

2021年第10届广东省创意机器人大赛培训

编程型机器人 编程基础

朱金辉

华南理工大学 软件学院
智能软件与机器人研究室

2021年7月



提纲

- 一. 机器人控制器
- 二. 软件安装
- 三. 简单程序——闪烁的LED
- 四. 进阶程序——闪烁的LED

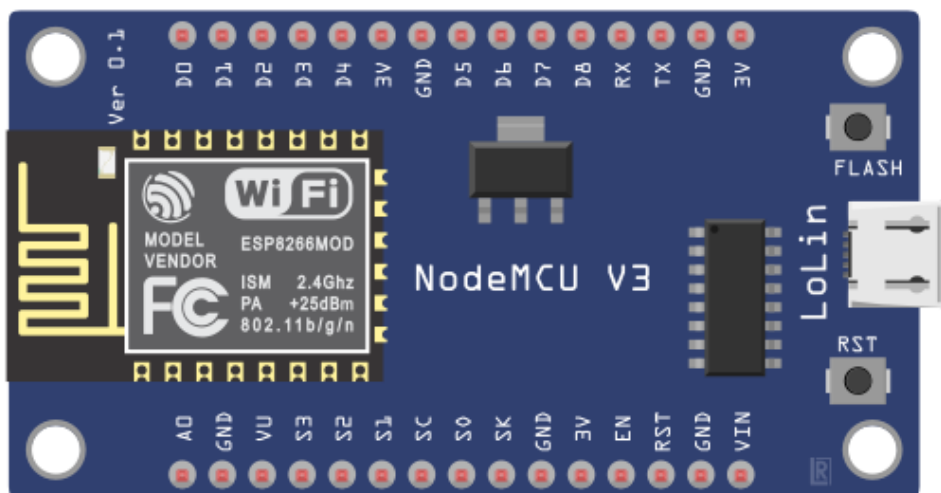


一、机器人控制器



机器人的“大脑”

- 计算机是机器人的“大脑”，称为控制器。
- 计算机上运行机器人软件。软件的运行过程就是机器人的思考过程。
- 单片机是一种小型的计算机。

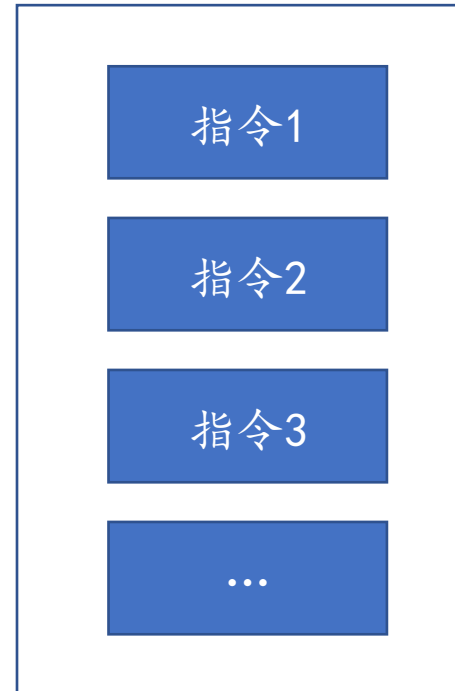




计算机编程

- 通过编写程序来产生机器人软件。
- 程序由一至多条按一定顺序排列的指令组成。例如：加法、乘法等。
- 编程语言：
c/c++
python
...

程序



- 编译型语言：提前将源代码转换成二进制指令，即生成一个可执行文件（例如Windows下的exe文件）。例如C、C++等。转换工具称为编译器。
- 解释型语言：一边执行一边转换，需要哪些源代码就转换哪些源代码，不会生成可执行文件。例如Python等。转换工具称为解释器。



开发流程

源程序：microPython语言

```
led.py  
1 from machine import Pin  
2 import time  
3  
4 led = Pin(2, Pin.OUT)  
5  
6 for i in range(3):  
7     # turn on led  
8     led.value(0)  
9  
10    # delay  
11    time.sleep(0.5)  
12  
13    # turn off led  
14    led.value(1)  
15  
16    # delay  
17    time.sleep(0.5)  
18
```

