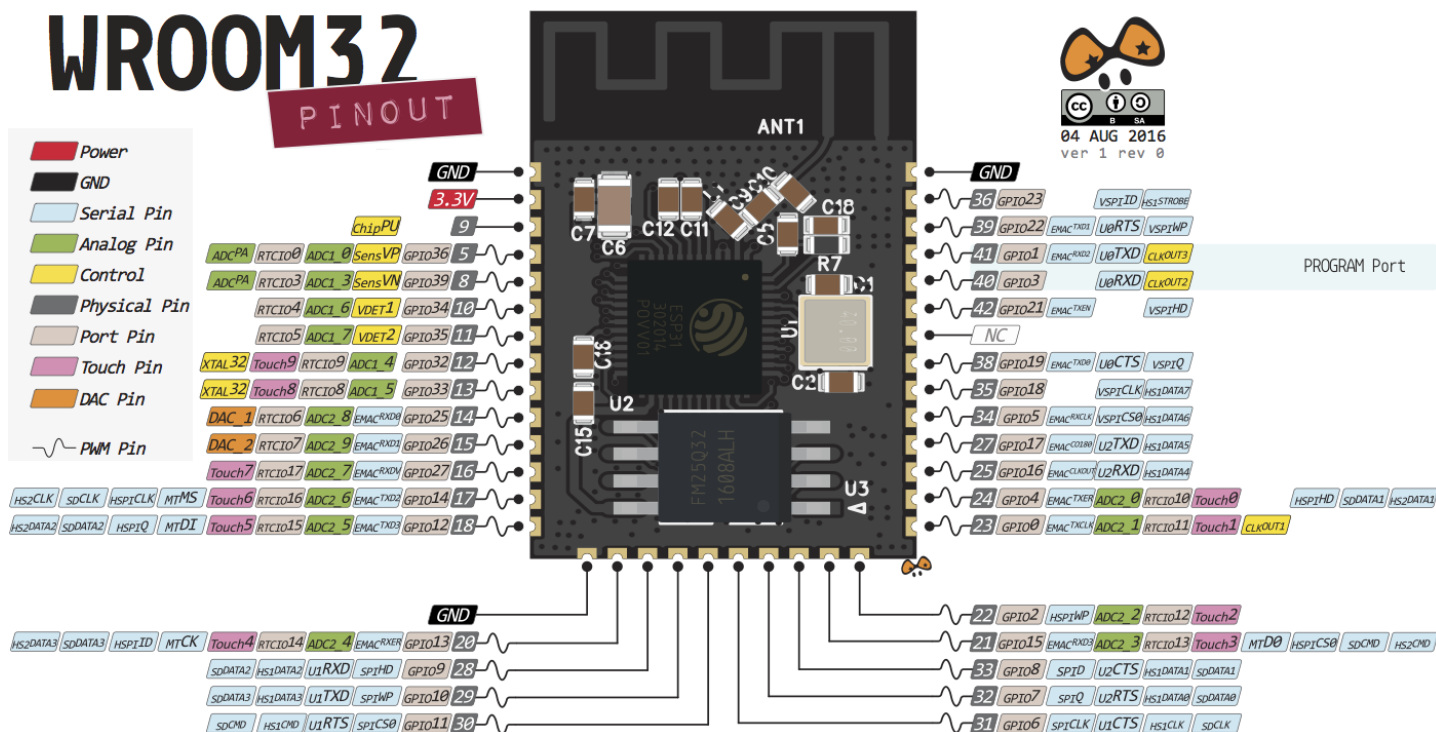


ESP32 – GPIO 引脚参考大全

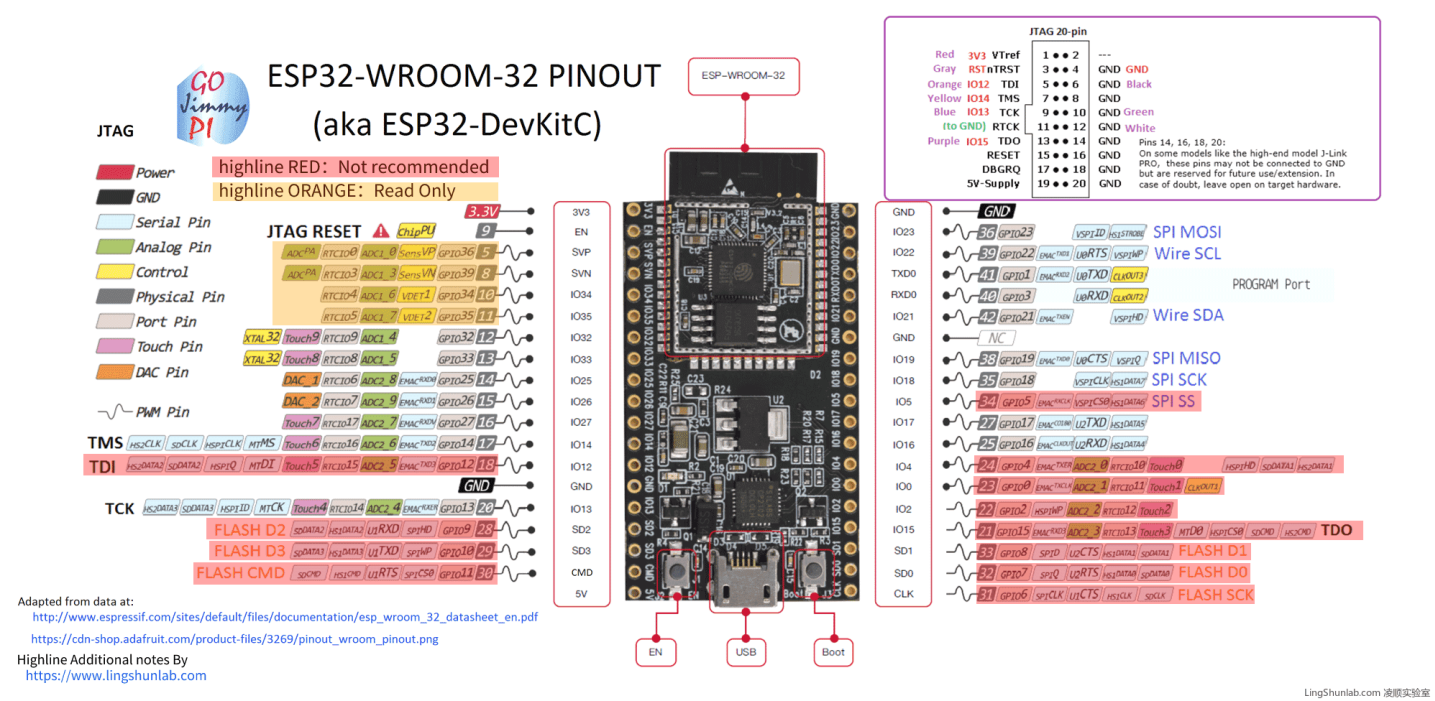
目录

- ESP32 模组引脚图
- ESP32 开发板引脚图
- 不建议使用或限制使用的引脚
 - Strapping 引脚
 - 集成在ESP-WROOM-32 的 SPI flash 引脚
 - 仅输入引脚
- ESP32 周边设备：
 - 18 个模数转换器 (ADC) 通道
 - 4组SPI接口
 - I2C 通讯协议
 - PWM
 - 数模转换器 (DAC)
 - I2S 高速数位音讯传输标准协议
 - 电容式触控 GPIO
 - RTC GPIO
 - 中断
 - 启用 (EN)
 - GPIO 电流消耗
 - ESP32 内置霍尔效应传感器
- 本文的资料参考：

ESP32 模组引脚图



ESP32 开发板引脚图



重要的说在前面，在使用ESP32的时候不建议使用以下引脚，因为会在项目中出现各种无解的问题。

不建议使用或限制使用的引脚

不建议使用 **Strapping** 引脚，**SPI flash** 引脚 以及 仅输入的引脚

Strapping 引脚

- GPIO 0
- GPIO 2
- GPIO 4
- GPIO 5 (启动时必须为高电平)
- GPIO 12 (启动时必须为低电平)
- GPIO 15 (启动时必须为高电平)

注意：
在硬件上要注意使用外接模块时不能将GPIO12拉高，否则将导致ESP32启动异常。还有一些GPIO在启动或重置时其状态更改为高或者输出PWM信号，在使用时需要注意。

集成在ESP-WROOM-32 的 SPI flash 引脚

GPIO 6 到 GPIO 11 在一些 ESP32 开发板中公开。但是，这些引脚连接到 ESP-WROOM-32 芯片上的集成 SPI 闪存，不推荐用于其他用途。所以，不要在你的项目中使用这些引脚：

- GPIO 6 (SCK/CLK)
- GPIO 7 (SD0/SD0)

GPIO 8 (SDI/SD1)
GPIO 9 (SHD/SD2)
GPIO 10 (SWP/SD3)
GPIO 11 (CSC/CMD)

