (请参考表：管脚清单说明。)																									