解释执行



microPython
解释器

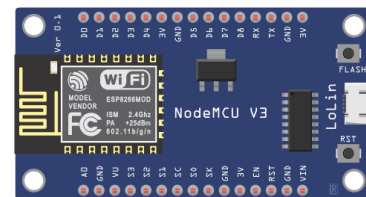


个人电脑

连线



下载



嵌入式控制板



二、软件安装

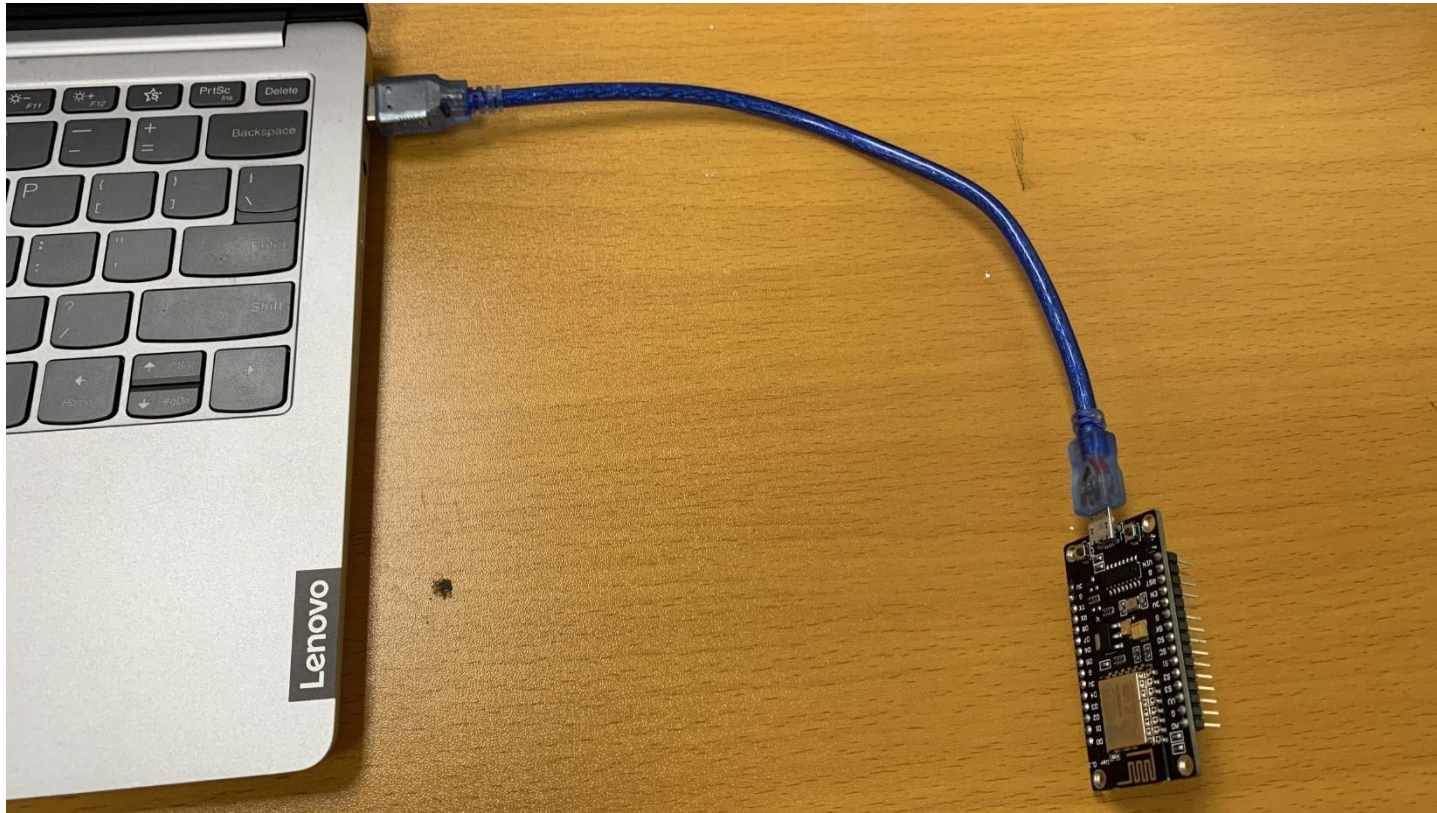


控制板连接电脑

电脑

USB线

esp8266板





查看COM端口 (win10)

- 控制板连接到计算机USB口

Windows 设置

查找设置

系统
显示、声音、通知、电源

设备
蓝牙、打印机、鼠标

网络 and Internet
WLAN、飞行模式、VPN

个性化
背景、锁屏、颜色

帐户
你的帐户、电子邮件、同步设置、工作、家庭

时间和语言
语音、区域、日期

轻松使用

搜索

设备

- 蓝牙和其他设备
- 打印机和扫描仪
- 鼠标
- 触摸板
- 输入
- 笔和 Windows Ink
- 自动播放

蓝牙和其他设备

未连接

LED55K2
未连接

LENOVO-PC: thinkpad:
未连接

NPI15EFC6 (HP LaserJet Professional M1213nf MFP)
驱动程序无法使用

Realtek USB GbE Family Controller
已连接到 USB 3.0

USB-SERIAL CH340 (COM6)

