

ESP32

常见问题



版本 1.0
乐鑫信息科技
版权所有 © 2018

关于本手册

本文介绍 ESP32 的常见问题以及解答。

发布说明

日期	版本	发布说明
2018.08	V1.0	首次发布。

文档变更通知

用户可通过乐鑫官网订阅页面 <https://www.espressif.com/zh-hans/subscribe> 订阅技术文档变更的电子邮件通知。

证书下载

用户可通过乐鑫官网证书下载页面 <https://www.espressif.com/zh-hans/certificates> 下载产品证书。

目录

1. RF	1
1.1. 射频匹配和物料	1
1.1.1. 客户自研产品如何优化二次谐波等杂散?	1
1.2. Wi-Fi 性能.....	1
1.2.1. 乐鑫芯片支持的 Wi-Fi 协议以及性能指标有哪些?	1
1.2.2. 80 MHz 倍频杂散较差该如何解决?	2
2. 应用	3
2.1. 函数和 API.....	3
2.1.1. 怎样查看线程使用过的最大栈大小?	3
3. 音频	4
3.1. 内存系统和解码	4
3.1.1. 如何使用 PSRAM, 相关文档或者实例从哪里获得?	4
3.1.2. 如何获得剩余 PSRAM 和 RAM?	4
3.2. 外设	5
3.2.1. ESP32 只有一个 CHIP_PU 管脚, 没有复位管脚, 该芯片上的 CHIP_PU 管脚可实现相同的复位功能吗? 还是只作为低功耗的硬件控制接口? 如果没有复位信号, 是否意味可通过看门狗或其他方式保证永不死机?	5
3.2.2. 乐鑫提供的参考设计中 I2S 信号分布太散, 是否可以配置集中一些, 比如配置到: GPIO5、GPIO18、GPIO23、GPIO19、GPIO22 管脚上; I2C 配置到 GPIO25、GPIO26 或 GPIO32、GPIO33 管脚上?	5
3.2.3. 脉冲宽度调制 (PWM) 信号输出是否可以分配到除了 flash、SD、I2S、I2C、UART 之外的任意 I/O 上?	5
4. AT	6
4.1. ESP32 AT 相关资源从哪里获得?	6
4.2. AT 提示 “busy” 是什么原因?	6
4.3. 为什么 BLE 开始广播后有些手机扫描不到?	6

5. 蓝牙	7
5.1. 经典蓝牙.....	7
5.1.1. ESP32 的经典蓝牙支持哪些配置?	7
5.1.2. 如何使用 ESP32 蓝牙连接手机播放音乐?	7
5.1.3. ESP32 的 SPP 性能如何?	7
5.1.4. ESP32 的经典蓝牙连接兼容性如何?	7
5.1.5. ESP32 的经典蓝牙工作电流是多少?	7
5.2. 低功耗蓝牙	8
5.2.1. ESP32 支持哪些 BLE Profile?	8
5.2.2. ESP32 的 BLE 吞吐量是多少?	8
5.2.3. ESP32 的 BLE 蓝牙配网兼容性、性能如何? 是否开源?	8
5.2.4. ESP32 的 BLE 工作电流是多少?	8
5.3. 共存	9
5.3.1. ESP32 的蓝牙双模如何共存及使用?	9
5.3.2. ESP32 的蓝牙与 Wi-Fi 如何共存?	9
5.3.3. ESP32 的蓝牙内存占用是多少?	9
6. Wi-Fi and LwIP	11
6.1. 性能	11
6.1.1. 如何测试 Wi-Fi 性能?	11
6.1.2. ESP32 Wi-Fi 吞吐量?	11
6.2. 内存	11
6.2.1. Wi-Fi 启动后会占用多少内存?	11
6.2.2. 休眠方式有哪几种及其区别是什么?	11
6.3. Scan	12
6.3.1. 内存不够时如何进行 AP 扫描?	12
6.3.2. ESP32 扫描一次需要花多长时间?	12
6.4. RX/TX	13
6.4.1. 如何发送 RAW 802.11 报文?	13