仅输入引脚

GPIO 34 到 39 是 GPI – 仅输入引脚。这些引脚没有内部上拉或下拉电阻。它们不能用作输出，因此只能将这些引脚用作输入：

GPIO 34
GPIO 35
GPIO 36
GPIO 39

这些引脚都是ESP32用于引导加载程序或者烧录模式/在大多数内置USB/Serial的开发板上，不需要担心这些引脚的状态，开发板会把这些引脚设置为正确的状态，以便使用烧录或启动模式。

但是，如果你有外设连接到这些引脚上，当你在尝试上传新代码、用新固件烧写ESP32或重置电路板时可能会遇到麻烦，例如不明原因的错误和失败。可能是因为这些外设阻止ESP32进入正确的模式。

所以以上的引脚 不建议在项目中使用。

ESP32 周边设备：

- 18 个模数转换器 (ADC) 通道
- 4 组SPI接口
- 3 组UART接口
- 1 组I2C接口
- 16 个PWM输出通道
- 2 个数模转换器 (DAC)
- 2 个I2S接口
- 10 个电容式感应 GPIO
- 16 个 RTC GPIO
- ESP32 内置2个霍尔传感器

18 个模数转换器 (ADC) 通道

ESP32 有 18 x 12 位 ADC 输入通道（而ESP8266 只有 1x 10 位 ADC）。这些是可用作 ADC 和相应通道的 GPIO：

与某些数字外设（PWM、软件 SPI 和 I2C）不同，ADC 引脚是固定的，即您必须使用具有 ADC 功能的预定义 GPIO 引脚，并且不能在软件中对其进行配置。但是，您必须了解一些限制。

尽管 ESP32 有 18 个通道的 ADC，但并非所有 ADC 引脚都可供用户使用。在 8 个 ADC1 通道中，只有 6 个可用（ACD1_CH0 和 ACD1_CH3 到 ACD1_CH7），而 ADC1_CH1 和 ADC1_CH2 不可用（即使引脚在 ESP32 开发板中没有暴露）。

来到ADC2，就有些复杂了。当您使用 ESP32 的 Wi-Fi 时，Wi-Fi Driver 使用 ADC2 Peripheral。因此，只有在未启动 Wi-Fi 驱动程序时才能使用 ADC2。

即使您正在使用 ADC2（假设未使用 Wi-Fi），所有引脚也并非随时可用，因为与 ADC2 相关的一些引脚用于其他重要目的（引导绑定）。

ADC Channel	Pin Name	GPIO Pin	Notes
ACD1_CH0	A0	36	可用 / 内置霍尔传感器
ACD1_CH1	\	37	不可用
ACD1_CH2	\	38	不可用
ACD1_CH3	A3	39	可用 / 内置霍尔传感器
ACD1_CH4	A4	32	可用
ACD1_CH5	A5	33	可用
ACD1_CH6	A6	34	可用
ACD1_CH7	A7	35	可用
ACD2_CH0	A10	4	启动WIFI时不可用
ACD2_CH1	A11	0	不可用 / 用于BOOT
ACD2_CH2	A12	2	用于BOOT Strapping
ACD2_CH3	A13	15	用于BOOT Strapping
ACD2_CH4	A14	13	启动WIFI时不可用
ACD2_CH5	A15	12	启动WIFI时不可用
ACD2_CH6	A16	14	启动WIFI时不可用
ACD2_CH7	A17	27	启动WIFI时不可用
ACD2_CH8	A18	25	启动WIFI时不可用
ACD2_CH9	A19	26	启动WIFI时不可用

ADC 输入通道具有 12 位分辨率。这意味着您可以获得范围从 0 到 4095 的模拟读数，其中 0 对应于 0V，4095 对应于 3.3V。您还可以在代码和 ADC 范围上设置通道的分辨率。

ESP32 ADC 引脚没有线性行为。您可能无法区分 0 和 0.1V，或 3.2 和 3.3V。使用 ADC 引脚时需要牢记这一点。您将获得类似于下图所示的行为。

4组SPI接口

默认情况下，可以用的SPI的引脚映射是：

SPI	MOSI	MISO	CLK	CS
VSPI (SPI3)	GPIO 23	GPIO 19	GPIO 18	GPIO 5

SPI	MOSI	MISO	CLK	CS
HSPI (SPI2)	GPIO 13	GPIO 12	GPIO 14	GPIO 15

SPI（Serial Peripheral Interface） 协议是由摩托罗拉公司提出的通讯协议，即串行外围设备接口，是一种高速全双工的通信总线。它被广泛地使用在 ADC、LCD 等设备与 MCU 间，要求通讯速率较高的场合。