删除设备

康佳电视投屏-KONKA_10009(dlna)
未连接



查看COM端口 (win7)

- 控制板连接到计算机USB口





1. Thonny软件下载

- 软件github网址:
- <https://github.com/thonny/thonny/releases/tag/v3.3.1>

The screenshot shows the GitHub repository page for Thonny. The repository name is 'thonny / thonny'. The page displays the 'Releases' section for version 'v3.3.1'. The 'Assets' section lists 14 files, with 'thonny-3.3.1-windows-portable.zip' highlighted by a purple box. The assets are as follows:

Asset Name	Size
thonny-3.3.1-i686.tar.gz	28.7 MB
thonny-3.3.1-windows-portable.zip	25.5 MB
thonny-3.3.1-x86_64.tar.gz	31 MB
thonny-3.3.1.bash	762 Bytes
thonny-3.3.1.exe	16.7 MB
thonny-3.3.1.pkg	24 MB
thonny-3.3.1_alt.pkg	31 MB
thonny-xxl-3.3.1-i686.tar.gz	105 MB
thonny-xxl-3.3.1-x86_64.tar.gz	113 MB
thonny-xxl-3.3.1.bash	766 Bytes
thonny-xxl-3.3.1.exe	55.5 MB
thonny-xxl-3.3.1.pkg	100 MB
Source code (zip)	
Source code (tar.gz)	



Thonny软件安装

- 下载完成后解压缩
- 进入thonny-3.3.1-windows-portable文件夹
- 找到thonny.exe，双击就可打开使用
- 可右键添加快捷方式至桌面

> thonny-3.3.1-windows-portable

名称

- vcruntime140.dll
- ucrtbase.dll
- thonny_python.ini
- thonny.exe
- README.rst
- pythonw.exe
- python37.dll





设置端口和解释器

- 在Thonny软件中设置端口和解释器
- 选择解释器为MicroPython (ESP8266)

Windows10

查看com编号

蓝牙和其他设备

Realtek USB GbE Family Controller
已连接到 USB 3.0

USB-SERIAL CH340 (COM6)

删除设备

Thonny

文件 编辑 视图 运行 工具 帮助

选择解释器...

运行当前脚本 F5
调试当前脚本(nicer) Ctrl+F5
调试当前脚本(faster) Shift+F5
调试当前脚本(birdseye) Ctrl+Shift+B

Thonny 设置

常规 解释器 编辑器 主题&字体 运行&调试 终端 Shell 助手

Thonny应该使用哪个解释器或设备运行你的代码?

MicroPython (ESP8266)

Thonny的解释器(默认)
可选的python3解释器或虚拟环境
Remote Python 3 (SSH)
MicroPython (SSH)
MicroPython (BBC micro:bit)
MicroPython (ESP32)
MicroPython (ESP266)
MicroPython(一般)
CircuitPython(一般)
特殊的虚拟环境(已弃用)

Port or WebREPL
USB-SERIAL CH340 (COM6)

[Install or update firmware](#)