6.5. LwIP	13
6.5.1. 关闭 TCP 连接后相关的资源最长多久可以释放完毕?	13
6.5.2. LwIP 最多能够创建多少个 Socket?	13
7. 数字.....	14
7.1. RTC	14
7.1.1. 发生欠压复位 (Brownout Reset) 的原因是什么?	14
7.2. 蓝牙	14
7.2.1. 蓝牙的发射功率是多少?	14
8. 外设.....	15
8.1. 引脚	15
8.1.1. 使用哪些引脚时需要注意?	15
8.2. 触摸	15
8.2.1. 在使用 ESP32 做触摸相关应用时读取的数据很不稳定, 请问有什么参考资料从哪里获取?	15
8.3. 调试	15
8.3.1. 如何关闭默认通过 UART0 发送的调试消息?	15



1.

RF

1.1. 射频匹配和物料

1.1.1. 客户自研产品如何优化二次谐波等杂散？

二次谐波主要来源于射频链路辐射和 PA 电源辐射，同时易受到客户底板（板子尺寸）及产品整机影响，因此有如下建议：

- 在射频匹配中使用一个接近 2.4 pF 大小的对地电容，可较好地优化射频链路上的杂散辐射；
- 在 PA 电源（芯片 3、4 管脚）入口增加一个串联电感可较好减少 PA 电源的杂散辐射。

1.2. Wi-Fi 性能

1.2.1. 乐鑫芯片支持的 Wi-Fi 协议以及性能指标有哪些？

Standard/Speed (bps)	Standard TX EVM (db)	ESP32 TX Power (dbm)	ESP32 TX EVM (db)	ESP32 RX Sensitivity (dbm)
802.11b 1M	-10	19.5 ± 2 dB	-17	-98
802.11b 2M	-10	19.5 ± 2 dB	-17	-96
802.11b 5.5M	-10	19.5 ± 2 dB	-17	-94
802.11b 11M	-10	19.5 ± 2 dB	-17	-91
802.11g 6M	-5	18 ± 2 dB	-19	-93
802.11g 9M	-8	18 ± 2 dB	-19	-92
802.11g 12M	-10	18 ± 2 dB	-19	-90
802.11g 18M	-13	18 ± 2 dB	-20	-88
802.11g 24M	-16	16.5 ± 2 dB	-20	-85
802.11g 36M	-19	16.5 ± 2 dB	-24	-82
802.11g 48M	-22	15 ± 2 dB	-26	-78
802.11g 54M	-25	14 ± 2 dB	-27	-74
802.11n HT20 MCS0/6.5M/7.2M	-5	18 ± 2 dB	-19	-90



802.11n HT20 MCS1/13M/14.4M	-10	18 ± 2 dB	-19	-90
802.11n HT20 MCS2/19.5M/21.7M	-13	18 ± 2 dB	-21	-87
802.11n HT20 MCS3/26M/28.9M	-16	16.5 ± 2 dB	-20	-84
802.11n HT20 MCS4/39M/43M	-19	16.5 ± 2 dB	-23	-81
802.11n HT20 MCS5/52M/57.8M	-22	15 ± 2 dB	-26	-77
802.11n HT20 MCS6/58.5M/65M	-25	14 ± 2 dB	-27	-75
802.11n HT20 MCS7/65M/72.2M	-27	13 ± 2 dB	-29	-71
802.11n HT40 MCS0/6.5M/7.2M	-5	18 ± 2 dB	-19	-89
802.11n HT40 MCS1/13M/14.4M	-10	18 ± 2 dB	-19	-87
802.11n HT40 MCS2/19.5M/21.7M	-13	18 ± 2 dB	-21	-84
802.11n HT40 MCS3/26M/28.9M	-16	16.5 ± 2 dB	-20	-82
802.11n HT40 MCS4/39M/43M	-19	16.5 ± 2 dB	-23	-79
802.11n HT40 MCS5/52M/57.8M	-22	15 ± 2 dB	-26	-75
802.11n HT40 MCS6/58.5M/65M	-25	14 ± 2 dB	-27	-73
802.11n HT40 MCS7/65M/72.2M	-27	13 ± 2 dB	-29	-69

