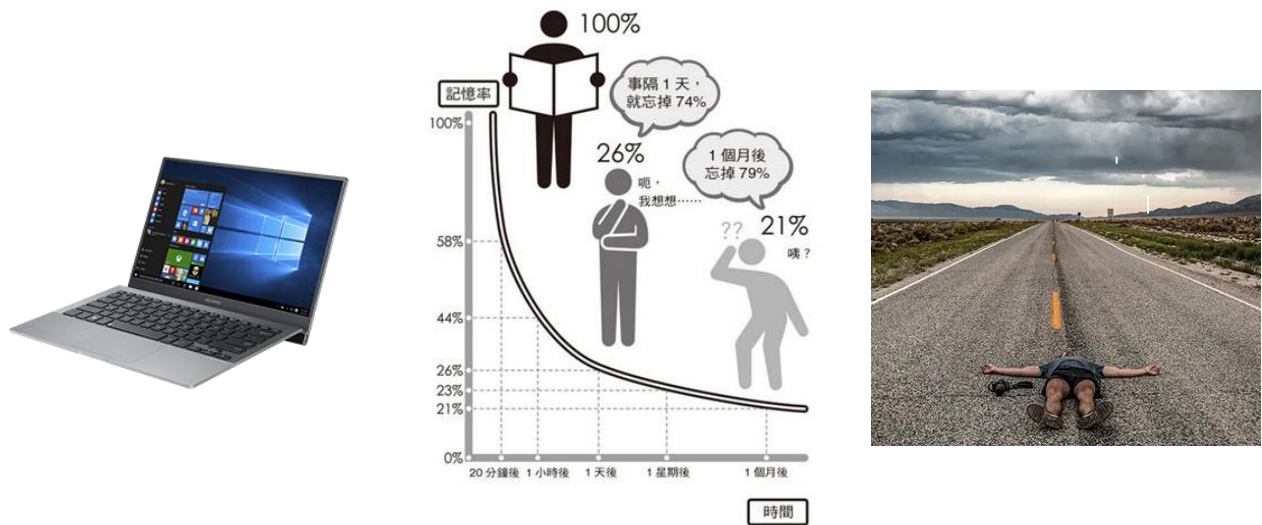


Chapter 1：OLED 顯示中、英文

本章重點

介紹 OLED 顯示中、英文，使用跑馬燈顯示全文，並加入到其他專案中。

準備材料



可上網的電腦

避免記憶遺忘的陷阱

反覆練習、再練習的堅持

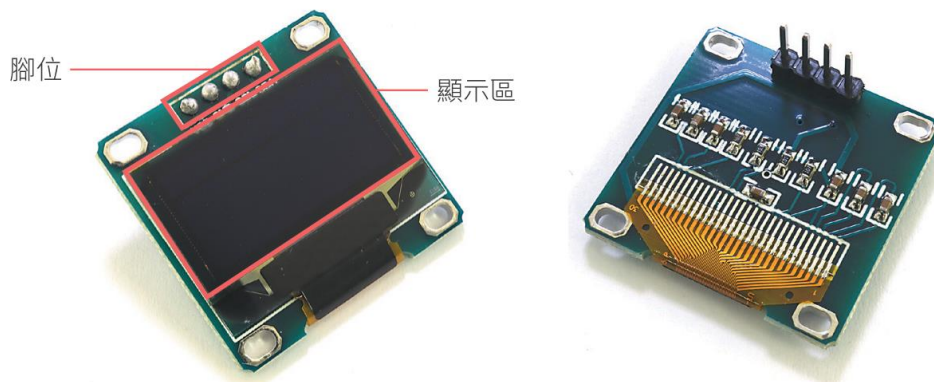
學習目標

1. OLED 顯示中、英文。
2. 將 OLED 加入專案中。

1-1. OLED 顯示中、英文

1-1-1. 認識 OLED 模組

OLED (Organic LED, 有機 LED)，目前普遍用於手機和電視螢幕。這裡使用 0.96 吋 OLED 模組，驅動晶片為 SSD1306，解析度 128x64 像素。



ESP32 必須透過 I2C 通訊協定來控制這種 OLED 模組。兩個裝置之間需要用兩條線來當作通訊的橋梁，這兩條橋的名字分別是：

- SDA (Serial Data Line, holds Data or address signal)，簡單來說就是用來傳輸資料的線路，雙向傳輸，收發都靠這條。
- SCL (Serial Clock Line, holds Clock signal)，用來同步通訊時序的線，這條線的震盪速度就代表傳輸速度。

在 OLED 顯示英文

1-1

OLED 顯示英文。

```
1. from machine import Pin, I2C
2. from ssd1306 import SSD1306_I2C
3.
4. i2c = I2C(scl = Pin(22), sda = Pin(21)) # 初始化 I2C 介面
5. oled = SSD1306_I2C(128, 64, i2c) # 創建一個 SSD1306 顯示物件
6.
7. oled.text('I love PYTHON!', 0, 0) # 在坐標 (0, 0) 處開始顯示
8. oled.show() # 刷新顯示，將內容顯示在 OLED 螢幕上
```