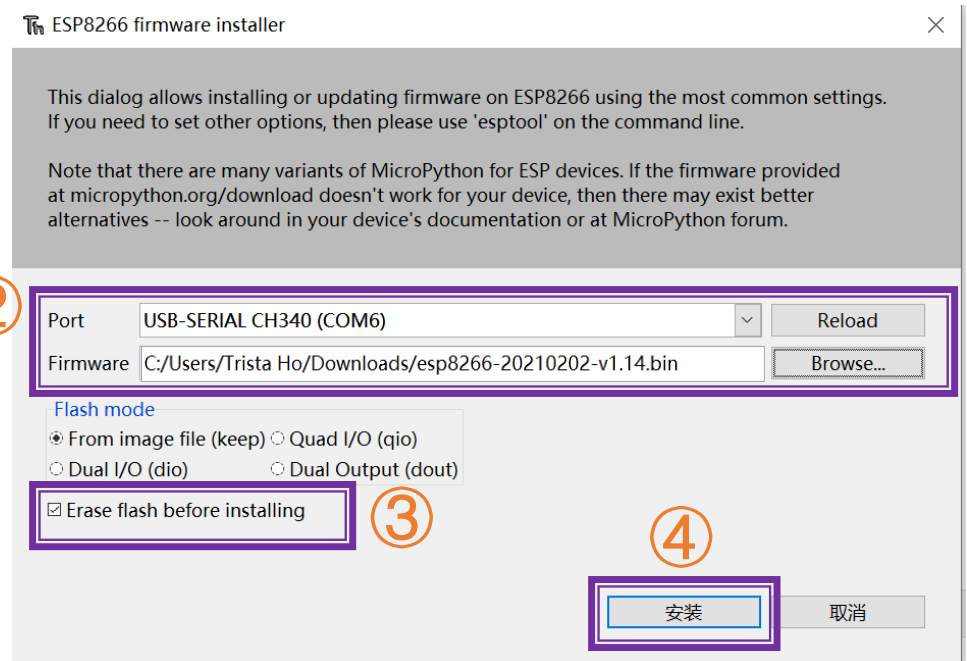
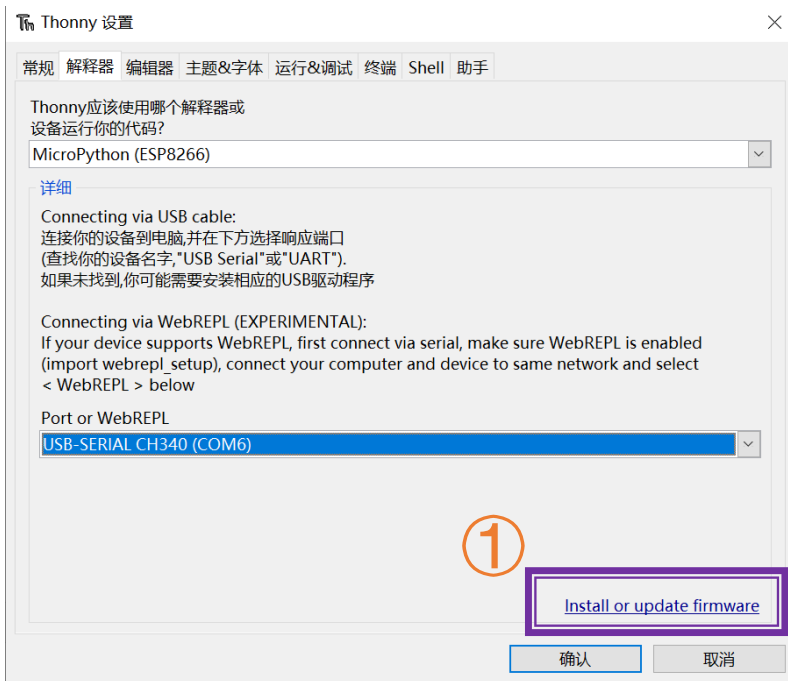
确认

取消



烧写固件

- 选择 install or update firmware
- 选择端口和固件 (esp8266-20210202-v1.14.bin)
- 选择在安装前清空闪存
- 安装，等待固件下载完毕





Hello world

- 可通过终端Shell进行交互式编程

Shell ×

```
MicroPython v1.14 on 2021-02-02; ESP module with ESP8266  
Type "help()" for more information.
```

```
>>> print("Hello World!")
```

```
Hello World!
```

```
>>> a = 1
```

```
>>> b = 2
```

```
>>> c = a+b
```

```
>>> c
```

```
3
```

```
>>> |
```

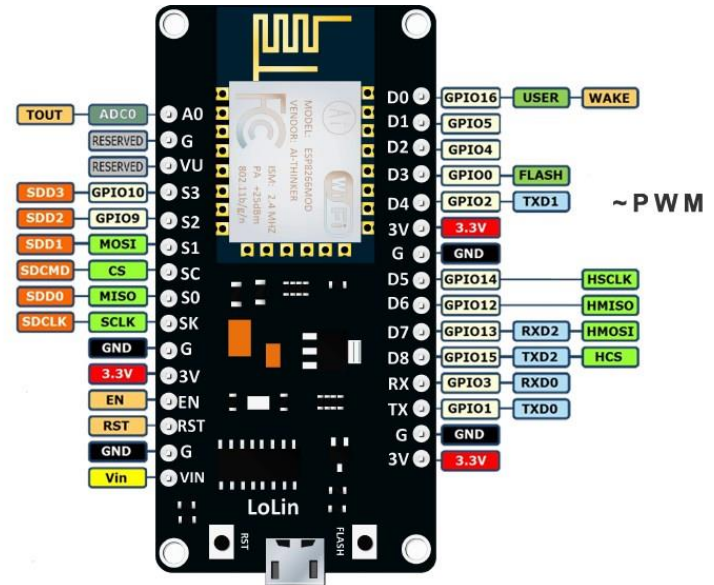


三、简单程序——闪烁的LED



功能实现

- 编写程序控制Led亮1秒，灭1秒，亮1秒，灭1秒.....





