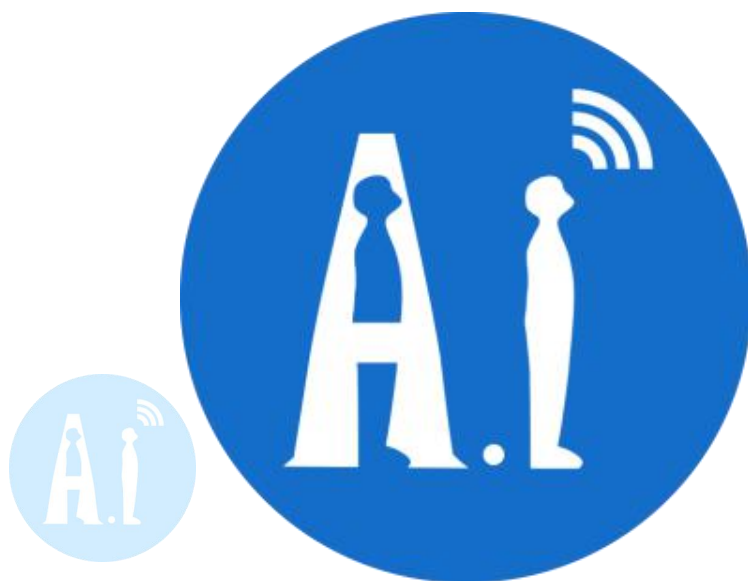


ESP32 模组系列常见问题



版本 V1.0

版权 ©2019

关于本手册

本文介绍 ESP32 模组的常见问题以及解答。

日期	版本	发布说明	制定	核审
2019. 12. 18	V1. 0	首次发布	Mr. Su	徐宏

文档更新通知：

用户可关注安信可官网 ESP32 模组系列：<https://docs.ai-thinker.com/esp32> 获取最新文档

目录

(一) 硬件篇.....	5
1. ESP32-S,ESP-WROOM-32,ESP32-A1S 模组有什么区别?.....	5
2. ESP32-S, ESP32-A1S 能否外接天线, 如果需要外接天线需要怎么操作?.....	5
3. ESP32-CAM 可以应用哪些场景, 支不支持二次开发, 如果需要二次开发该如何使用?.....	5
4. ESP32-CAM 支持哪款摄像头, 目前已经实现什么功能?.....	6
5. ESP32-A1S 模组主要应用哪个领域, 内置音频解码芯片是哪一个型号, 出厂默认固件是哪一个, 支不支持二次开发.....	6
6. 发生压复位 (Brownout Reset) 的原因是什么?	6
7. 如何关闭默认通过 UART0 发送的调试消息?	6
8. ESP32 有多少组串口可供使用?	6
9. ESP32 如何进入 download_boot 模式?.....	7
10. 客户自研产品如何优化二次谐波等杂散?.....	7
11. ESP32 的不同省电模式下的功耗分别是多少, 如何进入低功耗模式?.....	7
(二) 软件篇.....	8
1. 如何使用 PSRAM,相关文档和实例如何获取?	8
2. AT 提示 “busy” 是什么原因?	8
3. ESP32 的蓝牙和 WIFI 如何共存?	9
4. 如何查看线程使用过的最大栈的大小?	9
5. ESP32 的经典蓝牙支持哪些配置?	9
6. ESP32 蓝牙作为主设备可以同时连接多少个从设备?	9

7.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?	10
8.ESP32 扫描一次需要花多少时间?	10
9.ESP32 的 BLE 蓝牙配网兼容性、性能如何? 是否开源?	10
10.如何测试 Wi-Fi 性能?	10
11.如何修改默认上电校准方式?	11
12.ESP32 一键配网的原理是什么? (SmartConfig)	11
12. 有哪些 AT 指令设置之后数据是保存到内存?	11
14.ESP32AT 指令具体打印信息有具体说明吗?	12
15.如何通过 SDK 开发获取模块的 MAC 地址?	13
16.ESP32 的启动过程是怎样的?	14