芯片的管脚上只占用四根线。

MISO：主器件数据输出，从器件数据输入。

MOSI：主器件数据输入，从器件数据输出。

SCK：时钟信号，由主设备控制发出。

NSS（CS）：从设备选择信号，由主设备控制。当NSS为低电平则选中从器件。

ESP32集成了4组SPI外设。

- SPI0和SPI1在内部用于访问ESP32所连接的闪存。两个控制器共享相同的SPI总线信号，并且有一个仲裁器来确定哪个可以访问该总线。
- SPI2和SPI3是**通用SPI控制器**，有时分别称为HSPI和VSPI。它们向用户开放。SPI2和SPI3具有独立的总线信号，分别具有相同的名称。每条总线具有3条CS线，最多能控制6个SPI从设备。

需要注意的是，HSPI和VSPI并不是网友们认为的high-speed SPI 和Very High-speed SPI，这是SPI和HSPI、VSPI是一样的，只不过是换个名字用于区分，SPI相当于SPI0或SPI1，HSPI相当于SPI2，VSPI相当于SPI3。

GP-SPI 四线 全双工信号总线	GP-SPI 三线 半双工信号总线	QSPI 信号总线	引脚功能信号		
			SPI 信号总线	HSPI 信号总线	VSPI 信号总线
MOSI	DATA	D	SPID	HSPID	VSPID
MISO	-	Q	SPIQ	HSPIQ	VSPIQ
CS	CS	CS	SPICS0	HSPICS0	VSPICS0
CLK	CLK	CLK	SPICLK	HSPICLK	VSPICLK
-	-	WP	SPIWP	HSPIWP	VSPIWP
-	-	HD	SPIHD	HSPIHD	VSPIHD

<https://blog.csdn.net/jdsnpgxj>

I2C 通讯协议

ESP32默认I2C引脚为：

- GPIO 21 （SDA）
- GPIO 22 （SCL）

其实在ESP32中任何引脚都可以定义为SDA或SCL，但不到逼不得已不推荐这么做。

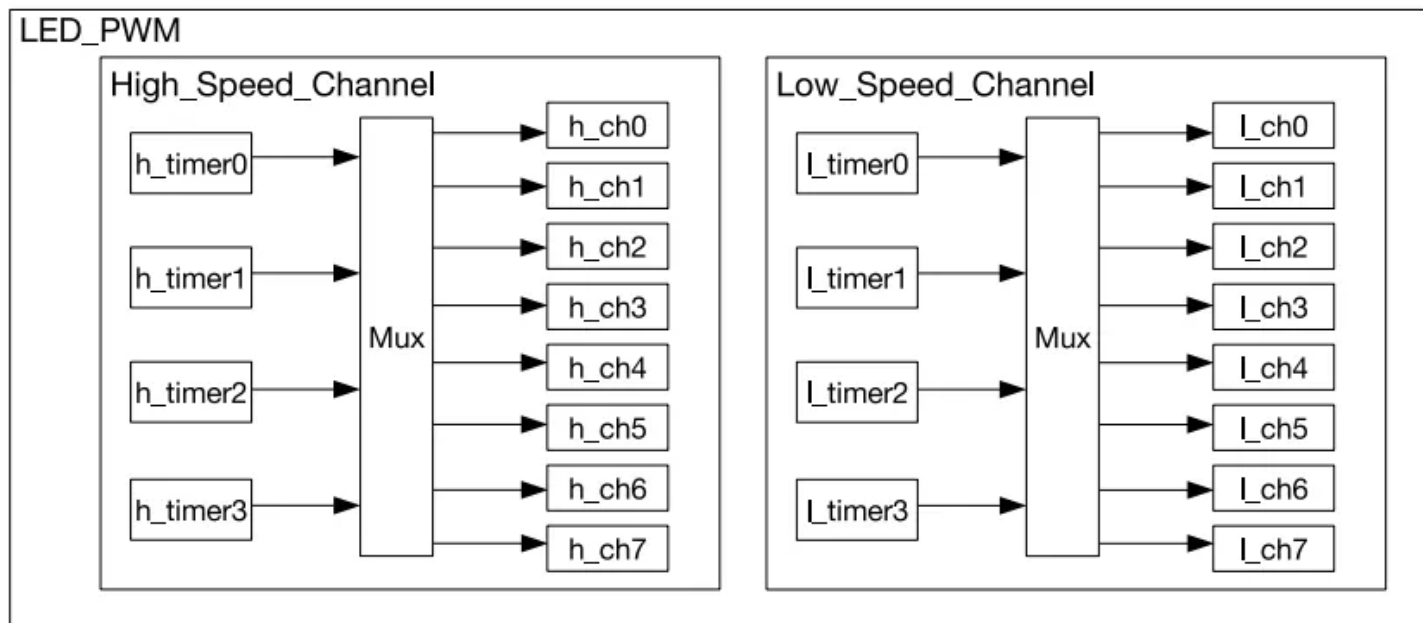
可在Arduino IDE 中使用以下语句配置其它引脚为SDA或SCL

```
Wire.begin(SDA, SCL);
```

