# **快速参考ESP8266**

## **通用方法**

**[machine](https://docs.micropython.org/en/latest/esp8266/library/machine.html" \l "module-machine" \o "机器：与硬件有关的功能)类库：**

|  |
| --- |
| import machine  machine.freq() *# get the current frequency of the CPU*  machine.freq(160000000) *# set the CPU frequency to 160 MHz* |

**esp类库：**

|  |
| --- |
| **import** esp  esp**.**osdebug(None) *# turn off vendor O/S debugging messages*  esp**.**osdebug(0) *# redirect vendor O/S debugging messages to UART(0)* |

## **联网**

**network类库**：

|  |
| --- |
| **import** network  wlan **=** network**.**WLAN(network**.**STA\_IF) *# create station interface*  wlan**.**active(True) *# activate the interface*  wlan**.**scan() *# scan for access points*  wlan**.**isconnected() *# check if the station is connected to an AP*  wlan**.**connect('essid', 'password') *# connect to an AP*  wlan**.**config('mac') *# get the interface's MAC adddress*  wlan**.**ifconfig() *# get the interface's IP/netmask/gw/DNS addresses*  ap **=** network**.**WLAN(network**.**AP\_IF) *# create access-point interface*  ap**.**active(True) *# activate the interface*  ap**.**config(essid**=**'ESP-AP') *# set the ESSID of the access point* |

**连接到本地WiFi网络：**

|  |
| --- |
| **def do\_connect():**  **import** network  wlan **=** network**.**WLAN(network**.**STA\_IF)  wlan**.**active(True)  **if** **not** wlan**.**isconnected():  **print**('connecting to network...')  wlan**.**connect('essid', 'password')  **while** **not** wlan**.**isconnected():  **pass**  **print**('network config:', wlan**.**ifconfig()) |

## **延迟和时间**

**time 类库**

|  |
| --- |
| **import** time  time**.**sleep(1) *# sleep for 1 second*  time**.**sleep\_ms(500) *# sleep for 500 milliseconds*  time**.**sleep\_us(10) *# sleep for 10 microseconds*  start **=** time**.**ticks\_ms() *# get millisecond counter*  delta **=** time**.**ticks\_diff(time**.**ticks\_ms(), start) *# compute time difference* |

## **计时器**

|  |
| --- |
| **import** time  time**.**sleep(1) *# sleep for 1 second*  time**.**sleep\_ms(500) *# sleep for 500 milliseconds*  time**.**sleep\_us(10) *# sleep for 10 microseconds*  start **=** time**.**ticks\_ms() *# get millisecond counter*  delta **=** time**.**ticks\_diff(time**.**ticks\_ms(), start) *# compute time difference* |

周期以毫秒为单位。

## **引脚和GPIO**

使用machine.Pin类库：

|  |
| --- |
| **from** machine **import** Pin  p0 **=** Pin(0, Pin**.**OUT) *# create output pin on GPIO0*  p0**.**high() *# set pin to high*  p0**.**low() *# set pin to low*  p0**.**value(1) *# set pin to high*  p2 **=** Pin(2, Pin**.**IN) *# create input pin on GPIO2*  **print**(p2**.**value()) *# get value, 0 or 1*  p4 **=** Pin(4, Pin**.**IN, Pin**.**PULL\_UP) *# enable internal pull-up resistor*  p5 **=** Pin(5, Pin**.**OUT, value**=**1) *# set pin high on creation* |

可用引脚为：0,1,2,3,4,5,12,13,14,15,16，其对应于ESP8266芯片的实际GPIO引脚号。请注意，许多终端用户板使用自己的adhoc引脚编号（标记为D0，D1，...）。由于MicroPython支持不同的单板和模块，所以选择物理引脚编号作为最低的公分母。对于逻辑引脚和物理芯片引脚之间的映射，请参阅电路板文档。

注意，引脚（1）和引脚（3）分别是REPL UART TX和RX。还要注意，Pin（16）是一个特殊的引脚（用于从深睡眠模式唤醒），可能不适用于更高级别的类 Neopixel。

## **PWM（脉宽调制）**

除引脚（16）外的所有引脚都可以使能PWM。所有通道都有一个频率，范围介于1到1000（以Hz为单位）。占空比介于0和1023之间。

使用machine.PWM类：

|  |
| --- |
| **from** machine **import** Pin, PWM  pwm0 **=** PWM(Pin(0)) *# create PWM object from a pin*  pwm0**.**freq() *# get current frequency*  pwm0**.**freq(1000) *# set frequency*  pwm0**.**duty() *# get current duty cycle*  pwm0**.**duty(200) *# set duty cycle*  pwm0**.**deinit() *# turn off PWM on the pin*  pwm2 **=** PWM(Pin(2), freq**=**500, duty**=**512) *# create and configure in one go* |

## **ADC（模数转换）**

ADC在专用引脚上可用。请注意，ADC引脚上的输入电压必须在0v和1.0v之间。

使用machine.ADC类：

|  |
| --- |
| **from** machine **import** ADC  adc **=** ADC(0) *# create ADC object on ADC pin*  adc**.**read() *# read value, 0-1024* |