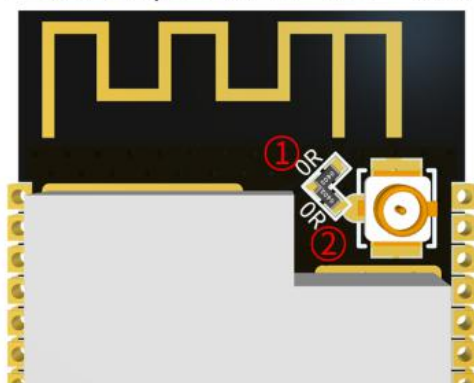
(一) 硬件篇

1. ESP32-S, ESP-WROOM-32, ESP32-A1S 模组有什么区别?

答: ESP32-S 与 ESP32-WROOM-32 为通用类型的模组, 软硬件部分, 除了天线形式有差异, 其他部分都一样, ESP32-A1S 模组内置的是 AC101 音频解码芯片, 可以进行音频相关的开发, 支持播放音乐和录音。

2. ESP32-S, ESP32-A1S 能否外接天线, 如果需要外接天线需要怎么操作?

答: PCB天线/IPEX座 二选一说明



- 1、模组默认连接到 PCB 板载天线(即只接①处的 0R 电阻);
- 2、如需使用 IPEX 座外接天线, 只接②处的 0R 电阻即可(需自行更换 0R 电阻位置);
- 3、0R 电阻只能放在 ① 或 ② 的位置上, 对应使用不同天线, 不能同时放置!

3. ESP32-CAM 可以应用哪些场景, 支不支持二次开发, 如果需要二次开发该如何使用?

答: 可以应用在家庭智能图传, 无线监控, QR 无线识别等场景, 支持二次开发, 开发语言为 C 语言, 如果需要二次开发需要搭建专门的编译环境, 可以使用我司提供的安信可一体化开发环境。

4. ESP32-CAM 支持哪款摄像头，目前已经实现什么功能？

答:目前程序上已经支持 OV2640 型号和 OV7725 型号摄像头的驱动，目前已经实现了 OV2640 输出 JPEG 格式的图片、支持 OV2640 和 OV7725 输出灰度格式图片和 QR 识别的功能。

5. ESP32-A1S 模组主要应用哪个领域，内置音频解码芯片是哪一个型号，出厂默认固件是哪一个，支不支持二次开发

答:主要应用于家庭智能设备、车载智能设备、智能音箱等方案，内置音频解码芯片为 AC101，出厂默认烧录测试固件，工作状态分为两种模式：TF 卡模式和蓝牙模式，两种模式通过按键进行切换，当没有插入 TF 卡时只能切入到蓝牙模式，当插入 TF 卡后可以使用两种任一模式，TF 卡检测只在程序启动时进行，当程序启动后不在检测 TF 卡插入。不支持 AT 指令，如果需要其他功能请参考我们的 ai-example 例子进行固件二次开发。

6. 发生压复位（Brownout Reset）的原因是什么？

答：1. 检查电源是否有问题，低于低电压阈值。
2. 由于芯片作时候电流波动较大，请确认电源的驱动能力是否足够。此外，如果用户使用 USB To TTL 为板子供电，应保证 USB 的质量足以提供足够电流。

7. 如何关闭默认通过 UART0 发送的调试消息？

答:1.Bootloader 信息：GPIO15 接地
2. IDF 信息：可以前往 menuconfig/Component config/Log output 进行配置

8. ESP32 有多少组串口可供使用？

串口：3组
答:

	Rx	Tx
uart0	GPIO3	GPIO1
uart1	GPIO9	GPIO10
uart2	GPIO16	GPIO17

9. ESP32 如何进入 download_boot 模式?

答：进入DOWNLOAD_BOOT模式后，uart0会打印出信息：

```
1 rst:0x1 (POWERON_RESET),boot:0x3 (DOWNLOAD_BOOT(UART0/UART1/SDIO_REI_RE0_V2))
2 waiting for download
```

GPI00 引脚不需要保持电平，芯片内部是检测 EN 脚上升沿时的 BOOT 脚的电平，之后 BOOT 脚的电平不管是什么都可以。

10. 客户自研产品如何优化二次谐波等杂散?