交互式编程

- Shell输入

```
>>> from machine import Pin
```

```
>>> import time
```

```
>>> led = Pin(2, Pin.OUT)
```

```
>>> led.value(0)
```

- Shell输入

```
>>> led.value(1)
```



交互式编程



- 循环

```
>>> while True:  
    led.value(0)  
    time.sleep(1)  
    led.value(1)  
    time.sleep(1)
```





LED程序: `main.py`

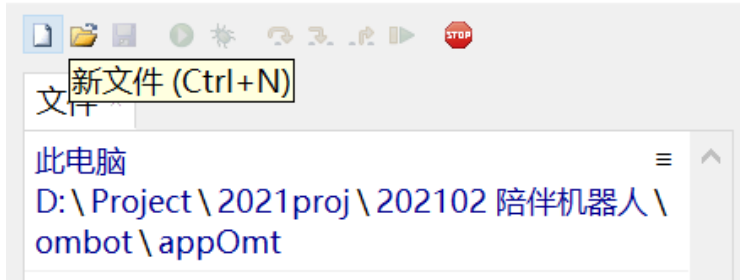
```
1  from machine import Pin
2  import time
3
4  led = Pin(2, Pin.OUT)
5
6  while True:
7      led.value(0) # turn on led
8      time.sleep(1)
9
10     led.value(1) # turn off led
11     time.sleep(1)
```



新建程序

Thonny

文件 编辑 视图 运行 工具 帮助



<untitled> * x

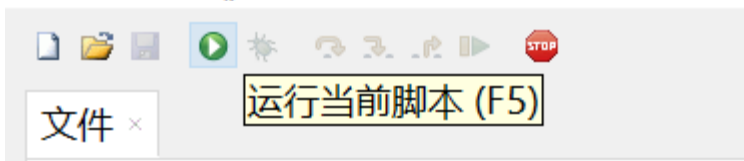
```
1 from machine import Pin
2 import time
3
4 led = Pin(2, Pin.OUT)
5
6 while True:
7     led.value(0) # turn on led
8     time.sleep(1)
9
10    led.value(1) # turn off led
11    time.sleep(1)
```



运行

Thonny - MicroPython设备 :: /main.py

文件 编辑 视图 运行 工具 帮助



```
MicroPython v1.14 on 2021-02-02; ESP module with ESP8266
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
```

Ctrl - C结束

```
>>> %Run -c $EDITOR_CONTENT
Traceback (most recent call last):
  File "<stdin>", line 11, in <module>
KeyboardInterrupt:
>>>
```



上传程序

```
<untitled> * *  
1 from machine import Pin  
2 import time  
3  
4 led = Pin(2, Pin.OUT)  
5  
6 while (True):  
7     led.value(0) # tu  
8     time.sleep(1)  
9  
10    led.value(1) # tu  
11    time.sleep(1)
```

Save to MicroPython设备

Name	Size (bytes)
boot.py	230

保存中 ...

Saving to /main.py

MicroPython设备

- boot.py
- main.py



ESP8266参考手册

http://docs.micropython.01studio.org/zh_CN/latest/esp8266/quickref.html

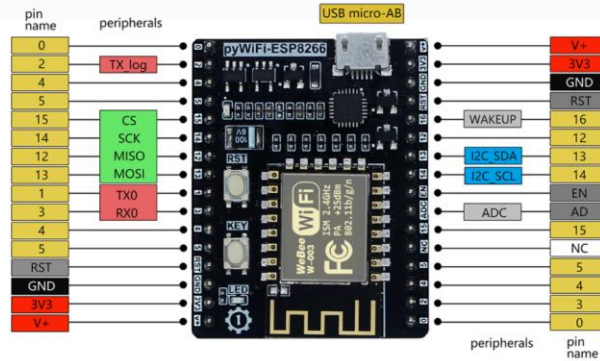
- MicroPython 开发和构建
- MicroPython 许可证信息
- pyboard 快速参考手册
- 哥伦布 快速参考手册
- ESP8266 快速参考手册
 - General information about the ESP8266 port
 - MicroPython tutorial for ESP8266
 - 安装 MicroPython
 - 通用控制
 - Networking
 - 延时和时间
 - 定时器
 - 引脚和GPIO口
 - UART (serial bus)
 - PWM (脉宽调制)
 - ADC (模数转换)
 - 软件SPI总线
 - 硬件SPI总线
 - I2C总线
 - 实时时钟 (RTC)
 - 深度睡眠模式
 - 单总线驱动 (OneWire)
 - NeoPixel 彩灯驱动
 - APA102 驱动
 - DHT 驱动
 - WebREPL (Web浏览器交互提示)
- ESP32 快速参考手册
- WiPy 快速参考手册
- Quick reference for the UNIX and Windows ports
- Versions and Downloads