除此之外，目前乐鑫 ESP32 系列芯片还支持 BT 和 BLE（低功耗蓝牙）。

1.2.2. 80 MHz 倍频杂散较差该如何解决？

若 80 MHz 倍频杂散超标，如 160 MHz、240 MHz、320 MHz 等均比较高，可在发送数据 (TXD) 串口线路中串联一个阻值约为 470 Ω 的电阻，即可有效抑制 80 MHz 倍频杂散。



2.

应用

2.1. 函数和 API

2.1.1. 怎样查看线程使用过的最大栈大小?

请使用 `UBaseType_t uxTaskGetStackHighWaterMark(TaskHandle_t xTask)` 函数，任务的堆栈空间会随着任务执行以及中断处理而增长或缩小。该函数可以返回任务启动后的最小剩余堆栈空间。换句话说，可以间接估算出一个任务最多需要多少堆栈空间。在文件 *FreeRTOSConfig.h* 中，宏 `INCLUDE_uxTaskGetStackHighWaterMark` 必须设置成 1，此函数才有效。注意，该选项默认有效。详情见 <https://www.freertos.org/uxTaskGetStackHighWaterMark.html>。



3.

音频

3.1. 内存系统和解码

3.1.1. 如何使用 PSRAM，相关文档或者实例从哪里获得？

为了使用 RSPRM，可前往 [make menuconfig -> Component config -> ESP32-specific](#) 进行设置：

- ESP-IDF V2.1：请选择 [Capability allocator can allocate SPI RAM memory](#)；
- ESP-IDF V3.0 及之后版本，请选择 [Support for external, SPI-connected RAM](#)。

使能 PSRAM 后，通过 [malloc](#) 使能 [malloc\(\) can also allocate in SPI SRAM](#) 动态分配 PSRAM。在允许的情况下，将使用 PSRAM 自动进行动态内存分布。

此外，可以使用 [Always put malloc\(\)s smaller than this size, in bytes, in internal RAM](#) 设置阈值。

- 如果 [malloc](#) 动态内存分配大于设定值，则使用 PSRAM。否则，使用 RAM。
- 如果剩余 RAM 不足，则使用 PSRAM。

3.1.2. 如何获得剩余 PSRAM 和 RAM？

版本	类型	子类型	API
ESP-IDF V3.0 及以后版本	有 PSRAM	所有剩余内存（包括 PSRAM 和内部 RAM）	<code>esp_get_free_heap_size()</code>
		剩余 RAM 内存	<code>heap_caps_get_free_size(MALLOC_CAP_INTERNAL)</code>
		剩余 DRAM 内存	<code>heap_caps_get_free_size(MALLOC_CAP_INTERNAL MALLOC_CAP_8BIT)</code>
	无 PSRAM	剩余 RAM 内存	<code>heap_caps_get_free_size(MALLOC_CAP_INTERNAL)</code>
		剩余 DRAM 内存	<code>esp_get_free_heap_size()</code>
		所有剩余内存（包括 PSRAM 和内部 RAM）	<code>xPortGetFreeHeapSizeCaps(MALLOC_CAP_SPIRAM)</code>
ESP-IDF V2.1	有 PSRAM	剩余 RAM 内存	<code>xPortGetFreeHeapSizeCaps(MALLOC_CAP_INTERNAL)</code>
		剩余 DRAM 内存	<code>xPortGetFreeHeapSizeCaps(MALLOC_CAP_INTERNAL MALLOC_CAP_8BIT)</code>
		所有剩余内存（包括 PSRAM 和内部 RAM）	<code>xPortGetFreeHeapSizeCaps(MALLOC_CAP_SPIRAM)</code>
	无 PSRAM	剩余 RAM 内存	<code>esp_get_free_heap_size()</code>



