OLED 圖形顯示器

前言

OLED (Organic Light Emitting Diode)，有機發光二極體，是一種顯示技術，利用有機化合物在電流驅動下自動發光來顯示圖像和文字。由於不需要背光源，所以 OLED 顯示器可以做到很薄且省電。可實現更高的對比度和更深的黑色。此 OLED 基於 SSD1306 驅動晶片，是使用 I2C 的 OLED 顯示器。

硬體連接

1. 材料
   - 母對母跳線 X 4 條
   - I2C OLED 顯示器(128x64 像素) X 1
2. 連接至 PI

   

I2C OLED	Raspberry Pi
VCC	5V
GND	GND
SDA	GPIO2(SDA)
SCL	GPIO3(SCL)

3. 啟用 I2C 介面

   sudo raspi-config

   選擇 Interface Options → I2C → 啟用（Enable）。

   重新啟動樹莓派： sudo reboot
4. 掃描連接 I2C 的裝置

   sudo i2cdetect -y 1

- `y`: 表示關閉互動模式，不顯示警告訊息。
- `1`: 表示檢查所有可用的 I2C 接腳(I2C 的匯流排編號)

執行結果，可以看到 `OLED` 的 `I2C` 位址為: `0x3c`。

5. 安裝套件
   - 要在 `Raspberry Pi` 中使用 `OLED` 顯示器，可以安裝 `adafruit-circuitpython-ssd1306` 套件，指令如下:

     `pip install adafruit-circuitpython-ssd1306`
   - 此套件會使用到 `Python` 影像函式庫 `PIL` 及 `Numpy` 模組，若還未安裝 `PIL` 函式庫，安裝指令如下:

     ```
     pip install Pillow
     ```

使用方法

1. 使用 `adafruit-ssd1306` 套件的方法
   - 我們需要初始化，設定 `OLED` 及其 `I2C` 位址，程式如下:

     ```
     disp = adafruit_ssd1306.SSD1306_I2C(128, 64, i2c, addr=0x3c)
     ```

   - 若要將 `OLED` 設為空白顯示，程式如下:

     ```
     disp.fill(0)
     disp.show()
     ```

   - 若要設定繪製的圖形為黑白(`1-bit`)格式，且 `OLED` 的寬度及高度為 `width` 及 `height`，程式如下:

     ```
     image = Image.new("1", (width, height))
     draw = ImageDraw.Draw(image)
     ```

   - 要在 `OLED` 上顯示文字，我們先使用 `draw.rectangle()` 函式，將 `OLED` 設為空白(黑色)，再使用 `draw.text()` 函式，將文字以白色顯示出來，程式如下:

     ```
     draw.rectangle((0, 0, width, height), outline=0, fill=0)
     draw.text((0, 0), '顯示文字', font=字型, fill=255)
     ```

範例: 在 OLED 顯示器上顯示系統時間及日期

```
# 匯入 Raspberry Pi 的 I2C 控制模組
import board  # 提供 I2C 介面初始化
# 匯入 Adafruit SSD1306 OLED 顯示器驅動程式
import adafruit_ssd1306  # 控制 SSD1306 OLED 顯示器
# 匯入 PIL 影像處理相關模組
from PIL import Image, ImageDraw, ImageFont  # 用於建立影像、繪圖與字型
# 匯入 sleep 函式以便延遲
from time import sleep  # 用於主迴圈延遲
# 匯入 datetime 以取得目前時間
from datetime import datetime  # 取得系統時間

# 初始化 I2C 介面，連接到 Raspberry Pi 的預設 I2C 腳位
i2c = board.I2C()  # 建立 I2C 物件
# 初始化 SSD1306 OLED 顯示器，設定解析度與 I2C 位址
# 128x64 為螢幕解析度，addr=0x3c 為常見 I2C 位址
# 若使用其他型號請確認驅動與參數
# 注意：此程式僅適用於 SSD1306，若為 SSD1300 請更換驅動

disp = adafruit_ssd1306.SSD1306_I2C(128, 64, i2c, addr=0x3c)  # 初始化 OLED 顯示器

# 載入字型檔案，18pt 與 30pt 分別用於小字與大字
# 請確保 'FreeSans.ttf' 字型檔案存在於執行目錄
small_font = ImageFont.truetype('FreeSans.ttf', 18)  # 小字型
large_font = ImageFont.truetype('FreeSans.ttf', 30)  # 大字型

# 清空螢幕內容（全黑）
disp.fill(0)  # 填滿螢幕為黑色
# 將清空後的畫面顯示出來
disp.show()  # 更新顯示器內容

# 取得螢幕寬度與高度，方便後續繪圖定位
width = disp.width  # 螢幕寬度（像素）
height = disp.height  # 螢幕高度（像素）
print(width, height)  # 印出螢幕尺寸供除錯
# 建立一個 1-bit（黑白）影像物件，作為畫布
```

```
image = Image.new('1', (width, height))  # 1: 黑白模式
# 建立繪圖物件，可在 image 上繪製圖形與文字
draw = ImageDraw.Draw(image)  # 建立繪圖介面

# 定義顯示訊息的函式，顯示兩行文字
def disp_mess(top_line, line_2):
    # 先清空畫布（填滿黑色）
    draw.rectangle((0, 0, width, height), outline=0, fill=0)  # 清除前一畫面
    # 在畫布上方顯示第一行大字
    draw.text((0, 0), top_line, font=large_font, fill=255)  # 顯示大字
    # 在畫布下方顯示第二行小字
    draw.text((0, 40), line_2, font=small_font, fill=255)  # 顯示小字
    # 將畫布內容傳送到 OLED 顯示器
    disp.image(image)  # 設定顯示內容
    disp.show()  # 更新顯示器

# 主程式迴圈，每秒更新一次時間與日期
while True:
    # 取得目前系統時間
    now = datetime.now()  # 取得現在時間
    # 格式化日期字串（年 月 日）
    date_mess = now.strftime("%Y %m %d")  # 日期字串
    # 格式化時間字串（時:分:秒）
    time_mess = now.strftime("%H:%M:%S")  # 時間字串
    # 印出目前日期與時間（除錯用）
    print(date_mess, time_mess)  # 印出資訊
    # 在 OLED 顯示器上顯示時間與日期
    disp_mess(time_mess, date_mess)  # 顯示於螢幕
    # 暫停 1 秒後再更新
    sleep(1)  # 每秒更新一次
```

執行後，每隔 1 秒會在 `OLED` 顯示 `Raspberry Pi` 目前的系統時間及日期。