※ MicroPython 內建的 ssd1306 模組只支援英文。

在 OLED 顯示系統時間

1-2

OLED 顯示系統時間。

```
1. from machine import Pin, I2C
2. from ssd1306 import SSD1306_I2C
3. import utime # 引入 utime 庫，用於時間相關的功能
4.
5. i2c = I2C(scl = Pin(22), sda = Pin(21)) # 初始化 I2C 介面
6.
7. oled = SSD1306_I2C(128, 64, i2c) # 創建一個 SSD1306 顯示物件
8.
9. while True:
10.     system_time = utime.ticks_ms() # 獲取當前系統時間（以毫秒為單位）
11.     oled.fill(0) # 清空顯示器的內容
12.     oled.text('System time: ', 0, 0) # 顯示於第一行
13.     oled.text(str(system_time) + ' ms', 0, 16) # 顯示於第三行
14.     oled.show() # 更新顯示器以顯示新內容
15.     utime.sleep_ms(100) # 暫停 100 毫秒，防止過快更新
```

常用小程式：i2c scanner

若找不到 OLED，可先用以下程式掃描 i2c 的設備。

1-3

i2c scanner。

```
1. from machine import Pin, I2C
2.
3. i2c = I2C(scl = Pin(22), sda = Pin(21))
4. print('Scan i2c bus...') # 輸出提示信息，開始掃描 I2C 總線
5. devices = i2c.scan() # 掃描 I2C 總線上的設備，返回找到的設備地址
6.
7. if len(devices) == 0: # 檢查是否找到任何 I2C 設備
8.     print('No i2c device !') # 如果沒有設備，輸出提示
9. else:
10.     print('i2c devices found:', len(devices)) # 如果找到，輸出設備數量
11.
12.     for device in devices: # 列印每個設備的十進制和十六進制地址
```

```
13. print('Decimal address: ', device, ' | Hexa address: ',  
hex(device))
```

1-1-2. OLED 顯示中文

要顯示中文字，需自訂 MicroPython 的字型檔，實現不同字體的顯示，並可以調節字體數量，方便較小空間使用。

可參考以下教學：

網址

<https://www.esp56.com/show/?id=MzY0>

下面介紹教學中，最簡單的使用方式，直接使用已制作好的字體模組 `ufont.py`：

Step 1

下載中文字庫。

網址

<https://github.com/AntonVanke/MicroPython-Chinese-Font>

下載後，將 `demo/ssd1306_demo.py`、`driver/ssd1306.py`、`ufont.py`、`unifont-14-12917-16.v3.bmf` 上傳到開發板根目錄。其中，運行 `ssd1306_demo.py`，可示範語法的使用方式。

Step 2

導入 `ufont` 模組 (`import ufont`)，並撰寫程式。

1-4

OLED 顯示中文。

```
1. from machine import I2C, Pin  
2. from ssd1306 import SSD1306_I2C  
3. import ufont  
4. from time import sleep  
5.  
6. i2c = I2C(scl = Pin(22), sda = Pin(21))  
7. display = SSD1306_I2C(128, 64, i2c)  
8. font = ufont.BMFont('unifont-14-12917-16.v3.bmf')  
9.  
10. # 最簡單的顯示 '你好'  
11. font.text(display, '你好', 0, 0, show = True)  
12. sleep(1)  
13.  
14. # 如果想讓文字顯示在螢幕正中間，可以通過指定文本左上角位置來修改顯示位置
```

```
15. font.text(display, '妳好', 48, 16, show = True)
16. sleep(1)
17.
18. # 上一次顯示顯示的文字不會消失。因為沒有指定清屏參數：'clear=True'
19. font.text(display, 'Hello python! 你好', 24, 16, show = True,
              clear = True)
20. sleep(1)
21.
22. # 中文寬度大概是字母的兩倍，如果你需要等寬，可以指定參數 'half_char=False'
23. font.text(display, 'HELLO', 32, 16, show = True, clear = True,
              half_char = False)
24. sleep(1)
25.
26. # 文字如果很長，超出螢幕邊界，需要指定參數 'auto_wrap=True' 來自動換行
27. words = '他日若遂凌云志，敢笑黃巢不丈夫！'
28. font.text(display, words, 0, 8, show = True, clear = True,
              auto_wrap = True)
29. sleep(1)
30.
31. # 添加 'line_spacing: int' 參數來調整行間距，此處指定 8 個像素
32. font.text(display, words, 0, 8, show = True, clear = True,
              auto_wrap = True, line_spacing = 8)
33. sleep(1)
34.
35. # 調整字體大小，可以指定 'font_size: int' 參數。注意：這會嚴重增加執行時間
36. font.text(display, 'T:15°C', 24, 8, font_size = 32, show = True,
              clear = True)
37. sleep(1)
```