3.2. 外设

- 3.2.1. ESP32 只有一个 CHIP_PU 管脚，没有复位管脚，该芯片上的 CHIP_PU 管脚可实现相同的复位功能吗？还是只作为低功耗的硬件控制接口？如果没有复位信号，是否意味可通过看门狗或其他方式保证永不死机？

CHIP_PU 即复位管脚。

- 3.2.2. 乐鑫提供的参考设计中 I2S 信号分布太散，是否可以配置集中一些，比如配置到：GPIO5、GPIO18、GPIO23、GPIO19、GPIO22 管脚上；I2C 配置到 GPIO25、GPIO26 或 GPIO32、GPIO33 管脚上？

所有 I2S 的 I/O 均可任意分配，需要注意有的 I/O 只能作为输入，请参考 [《ESP32 技术规格书》](#) 最后一页。

- 3.2.3. 脉冲宽度调制 (PWM) 信号输出是否可以分配到除了 flash、SD、I2S、I2C、UART 之外的任意 I/O 上？

PWM 可输入到任何管脚，除了只有输入功能的 I/O 之外。



4.

AT

4.1. ESP32 AT 相关资源从哪里获得?

- ESP32 AT bin 文件: <https://www.espressif.com/zh-hans/support/download/at>。
- ESP32 AT 文档: [中文版](#) | [English](#)

此外, 客户也可以基于乐鑫的 AT 核自行开发更多的 AT 指令, ESP32 AT 工程可以在 GitHub 下载: <https://github.com/espressif/esp32-at>。

4.2. AT 提示 “busy” 是什么原因?

AT 指令的处理是线性的, 也就是处理完前一条指令后, 才能接收下一条指令进行处理。提示 “busy” 表示正在处理前一条指令, 无法响应当前输入。

而任何串口的输入, 均被认为是指令输入, 因此, 当有多余的不可见字符输入时, 系统也会提示 “busy” 或者 “ERROR”。

例如, 串口输入 **AT+GMR (换行符 CR LF) (空格符)**, 由于 **AT+GMR (换行符 CR LF)** 已经是一条完整的 AT 指令了, 系统会执行该指令。

如果系统尚未完成 **AT+GMR** 操作, 就收到了后面的**空格符**, 将被认为是新的指令输入, 系统提示 “busy”。如果系统已经完成了 **AT+GMR** 操作, 再收到后面的空格符, 空格符将被认为是一条错误的指令, 系统提示 “ERROR”。

4.3. 为什么 BLE 开始广播后有些手机扫描不到?

1. 请先确认手机是否支持 BLE 功能。
2. 有的手机 (例如, 苹果手机) 在“设置”->“蓝牙”中只显示默认的经典蓝牙, BLE 广播会被手机过滤掉。建议使用专门的 BLE 应用来调试 BLE 功能。例如, 苹果手机可以使用 LightBlue 应用。
3. 蓝牙指定对 BLE 广播包的格式, 要求符合规范, 手机一般会对不符合格式的广播包进行过滤, 只有格式正确的才能被显示出来。例如, 上方问题中的示例就是一个符合规范的 BLE 广播数据包。具体请参考文档 CSS v7。



5.

蓝牙

5.1. 经典蓝牙

5.1.1. ESP32 的经典蓝牙支持哪些配置？

- ESP-IDF V3.1: HFP Client (not HF gateway)
- ESP-IDF V3.0: A2DP Source/A2DP Sink/AVRCP/AVDTP/SPP/RFCOMM

5.1.2. 如何使用 ESP32 蓝牙连接手机播放音乐？

用手机通过蓝牙播放音乐，ESP32 用作 A2DP Sink, [A2DP Sink Demo](#) 只是通过手机获取 SBC 编码的数据流，若要播放出声音，需要做编解码转换及编解码器、数/模转换器、扬声器等模块以最终输出声音。