www.01studio.org/micropython/picture/pyWiFi-ESP8266_pinout.png

Docs » ESP8266 快速参考手册

[Edit on GitHub](#)

ESP8266 快速参考手册



pyWiFi-ESP8266

MicroPython

www.01Studio.org

V+: 3.6-6V输入 (当由USB供电时电压为5V, 建议使用USB口供电)
 3V3: 3.3V输出, 最大电流600mA
 VBAT: 锂电池输入 (板载FET保护电路和锂电池充电电路, XH-2.54 2P接口在背面)

LED: 连接到引脚 2
 KEY: 连接到引脚 0
 I2C: 支持任意IO
 PWM: 支持任意IO (引脚16除外)
 引脚16连接到RST, 可以进入深度睡眠模式。

ESP8266开发板 (图片来源: 01Studio)

以下是快速参考内容, 如果你是第一次使用ESP8266开发板, 请考虑先阅读以下章节内容:

- General information about the ESP8266 port
- MicroPython tutorial for ESP8266

安装 MicroPython



例子

ESP8266 port

MicroPython tutorial for ESP8266

安装 MicroPython

通用控制

Networking

延时和时间

定时器

引脚和GPIO口

UART (serial bus)

PWM (脉宽调制)

ADC (模数转换)

软件SPI总线

硬件SPI总线

I2C总线

实时时钟 (RTC)

深度睡眠模式

单总线驱动 (OneWire)

NeoPixel 彩灯驱动

APA102 驱动

DHT 驱动

WebREPL (Web浏览器交互提示)

ESP32 快速参考手册

WiPy 快速参考手册

Quick reference for the UNIX and Windows ports

引脚和GPIO口

使用 `machine.Pin` 模块:

```
from machine import Pin

p0 = Pin(0, Pin.OUT) # 创建对象p0, 对应GPIO0口输出
p0.on()              # 设置引脚为 "on" (1) 高电平
p0.off()             # 设置引脚为 "off" (0) 低电平
p0.value(1)          # 设置引脚为 "on" (1) 高电平

p2 = Pin(2, Pin.IN)  # 创建对象p2, 对应GPIO2口输入
print(p2.value())    # 获取引脚输入值, 0 (低电平) or 1 (高电平)

p4 = Pin(4, Pin.IN, Pin.PULL_UP) # 打开内部上拉电阻
p5 = Pin(5, Pin.OUT, value=1) # 初始化时候设置引脚的值为 1 (高电平)
```

以下为可用引脚: 0, 1, 2, 3, 4, 5, 12, 13, 14, 15, 16, 分别对应ESP8266芯片的实际GPIO引脚编号。请注意, 很多用户使用自己的开发板有特定的引脚命名方式 (例如: D0, D1, ...)。由于MicroPython致力于支持不同的开发板和模块, 因此我们采用最原始简单且具有共同特征的引脚命名方式。如果你使用自己的开发板, 请参考其原理图。

请注意, 引脚 Pin(1) and Pin(3) 是串口交互 (REPL) UART TX 和 RX 引脚. 同时请注意 Pin(16) 是一个特殊的引脚 (用于从深度睡眠模式中唤醒) 所以有可能不能使用高级的类模块如 ``Neopixel``.

UART (serial bus)

See `machine.UART`.

```
from machine import UART
uart = UART(0, baudrate=9600)
uart.write('hello')
uart.read(5) # read up to 5 bytes
```




四、进阶程序——闪烁的LED



功能实现

- 编写程序控制Led灭灯的时间始终为1秒；
- 亮灯的时间从1秒开始并且以1秒递增，达到3秒后，又回到1秒，以1秒递增……





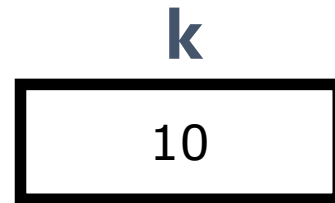
变量

- **变量**是指在程序的运行过程中可以发生变化的量，存放在计算机的内存单元中。
- 它是存放数据的地方，有一个**类型**、一个**名称**和一个**值**。