答：二次谐波主要来源于射频链路辐射和 PA 电源辐射，同时容易受到客户底板（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 2.4pF 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。

11. ESP32 的不同省电模式下的功耗分别是多少，如何进入低功耗模式?

答：出厂固件通过 AT 指令配置进行相应的功耗模式，以下表格是不同模式下的功耗：

功耗模式	描述			功耗
Active（射频工作）	Wi-Fi Tx packet			详见表 14
	Wi-Fi/BT Tx packet			
	Wi-Fi/BT Rx 和侦听			
Modem-sleep	CPU 处于工作状态	240 MHz [*]	双核芯片	30 mA ~ 68 mA
			单核芯片	N/A
		160 MHz [*]	双核芯片	27 mA ~ 44 mA
			单核芯片	27 mA ~ 34 mA
		正常速度：80 MHz	双核芯片	20 mA ~ 31 mA
			单核芯片	20 mA ~ 25 mA
Light-sleep	-			0.8 mA
Deep-sleep	ULP 协处理器处于工作状态			150 μ A
	超低功耗传感器监测方式			100 μ A @1% duty
	RTC 定时器 + RTC 存储器			10 μ A
Hibernation	仅有 RTC 定时器处于工作状态			5 μ A
关闭	CHIP_PU 脚拉低，芯片处于关闭状态			0.1 μ A

(二) 软件篇

1. 如何使用 PSRAM, 相关文档和实例如何获取?

答：为了使用 PSRAM，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](#);
- IESP-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。

2. AT 提示 “busy” 是什么原因?

答：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”。

3. ESP32 的蓝牙和 WIFI 如何共存？

答：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 scaninterval 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。

4. 如何查看线程使用过的最大栈的大小？

答：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。

5. ESP32 的经典蓝牙支持哪些配置？

答：ESP-IDF V3.1: HFP Client (not HF gateway)

ESP-IDF V3.0: A2DP Source/A2DP Sink/AVRCP/AVDTP/SPP/RFCOMM

6. ESP32 蓝牙作为主设备可以同时连接多少个从设备？

答：目前同时支持最多 7 个 BLE 连接，并且可以在 menuconfig 中配置该数量。

7. 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?

答: PWM 可输入到任何管脚, 除了只有输入功能的 I/O 之外。

8. ESP32 扫描一次需要花多少时间?

答: 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: $11 \times 120 + 2 \times 360 = 2040\text{ms}$;
- 在 Station 模式有连接, 或者 Station-AP 模式下, 全信道扫描总时间为: $11 \times 120 + 2 \times 360 + 13 \times 30 = 2430\text{ ms}$ 。

9. ESP32 的 BLE 蓝牙配网兼容性、性能如何? 是否开源?

答: 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。

10. 如何测试 Wi-Fi 性能?

答: 请使用 example/wifi/iperf 目录下代码进行测试。

11. 如何修改默认上电校准方式？

答：• 上电时 RF 初始化默认采用部分校准的方案

打开 menuconfig 中 CONFIG_ESP32_PHY_CALIBRATION_AND_DATA_STORAGE 选项。

• 不关注上电启动时间，可修改使用上电全校准方案

关闭 menuconfig 中 CONFIG_ESP32_PHY_CALIBRATION_AND_DATA_STORAGE 选项。

• 继续使用上电部分校准方案，若需在业务逻辑中增加触发全校准操作的功能擦除 NVS 分区中的内容，触发全校准操作。

12. ESP32 一键配网的原理是什么？（SmartConfig）

答：ESP32 采用 UDP 广播模式（UDP 接收 IP 地址是 255.255.255.255）。

WiFi 设备先 scan 环境下 AP，得到 AP 的相关信息，如工作的 channel，然后配置 WiFi 芯片工作在刚才 scan 到的 channel 上去接收 UDP 包，如果没有接收到，继续配置工作在另外的 channel 上，如此循环，直到收到 UDP 包为止。