5.1.3. ESP32 的 SPP 性能如何？

使用两块 ESP32 板子对跑 SPP，单向吞吐量可达 1900 Kbps，约 235 KB/s，已接近规范里的理论值。

5.1.4. ESP32 的经典蓝牙连接兼容性如何？

目前已经测试了如苹果、华为、小米、OPPO、魅族、一加、中兴、360 等常见品牌的大部分主流型号的手机，连接兼容性几乎全部通过，只有极个别版本的系统有差别。

5.1.5. ESP32 的经典蓝牙工作电流是多少？

A2DP (CPU 160 Mhz, DFS = false, commit a7a90f)			
Current	MAX (mA)	Min (mA)	Average (mA)
Scanning	106.4	30.8	37.8
Sniff	107.6	31.1	32.2
Play Music	123	90.1	100.4



5.2. 低功耗蓝牙

5.2.1. ESP32 支持哪些 BLE Profile?

目前支持完整的 GATT/SMP 等基础模块，可自行实现自定义配置；已经实现的配置有 BLE HID（设备端）、电池、DIS、Blu-Fi（蓝牙配网）等。

5.2.2. ESP32 的 BLE 吞吐量是多少?

ESP32 的 BLE 吞吐率取决于各种因素，例如环境干扰、连接间隔、MTU 大小以及对端设备性能等等。具体可以参考 IDF 中的 ble_throughput example，ESP32 板子之间的 BLE 通信最大吞吐量可达 700 Kbps，约 90 KB/s。

5.2.3. ESP32 的 BLE 蓝牙配网兼容性、性能如何？是否开源？

ESP32 的蓝牙配网，简称 Blu-Fi。

Blu-Fi 配网兼容性与 BLE 兼容性一致，测试过苹果、华为、小米、OPPO、魅族、一加、中兴等主流品牌手机，兼容性良好，只有个别型号需要降低 MTU 参数来使用（略微增加了连接时间）。

Blu-Fi 配网过程在 1 s~2 s 内就可完成。目前 Blu-Fi 配网支持诸多特性，如 WPA2 企业级证书传输、连接状态汇报、加密方式任意选择等，目前 Blu-Fi 协议及手机应用部分的代码不开源，但后续有可能开源。

5.2.4. ESP32 的 BLE 工作电流是多少?

Under CPU 160 Mhz, DFS = false; ESP-IDF V3.1			
Current	MAX (mA)	Min (mA)	Average (mA)
Advertising:			
Adv Interval = 40 ms	142.1	32	42.67
Scanning:			
Scan Interval = 160 ms, Window = 20 ms	142.1	32	44.4
Connection(Slave):			
Connection Interval = 20 ms, latency = 0	142.1	32	42.75
Connection(Slave):			
Connection Interval = 80 ms, latency = 0	142.1	32	35.33



5.3. 共存

5.3.1. ESP32 的蓝牙双模如何共存及使用?

ESP32 支持的双模蓝牙并没有特殊的地方，不需要做复杂的配置或调用即可使用。从开发者的角度来看，BLE 调用 BLE 的 API，经典蓝牙调用经典蓝牙的 API。经典蓝牙与 BLE 共存说明可参考文档 [ESP32 BT&BLE 双模蓝牙共存说明](#)。

5.3.2. ESP32 的蓝牙与 Wi-Fi 如何共存?

在 `menuconfig` 中，有个特殊选项 “**Software controls WiFi/Bluetooth coexistence**”，用于通过软件来控制 ESP32 的蓝牙和 Wi-Fi 共存，可以平衡 Wi-Fi、蓝牙控制 RF 的共存需求。请注意，如果使能 **Software controls WiFi/Bluetooth coexistence** 选项，BLE scan 间隔不应超过 0x100 slots（约 160 ms）。

- 若只是 BLE 与 Wi-Fi 共存，则开启这个选项和不开启均可正常使用。但不开启的时候需要注意 “BLE scan interval - BLE scan window > 150 ms”，并且 BLE scan interval 尽量小于 500 ms。
- 若经典蓝牙与 Wi-Fi 共存，则建议开启这个选项。