顯示全文的問題

當 OLED 無法顯示全文時，例如：

1-5

OLED 無法顯示全文。

```
1. from machine import SoftI2C, Pin
2. from ssd1306 import SSD1306_I2C
3. import ufont
4. from time import sleep
5.
6. # 使用 SoftI2C 初始化顯示屏
7. i2c = SoftI2C(scl = Pin(22), sda = Pin(21))
8. display = SSD1306_I2C(128, 64, i2c)
9. font = ufont.BMFont('unifont-14-12917-16.v3.bmf')
10.
11. # 定義顯示的文字
12. text = '台湾集成电路制造公司，简称台积电、TSMC、台积或台积公司，与旗下公司
    合称时则称作台积电集团，为台湾一家从事晶圆代工的公司。总部位于台湾新竹科学园
    区。台积电是全球第一家也是最大的集成电路代工制造商，其主要业务包括晶圆制造、
    封装、测试和技术服务。'
13.
14. font.text(display, text, 0, 0, show = True, auto_wrap = True)
```

最佳解決方式：以跑馬燈方式顯示。

1-6

OLED 無法顯示全文。

```
1. from machine import SoftI2C, Pin
2. from ssd1306 import SSD1306_I2C
3. import ufont
4. from time import sleep
5.
6. # 使用 I2C 初始化顯示屏
7. i2c = SoftI2C(scl = Pin(22), sda = Pin(21))
8. display = SSD1306_I2C(128, 64, i2c)
9. font = ufont.BMFont('unifont-14-12917-16.v3.bmf')
10.
11. # 定義顯示的文字
```

```
12. text = '台湾集成电路制造公司，简称台积电、TSMC、台积或台积公司，与旗下公司合
    称时则称作台积电集团，为台湾一家从事晶圆代工的公司。总部位于台湾新竹科学园区。
    台积电是全球第一家也是最大的集成电路代工制造商，其主要业务包括晶圆制造、封装、
    测试和技术服务。'
13.
14. # 設定初始的水平偏移和參數
15. offset = 0
16. display_width = 128 # 顯示屏的寬度
17. scroll_speed = 2 # 控制滾動速度
18.
19. # 估計文字的寬度（例如：每個字元寬度為 8 像素）
20. text_width = len(text) * 8
21.
22. while True:
23.     display.fill(0) # 清空顯示屏
24.     font.text(display, text, -offset, 0) # 根據 offset 向左移動文字
25.     display.show() # 更新顯示屏
26.     sleep(0.05) # 調整延遲以達到平滑或更快的滾動效果
27.
28.     # 更新偏移量
29.     offset += scroll_speed
30.
31.     # 當文字完全滾出顯示屏時，重置偏移量
32.     if offset > text_width:
33.         offset = 0 # 重置以從頭開始顯示文字
```

1-2. 將 OLED 加入專案中

你可以使用以下方式將 OLED 加入到你目前的專案中：

1. 直接動手修改程式碼。
2. 使用 ChatGPT 輔助，最佳的方式為提供相關的程式碼，並使用正確的提示語。

1-2-1. 開關狀態 + OLED

下面是按鈕開關的狀態顯示在螢幕上的程式：

1-7 開關狀態顯示在螢幕。

```
1. from machine import Pin
2. import time
3.
4. led_pin = Pin(5, Pin.OUT)
5. record_switch = Pin(18, Pin.IN, Pin.PULL_UP)
6.
7. while True:
8.     print(record_switch.value())
9.     if record_switch.value() == 0:
10.         led_pin.value(0)
11.     else:
12.         led_pin.value(1)
13.     time.sleep(0.1)
```

下面將按鈕開關的狀態顯示在 OLED 上，以下是使用 ChatGPT 生成出來的程式：

1-8 開關狀態顯示在 OLED。

```
14. from machine import SoftI2C, Pin
15. from ssd1306 import SSD1306_I2C
16. import ufont
17. from time import sleep
18.
19. # 初始化 I2C 和顯示屏
20. i2c = SoftI2C(scl = Pin(22), sda = Pin(21))
21. display = SSD1306_I2C(128, 64, i2c)
22. font = ufont.BMFont('unifont-14-12917-16.v3.bmf')
23.
```



```

24. # 定義 LED 和開關
25. led_pin = Pin(5, Pin.OUT)
26. record_switch = Pin(18, Pin.IN, Pin.PULL_UP)
27.
28. while True:
29.     # 讀取開關狀態
30.     switch_state = record_switch.value()
31.
32.     # 控制 LED
33.     if switch_state == 0:
34.         led_pin.value(0)    # 開關按下時 LED 關閉
35.     else:
36.         led_pin.value(1)    # 開關未按下時 LED 開啟
37.
38.     # 顯示開關狀態
39.     display.fill(0)    # 清空顯示屏
40.
41.     # 根據開關狀態顯示不同的文字
42.     if switch_state == 0:
43.         status_text = '开关状态: ON'
44.     else:
45.         status_text = '开关状态: OFF'
46.
47.     # 在 OLED 上顯示開關狀態
48.     font.text(display, status_text, 0, 0)
49.     display.show()    # 更新顯示屏
50.
51.     # 延遲一段時間
52.     sleep(0.1)

```