## **SPI总线**

有两个SPI驱动程序。一个在软件（bit-banging）中实现，并可在所有引脚上工作：

|  |
| --- |
| **from** machine **import** Pin, SPI  *# construct an SPI bus on the given pins# polarity is the idle state of SCK# phase=0 means sample on the first edge of SCK, phase=1 means the second*  spi **=** SPI(**-**1, baudrate**=**100000, polarity**=**1, phase**=**0, sck**=**Pin(0), mosi**=**Pin(2), miso**=**Pin(4))  spi**.**init(baudrate**=**200000) *# set the baudrate*  spi**.**read(10) *# read 10 bytes on MISO*  spi**.**read(10, 0xff) *# read 10 bytes while outputing 0xff on MOSI*  buf **=** bytearray(50) *# create a buffer*  spi**.**readinto(buf) *# read into the given buffer (reads 50 bytes in this case)*spi**.**readinto(buf, 0xff) *# read into the given buffer and output 0xff on MOSI*  spi**.**write(b'12345') *# write 5 bytes on MOSI*  buf **=** bytearray(4) *# create a buffer*  spi**.**write\_readinto(b'1234', buf) *# write to MOSI and read from MISO into the buffer*  spi**.**write\_readinto(buf, buf) *# write buf to MOSI and read MISO back into buf* |

硬件SPI更快（高达80Mhz），但仅适用于以下引脚： MISOGPIO12 MOSI是GPIO13，SCK是GPIO14。它具有与上述bitbanging SPI类相同的方法，除了构造函数和init的引脚参数（正如固定的那样）：

|  |
| --- |
| **from** machine **import** Pin, SPI  hspi **=** SPI(1, baudrate**=**80000000, polarity**=**0, phase**=**0) |

**I2C总线**

I2C驱动程序通过以下程序来实现，并可在TPYBoard v202所有引脚上工作：

|  |
| --- |
| **from** machine **import** Pin, I2C *# construct an I2C bus*  i2c **=** I2C(scl**=**Pin(5), sda**=**Pin(4), freq**=**100000)  i2c**.**readfrom(0x3a, 4) *# read 4 bytes from slave device with address 0x3a*  i2c**.**writeto(0x3a, '12') *# write '12' to slave device with address 0x3a*  buf **=** bytearray(10) *# create a buffer with 10 bytes*  i2c**.**writeto(0x3a, buf) *# write the given buffer to the slave* |

**深度睡眠模式**

将GPIO16连接到复位引脚（HUZZAH上的RST）。可以使用以下代码进行睡眠，唤醒并检查复位原因：

|  |
| --- |
| **import** machine  *# configure RTC.ALARM0 to be able to wake the device*  rtc **=** machine**.**RTC()rtc**.**irq(trigger**=**rtc**.**ALARM0, wake**=**machine**.**DEEPSLEEP)  *# check if the device woke from a deep sleep***if** machine**.**reset\_cause() **==** machine**.**DEEPSLEEP\_RESET:  **print**('woke from a deep sleep')  *# set RTC.ALARM0 to fire after 10 seconds (waking the device)*  rtc**.**alarm(rtc**.**ALARM0, 10000)  *# put the device to sleep*  machine**.**deepsleep() |

**OneWire驱动**

OneWire驱动程序通过以下程序实现，并可在TPYBoard v202所有引脚上工作：

|  |
| --- |
| **from** machine **import** Pin**import** onewire  ow **=** onewire**.**OneWire(Pin(12)) *# create a OneWire bus on GPIO12*  ow**.**scan() *# return a list of devices on the bus*  ow**.**reset() *# reset the bus*  ow**.**readbyte() *# read a byte*  ow**.**writebyte(0x12) *# write a byte on the bus*  ow**.**write('123') *# write bytes on the bus*  ow**.**select\_rom(b'12345678') *# select a specific device by its ROM code* |

**DS18S20和DS18B20的驱动程序：**

|  |
| --- |
| **import** time**,** ds18x20  ds **=** ds18x20**.**DS18X20(ow)  roms **=** ds**.**scan()  ds**.**convert\_temp()  time**.**sleep\_ms(750)  **for** rom **in** roms:  **print**(ds**.**read\_temp(rom)) |

确保在数据线上放置4.7k的上拉电阻。请注意，convert\_temp()每次要采样温度时都必须调用该方法。

**NeoPixel驱动**

使用neopixel类库：

|  |
| --- |
| **from** machine **import** Pin  **from** neopixel **import** NeoPixel  pin **=** Pin(0, Pin**.**OUT) *# set GPIO0 to output to drive NeoPixels*  np **=** NeoPixel(pin, 8) *# create NeoPixel driver on GPIO0 for 8 pixels*  np[0] **=** (255, 255, 255) *# set the first pixel to white*  np**.**write() *# write data to all pixels*  r, g, b **=** np[0] *# get first pixel colour* |

对于NeoPixel的入门使用：

|  |
| --- |
| **import** esp  esp**.**neopixel\_write(pin, grb\_buf, is800khz) |

**APA102驱动**

使用apa102类库：

|  |
| --- |
| **from** machine **import** Pin  **from** apa102 **import** APA102  clock **=** Pin(14, Pin**.**OUT) *# set GPIO14 to output to drive the clock*  data **=** Pin(13, Pin**.**OUT) *# set GPIO13 to output to drive the data*  apa **=** APA102(clock, data, 8) *# create APA102 driver on the clock and the data pin for 8 pixels*  apa[0] **=** (255, 255, 255, 31) *# set the first pixel to white with a maximum brightness of 31*apa**.**write() *# write data to all pixels*  r, g, b, brightness **=** apa[0] *# get first pixel colour* |

**对于APA102的入门使用：**

|  |
| --- |
| **import** esp  esp**.**apa102\_write(clock\_pin, data\_pin, rgbi\_buf) |

**DHT驱动**

DHT驱动程序通过以下代码实现，并可在TPYBoard v202所有引脚上工作：

|  |
| --- |
| **import** dht**import** machine  d **=** dht**.**DHT11(machine**.**Pin(4))  d**.**measure()d**.**temperature() *# eg. 23 (째C)*  d**.**humidity() *# eg. 41 (% RH)*  d **=** dht**.**DHT22(machine**.**Pin(4))  d**.**measure()d**.**temperature() *# eg. 23.6 (째C)*  d**.**humidity() *# eg. 41.3 (% RH)* |