在 V3.0 及以前版本，可能存在经典蓝牙与 Wi-Fi 共存性能较差的问题，目前 ESP32 已经解决了这个问题，在最新 ESP-IDF master branch 上可以成功实现在 ESP32 作为蓝牙音箱流畅播放音乐的同时使用 Wi-Fi 功能。

5.3.3. ESP32 的蓝牙内存占用是多少?

1. 控制器:

- BLE 单模: 40 KB (.data + .bss + 硬件)
- BR/EDR 单模: 65 KB (.data + .bss + 硬件)
- 双模: 70 KB (.data + .bss + 硬件)

2. 主设备:

- BLE:
 - GATT Client (Gatt Client 演示) : 24 KB (.bss+.data) + 23 KB (heap) = 47 KB
 - GATT Server (GATT Server 演示) : 23 KB (.bss+.data) + 23 KB (heap) = 46 KB
 - GATT Client & GATT Server: 24 KB (.bss+.data) + 24 KB (heap) = 48 KB
 - SMP: 5 KB



- 经典蓝牙 (经典蓝牙 A2DP_SINK 演示, 包含 SMP/SDP/A2DP/AVRCP) : 48 KB
(.bss+.data) + 24 KB (heap) = 72 KB (示例运行时额外增加 13 KB)

以上堆 (Heap) 均包含任务栈 (Task Stack), 因为任务栈是从堆里分配出来的, 算为堆。

3. 优化 PSRAM 版本:

在 ESP-IDF V3.0 及以后, 打开 menuconfig 里蓝牙菜单的 PSRAM 相关选项, 将 Bluedroid (Host) 的部分 .bss/.data 段及堆放入 PSRAM, 可额外省出近 50KB。



6. Wi-Fi and LwIP

6.1. 性能

6.1.1. 如何测试 Wi-Fi 性能?

请使用 `example/wifi/liperf` 目录下代码进行测试。

6.1.2. ESP32 Wi-Fi 吞吐量?

请见 [ESP32's Wi-Fi throughput](#)。

6.2. 内存

6.2.1. Wi-Fi 启动后会占用多少内存?

Wi-Fi 启动申请的内存主要包含以下几个方面：

1. Wi-Fi 任务和队列：6.144 KB
2. Wi-Fi AMPDU：2.092 KB
3. Wi-Fi 电源管理任务和队列：3.44 KB
4. Wi-Fi 内部数据结构：1.348 KB
5. 事件任务和队列：6.164 KB
6. LWIP 任务和邮箱：3.868 KB
7. Wi-Fi/LwIP 静态 RX/TX 缓存取决于 `menuconfig` 的配置
8. Wi-Fi/LwIP 动态 RX/TX 缓存取决于 `menuconfig` 的配置

1~6 项在初始化时分配，总共占用：23.056 KB。

7~8 项取决于 `menuconfig` 配置以及收发包情况，占用内存会动态变化。

说明：

- IDF V3.0 及之后的版本优化了第 2 项，节省 2.092 KB 内存；
- IDF V3.1 及之后的版本优化了第 3 项，节省 3.44 KB 内存。

6.2.2. 休眠方式有哪几种及其区别是什么?

有 Modem-sleep、Light-sleep 和 Deep-sleep 三种休眠方式。



- Modem-sleep:
 - Wi-Fi 协议中规定的 Station Legacy Fast 休眠方式，Station 发送 NULL 数据帧通知 AP 休眠或唤醒；
 - Station 连接上 AP 之后自动开启，进入休眠状态后关闭射频模块，休眠期间保持和 AP 的连接，AP 断开连接后 Modem-sleep 不工作；
 - ESP32 Modem-sleep 进入休眠状态后，还可以选择降低 CPU 时钟频率，进一步降低电流；
- Light-sleep:
 - 基于 Modem-sleep 的 Station 休眠方式；
 - 与 Modem-sleep 的不同之处在于：
 - ▶ 进入休眠状态后，不仅关闭 RF 模块，还暂停 CPU 和部分系统时钟；
 - ▶ 退出休眠状态后，CPU 继续运行。
- Deep-sleep:
 - 非 Wi-Fi 协议规定的休眠方式；
 - 进入休眠状态后，关闭除 RTC 模块外的所有其他模块；
 - 退出休眠状态后，整个系统重新运行（类似于系统重启）；
 - 休眠期间不保持到 AP 的连接。