- **变量初始化**

初始化整型变量例子

$k = 10$





添加变量

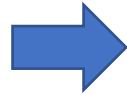
```
from machine import Pin
import time
```

```
led = Pin(2, Pin.OUT)
```

```
while True:
```

```
    led.value(0)
    time.sleep(1)
```

```
    led.value(1)
    time.sleep(1)
```



```
from machine import Pin
import time
```

```
led = Pin(2, Pin.OUT)
```

```
sleeptime = 1 # 初始化亮灯时间变量
```

```
while True:
```

```
    led.value(0)
    time.sleep(1)
```

```
    led.value(1)
    time.sleep(sleeptime) # 使用变量
```



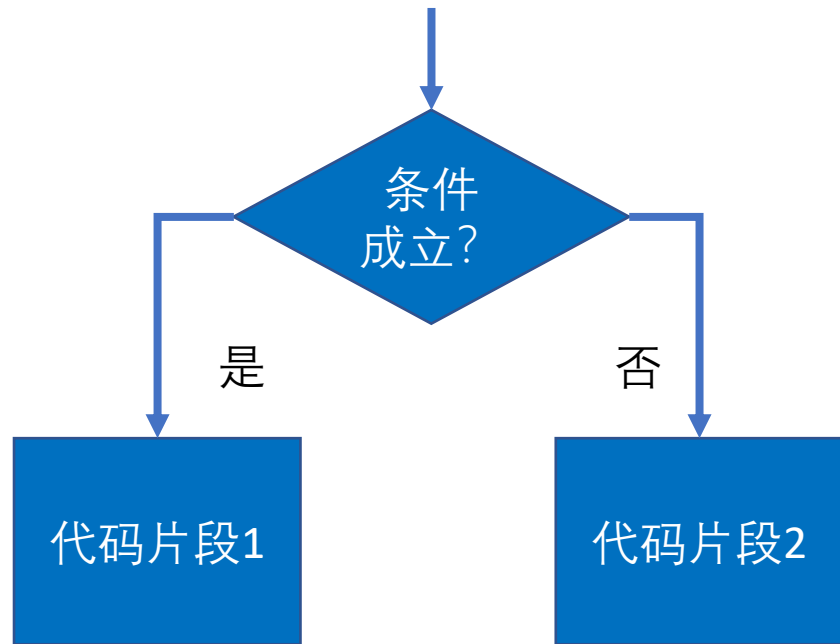
分支指令

if 条件:

代码片段1

else:

代码片段2





条件例子

if 条件成立:

代码片段1

else:

代码片段2

条件例子:

$x == y$

$x != y$

$x < y$

$x > y$

$x \leq y$

$x \geq y$

x 等于y

x 不等于y

x 小于y

x 大于y

x 小等于y

x 大等于y



添加分支指令

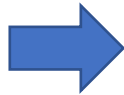
```
from machine import Pin
import time
```

```
led = Pin(2, Pin.OUT)
sleeptime = 1
```

```
while True:
```

```
    led.value(0)
    time.sleep(1)
```

```
    led.value(1)
    time.sleep(sleeptime)
```



```
sleeptime
```

```
1 from machine import Pin
2 import time
```

```
3
```

```
4 led = Pin(2, Pin.OUT)
5 sleeptime = 1
```

```
6
```

```
7 while True:
```

```
8     led.value(0)
9     time.sleep(1)
```

```
10
```

```
11     led.value(1)
12     time.sleep(sleeptime)
```

```
13
```