PWM

ESP32 PWM 控制器主要设计用于控制 LED 的强度，但它也可用于生成其他目的的 PWM 信号。它有16个通道，可以产生独立的PWM波形。

ESP32 PWM 控制器有 8 个高速通道和 8 个低速通道，我们总共有 16 个通道。它们根据速度分为两组。每组有 4 个定时器/8 个通道。这意味着每两个通道共享同一个定时器。因此，我们无法独立控制每对通道的 PWM 频率。



所有可以作为输出的引脚都可以用作 PWM 引脚（GPIO 34 到 39 不能产生 PWM）。

要设置 PWM 信号，您需要在代码中定义这些参数：

- 信号的频率；
- 占空比；
- 脉宽调制通道；
- 要输出信号的 GPIO。

数模转换器 (DAC)

ESP32 上有 2 x 8 位 DAC 通道，用于将数字信号转换为模拟电压信号输出。这些是 DAC 通道：

- DAC1 (GPIO25)
- DAC2 (GPIO26)

I2S 高速数位音讯传输标准协议

在ESP32引脚上实际是标记为 DAC1 和 DAC2

- GPIO 25
- GPIO 26

电容式触控 GPIO

ESP32 有 10 个内部电容式触摸传感器。它们可以感知任何带电荷的物体的变化，比如人体皮肤。因此，他们可以检测用手指触摸 GPIO 时引起的变化。这些引脚可以轻松集成到电容式焊盘中并取代机械按钮。电容式触控引脚还可用于将 ESP32 从深度睡眠中唤醒

这些内部触摸传感器连接到这些 GPIO：

- T0 (GPIO 4)
- T1 (GPIO 0)
- T2 (GPIO 2)
- T3 (GPIO 15)
- T4 (GPIO 13)
- T5 (GPIO 12)
- T6 (GPIO 14)
- T7 (GPIO 27)
- T8 (GPIO 33)
- T9 (GPIO 32)

RTC GPIO

ESP32 上有 RTC GPIO 支持。当 ESP32 处于深度睡眠时，可以使用路由到 RTC 低功耗子系统的 GPIO。当超低功耗 (ULP) 协处理器运行时，这些 RTC GPIO 可用于将 ESP32 从深度睡眠中唤醒。以下 GPIO 可用作外部唤醒源

- RTC_GPIO0 (GPIO36)
- RTC_GPIO3 (GPIO39)
- RTC_GPIO4 (GPIO34)
- RTC_GPIO5 (GPIO35)
- RTC_GPIO6 (GPIO25)
- RTC_GPIO7 (GPIO26)
- RTC_GPIO8 (GPIO33)
- RTC_GPIO9 (GPIO32)
- RTC_GPIO10 (GPIO4)
- RTC_GPIO11 (GPIO0)
- RTC_GPIO12 (GPIO2)
- RTC_GPIO13 (GPIO15)
- RTC_GPIO14 (GPIO13)
- RTC_GPIO15 (GPIO12)
- RTC_GPIO16 (GPIO14)
- RTC_GPIO17 (GPIO27)

中断

所有 GPIO 都可以配置为中断。

启用 (EN)

启用 (EN) 是 3.3V 稳压器的启用引脚。它被拉高，因此接地以禁用 3.3V 稳压器。例如，这意味着您可以使用连接到按钮的此引脚来重新启动 ESP32。

GPIO 电流消耗

根据 ESP32 数据表中的“推荐操作条件”部分，每个 GPIO 的绝对最大电流消耗为 40mA。

ESP32 内置霍尔效应传感器

ESP32 还具有内置霍尔效应传感器，可检测周围磁场的变化。

- GPIO 36
- GPIO 39

本文的资料参考：

<https://randomnerdtutorials.com/esp32-pinout-reference-gpios/>

<https://deepbluembedded.com/esp32-pwm-tutorial-examples-analogwrite-arduino/>

<https://www.jianshu.com/p/6fb40b3d0c6d>

<https://blog.csdn.net/jdsnpgxj/article/details/100013357>