6.3. Scan

6.3.1. 内存不够时如何进行 AP 扫描？

存放扫描结果的内存是动态申请的，因此当周边存在大量的 AP 时会占用很多内存，内存不够时，一个缓解办法是逐信道扫描。

6.3.2. ESP32 扫描一次需要花多长时间？

扫描花费的总时间取决于：

- 是被动扫描还是主动扫描，默认为主动扫描。
- 每个信道停留的时间，默认主动扫描为 120 ms，被动扫描为 360 ms。
- 国家码与配置的信道范围，默认为 1~13 信道。
- 是快速扫描还是全信道扫描，默认为快速扫描。



- Station 模式还是 Station-AP 模式，当前是否有连接。

默认情况下，1~11 信道为主动扫描，12~13 信道为被动扫描。

- 在 Station 模式没有连接的情况下，全信道扫描总时间为： $11 * 120 + 2 * 360 = 2040$ ms；
- 在 Station 模式有连接，或者 Station-AP 模式下，全信道扫描总时间为： $11 * 120 + 2 * 360 + 13 * 30 = 2430$ ms。

6.4. RX/TX

6.4.1. 如何发送 RAW 802.11 报文？

可调用 `esp_wifi_80211_tx` 发送（IDF v3.1 可用）。

6.5. LwIP

6.5.1. 关闭 TCP 连接后相关的资源最长多久可以释放完毕？

20s 或者发送的 `linger/send_timeout` 超时之后释放资源。

6.5.2. LwIP 最多能够创建多少个 Socket？

最多 32 个，默认为 10 个。



7.

数字

7.1. RTC

7.1.1. 发生欠压复位 (Brownout Reset) 的原因是什么?

1. 检查电源是否有问题，低于低电压阈值。
2. 由于芯片工作时候电流波动比较大，请确认电源的驱动能力是否足够。此外，如果用户使用 USB 为板子供电，应保证 USB 的质量足以提供足够电流。参考 [《ESP32 技术规格书》](#)。

7.2. 蓝牙

7.2.1. 蓝牙的发射功率是多少?

ESP32 蓝牙的发射功率有 9 档，对应功率 -12 ~ 12 dBm，间隔 3 dBm 一档。控制器软件对发射功率进行限制，根据产品声明的对应功率等级选取档位。



8.

外设

8.1. 引脚

8.1.1. 使用哪些引脚时需要注意？

IO6 - IO11 为 Flash 引脚，作为 flash 通信使用，无法用于 GPIO。

如果是 WROVER 模块，GPIO16 和 GPIO17 会被系统占用，无法用于 GPIO。

此外，ESP32 有 5 个 strapping 引脚，在使用时也因应该注意，请参考 [《ESP32 技术规格书》](#)。

8.2. 触摸

8.2.1. 在使用 ESP32 做触摸相关应用时读取的数据很不稳定，请问有什么参考资料从哪里获取？

请参考推荐的软硬件设计：https://github.com/espressif/esp-iot-solution/tree/master/examples/touch_pad_evb。

8.3. 调试

8.3.1. 如何关闭默认通过 UART0 发送的调试消息？

- Bootloader 信息：GPIO15 接地；
- IDF 信息：可前往 menuconfig / Component config /Log output 进行相关配置。



乐鑫 IoT 团队

www.espressif.com

免责声明和版权公告

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。

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 © 2018 乐鑫所有。保留所有权利。