12. 有哪些 AT 指令设置之后数据是保存到内存？

答：

指令	描述
AT+UART	AT+UART=115200,8,1,0,3
AT+UART_DEF	AT+UART_DEF=115200,8,1,0,3
AT+CWDHCP	AT+CWDHCP=1,1
AT+CIPSTAMAC	AT+CIPSTAMAC="18:fe:35:98:d3:7b"
AT+CIPAPMAC	AT+CIPAPMAC="1a:fe:36:97:d5:7b"
AT+CIPSTA	AT+CIPSTA="192.168.6.100"
AT+CIPAP	AT+CIPAP="192.168.5.1"
AT+CWDHCP5	AT+CWDHCP5=1,3,"192.168.4.10","192.168.4.15"
AT+SAVETRANSLINK	AT+SAVETRANSLINK=1,"192.168.6.10",1001
AT+CNMODE	AT+CNMODE=3
AT+CWJAP	AT+CWJAP="abc","0123456789"
AT+CWSAP	AT+CWSAP="ESP32","12345678",5,3
AT+CWAUTOCONN	AT+CWAUTOCONN=1
AT+CIPSSLCCONF	AT+CIPSSLCCONF=1,3,0,0

⚠ 注意：

NVS 参数区为 0xFA000 - 0x110000, 88 KB。

14. ESP32AT 指令具体打印信息有具体说明吗？

答：

ESP32 AT 指令中的提示信息说明如下：

提示信息	说明
ready	AT 固件成功启动
ERROR	指令输入错误，或者指令执行出错
WIFI CONNECTED	ESP station 连接到 AP
WIFI GOT IP	ESP station 获取到 IP 地址
WIFI DISCONNECT	ESP station 的 WiFi 连接断开
busy p...	busy processing，表示系统正在处理前一条指令，无法响应当前输入
<conn_id>,CONNECT	建立了 <conn_id> 号网络连接
<conn_id>,CLOSED	<conn_id> 号网络连接断开
+IPD	接收到网络数据
+STA_CONNECTED:<sta_mac>	有 station 连入 ESP softAP
+DIST_STA_IP:<sta_mac>,<sta_ip>	ESP softAP 给连入的 station 分配 IP 地址
+STA_DISCONNECTED:<sta_mac>	station 从 ESP softAP 断开连接
+BLECONN	BLE 建立连接

+BLECONN	BLE 建立连接
+BLEDISCONN	BLE 断开连接
+READ	BLE 接收到读操作
+WRITE	BLE 接收到写操作
+NOTIFY	BLE 接收到通知操作
+INDICATE	BLE 接收到指示操作
+BLESECNTFYKEY	BLE SMP 密钥
+BLEAUTHCMPL	BLE SMP 配对完成

15. 如何通过 SDK 开发获取模块的 MAC 地址？

答：

```
id wifi_init_sta()

uint8_t macaddr[6];
s_wifi_event_group = xEventGroupCreate();

tcpip_adapter_init();
ESP_ERROR_CHECK(esp_event_loop_init(event_handler, NULL) );

wifi_init_config_t cfg = WIFI_INIT_CONFIG_DEFAULT();
ESP_ERROR_CHECK(esp_wifi_init(&cfg));
wifi_config_t wifi_config = {
    .sta = {
        .ssid = EXAMPLE_ESP_WIFI_SSID,
        .password = EXAMPLE_ESP_WIFI_PASS
    },
};

ESP_ERROR_CHECK(esp_wifi_set_mode(WIFI_MODE_STA) );
ESP_ERROR_CHECK(esp_wifi_set_config(ESP_IF_WIFI_STA, &wifi_config) );
ESP_ERROR_CHECK(esp_wifi_get_mac(ESP_IF_WIFI_STA, macaddr));
ESP_ERROR_CHECK(esp_wifi_start() );

ESP_LOGI(TAG, "MACADDR : %02x:%02x:%02x:%02x:%02x:%02x\r\n", macaddr[0], macaddr[1], macaddr[2], macaddr[3], macaddr[4], macaddr[5]);
ESP_LOGI(TAG, "wifi_init_sta finished.");
ESP_LOGI(TAG, "connect to ap SSID:%s password:%s",
    EXAMPLE_ESP_WIFI_SSID, EXAMPLE_ESP_WIFI_PASS);
```

```
althinker@althinker: ~/esp/esp-idf/examples/wifi/getting_started/station
I (261) system api: Base MAC address is not set, read default base MAC address from BLK0 of EFUSE
I (291) wifi: Init dynamic tx buffer num: 32
I (291) wifi: Init data frame dynamic rx buffer num: 32
I (291) wifi: Init management frame dynamic rx buffer num: 32
I (301) wifi: Init management short buffer num: 32
I (301) wifi: Init static rx buffer size: 1600
I (311) wifi: Init static rx buffer num: 10
I (311) wifi: Init dynamic rx buffer num: 32
W (331) phy_init: failed to load RF calibration data (0x1102), falling back to full calibration
I (541) phy: phy_version: 4008, 544f89f, Jan 24 2019, 14:54:06, 0, 2
I (551) wifi: mode : sta (24:0a:c4:2c:9a:98)
I (551) wifi station: MACADDR : 24:0a:c4:2c:9a:98

I (551) wifi station: wifi_init_sta finished.
I (551) wifi station: connect to ap SSID:test_2G password:TEST123456ok
I (1281) wifi: n:6 2, o:1 0, ap:255 255, sta:6 2, prof:1
I (2271) wifi: state: init -> auth (b0)
I (2271) wifi: state: auth -> assoc (0)
I (2281) wifi: state: assoc -> run (10)
I (2301) wifi: connected with test_2G, channel 6
I (2301) wifi: pm start, type: 1
```