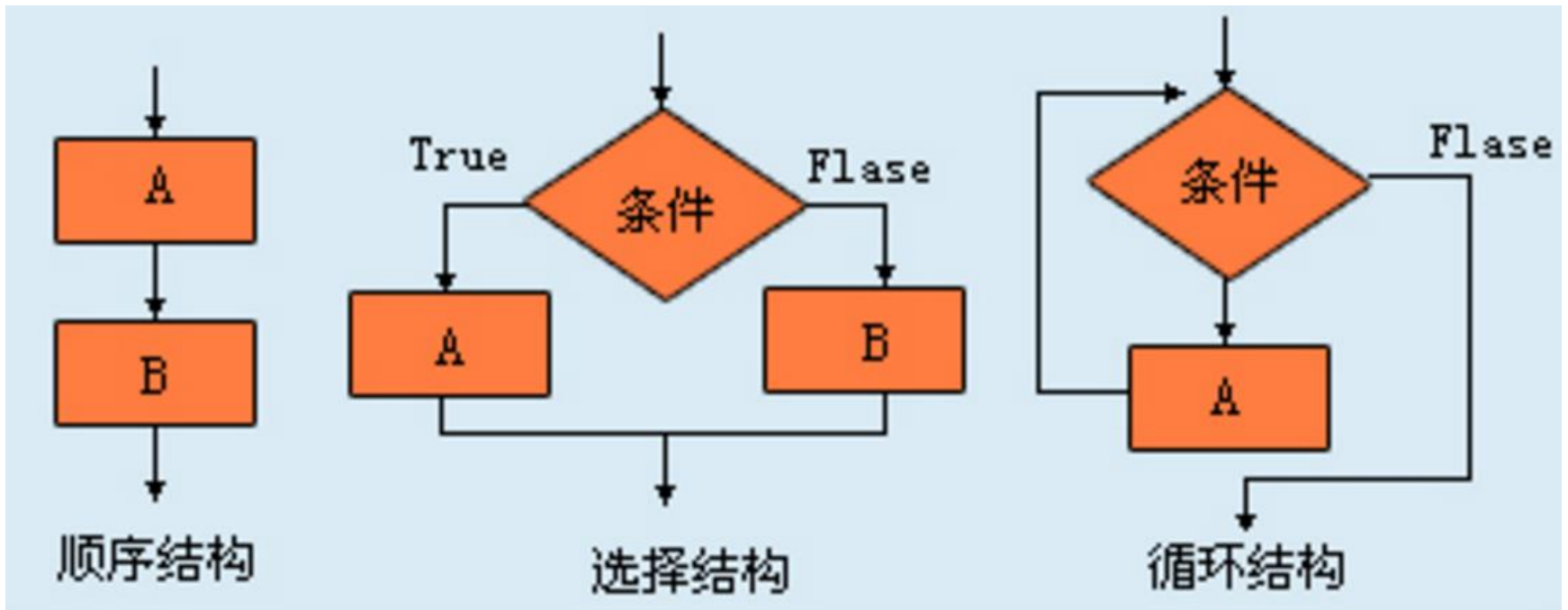
```
14         sleeptime = sleeptime + 1
```

```
15
```

```
16     else:
17         sleeptime = 1
```



程序结构

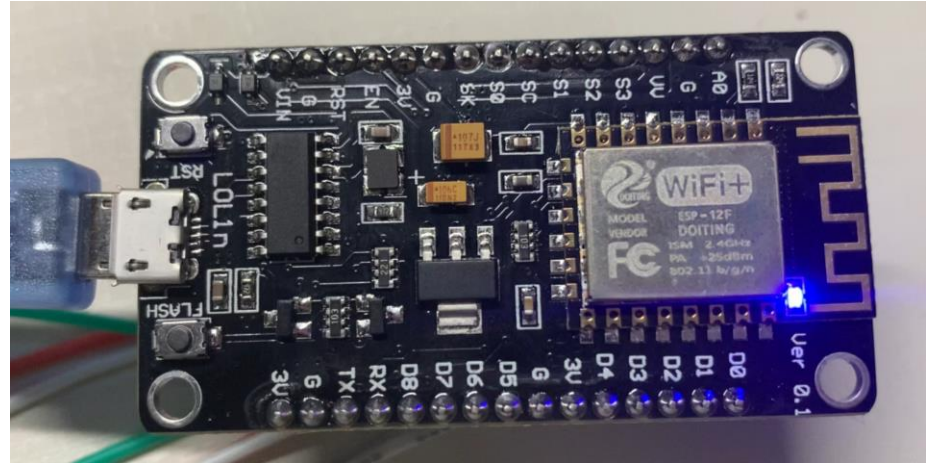




实验2. LED不等间隔闪烁



```
1 from machine import Pin
2 import time
3
4 led = Pin(2, Pin.OUT)
5 sleeptime = 1
6
7 while True:
8     led.value(0)
9     time.sleep(1)
10
11     led.value(1)
12     time.sleep(sleeptime)
13     print("sleeptime: ", sleeptime)
14     if sleeptime < 3:
15         sleeptime = sleeptime + 1
16     else:
17         sleeptime = 1
```



```
Shell x
>>>
>>>
>>>
>>>

Unable to connect to COM6: could not open port 'COM6': OSError:
Backend terminated or disconnected. Use 'Stop/Restart' to re:

MicroPython v1.14 on 2021-02-02; ESP module with ESP8266
Type "help()" for more information.
>>> |
```

谢谢！