16. ESP32 的启动过程是怎样的？

答：

[关于 ROM]

在esp32 上电运行后，芯片运行的第一个程序。这段程序是芯片设计与生产的时候，固化在硬件电路中的。所以它是不可修改的(Read Only Memory)。

esp32 的 ROM 负责检测芯片的strapping配置，来决定芯片应该处于什么状态。比如，esp32 上电后，ROM 程序会检查 [GPIO0, GPIO2, GPIO4, MTDO, GPIO5]的状态。

如果 GPIO0 / GPIO2 同时为低电平，则会进入下载模式，等待串口通信信息。

如果GPIO0为高电平，则会进入Flash 运行模式，启动SPI 驱动，并加载Flash中的程序段。

BOOT_MODE[5:0]:

(pull-up, pull-down, pull-down, pull-up, pull-up, SW4 /5/4/3/2/1/)

[GPIO0, GPIO2, GPIO4, MTDO, GPIO5]

1 x x x x --> SPI Boot

0 0 x x x --> Download Boot (Junit-Detection of UART0+UART1+SDIO_Slave)

下载模式的串口输出如下(115200), ROM默认会输出当前所处的模式。

rst:0x1 (POWERON_RESET),boot:0x3 (DOWNLOAD_BOOT(UART0/UART1/SDIO_RE1_RE0_V2))

waiting for download

其中，boot:0x3 表示的是芯片strapping pin脚的状态，

0x03对应 [GPIO0, GPIO2, GPIO4, MTDO, GPIO5] 的值为 [0, 0, 0, 1, 1]

所以处于 Download Boot 模式。

[关于下载模式]

当esp32处于下载模式时，会等待串口通信同步，并按照通信协议等待接收指令(协议可参考该文档：<https://github.com/espressif/esptool/wiki/Download-Protocol>)

通过esptool脚本，可以进行寄存器的读写，固件下载，程序运行等操作。

[关于STUB]

在ROM模式，由于芯片处于低频工作的状态，通信速率受限。

在esptool中，会将一段小程序加载到esp32的RAM中，并跳转执行RAM中的小程序。这段小程序包含了ROM中相同的串口通信协议，并对其进行了扩充。感兴趣的开发者，可以参考这里https://github.com/espressif/esptool/tree/master/asher_stub

[关于 Flash Boot 模式]

如果芯片启动时，GPIO0为高电平，芯片会进入Flash 运行模式。此时，启动SPI 驱动，并加载Flash中的程序段。ROM 会读取外置 Flash 的0x1000地址，加载并运行二级bootloader。

[关于 Bootloader]

bootloader可以认为是一个独立的小程序，bootloader会对芯片频率进行初始化，并且读取系统SPI的配置信息，对Flash 运行模式以及频率进行配置，然后根据分区表的定义，从对应的地址加载应用程序，并且运行应用程序固件。

联系我们：

官方官网：<https://www.ai-thinker.com>

开发 Wiki：<http://wiki.ai-thinker.com>

官方论坛：<http://bbs.ai-thinker.com>

样品购买：<https://anxinke.taobao.com>

商务合作：sales@aithinker.com

技术支持：support@aithinker.com

公司地址：广东省深圳市宝安区西乡街道航空路华丰第二工业园 B 座七楼

联系电话：0755-29162996/15302646692

