
第四版前言

2011 年發表後，Raspberry Pi 被發現可作為極低成本的 Linux 電腦及嵌入式運算平台。它受到教育工作者和業餘愛好者歡迎。

執筆此時，Raspberry Pi 已售出超過四千萬台。配置 8GB 記憶體之 Raspberry Pi 4 的效能已足以取代桌上型電腦，而內建鍵盤的 Pi 400 頗具取代桌上型電腦的能力。

用於瀏覽網路、email、辦公室軟體和相片編輯之開源 Linux 軟體的可用性，使 Raspberry Pi 更加流行。

連最新的 Raspberry Pi 4 和 Pi 400 都保有讓業餘愛好者可加入自己電子裝置的通用輸入／輸出接腳（GPIO）。

本版已徹底更新，並涵蓋了 Raspberry Pi 最新型號以及 Raspberry Pi OS 的許多變更與改進。尤其你會發現有新章節：

- 機器學習
- Raspberry Pi Pico 與 Pico W

本書設計讓你可以像一般書籍一樣連續閱讀，或隨機閱讀主題。你可以在目錄或索引搜尋想看的主題，直接跳去看。如果主題需要瞭解其他知識，本書也會引導你到其他主題，就像食譜在示範如何進一步烹飪前可能會先引導你至基礎醬汁的製作。

Raspberry Pi 的世界變化快速。一直都有大型且活躍的社群、新的介面板和軟體程式庫在發展。除了使用特定介面板或軟體的範例，本書也涵蓋基礎原則，讓你對如何使用 Raspberry Pi 生態系發展的新技術有更好的瞭解。

如您所期，本書伴隨大量程式碼（主要是 Python 程式）。這些程式都是開放原始碼並可於 Github（<https://oreil.ly/fEB8p>）取得。對於大部分基於軟體的主題，你只需要一塊 Raspberry Pi。對此，我建議使用 Raspberry Pi 3 或 4 model B。當主題牽涉到製作你自己的硬體和 Raspberry Pi 連接時，我會試著利用現成模組、麵包板和跳線來避免焊接。

如果你想讓麵包板專案更堅固，我建議使用麵包板一半大小且布局相同的洞洞板，例如 Adafruit 等商家賣的板子，以使設計更容易轉換為焊接方案。

使用本書

本書採用食譜書風格寫作，這意味著你不需要按照順序從頭讀到尾。本書是由個別訣竅集結成章。當訣竅需要先前提過的知識時，會將你引導至該訣竅。

當你試著完成 Raspberry Pi 專案時，你可能會發現，這需要在不同訣竅間跳來跳去。

我在本書規劃了一些我認為對不同類型讀者有用的路徑：

Raspberry Pi 初學者

請閱讀第 1 至 3 章，特別是從訣竅 1.1、1.2 和 1.4 開始，其他則隨興閱讀。

Python 學習者

如果你要用 Raspberry Pi 學習如何用 Python 寫程式，請閱讀第 4 至 7 章。你或許會發現還需要跳到前幾章的訣竅。

業餘電子愛好者

如果你還不會 Python，你需要從第 4 至 7 章學習一些技巧，然後讀完第 8 章和第 9 章後，才挑選後面章節的有趣方案以開始製作自己的 Raspberry Pi 電子專案。

本書編排慣例

本書具有如下的字型慣例：

斜體字 (*Italic*)

指出新名詞、網址 (URL)、email 位址、檔案名稱以及副檔名。中文使用楷體字。

感測器

14.0 簡介

本章我們會看看使用各種不同類型的感測器，讓 Raspberry Pi 能測量溫度、光線等性質的訣竅。

不像 Arduino 和 Raspberry Pi Pico 那樣的板子，一般的 Raspberry Pi 缺少類比輸入。這表示對於許多感測器來說，需要使用額外的類比數位轉換器（ADC）硬體。幸運的是，這蠻容易做到的。也能使用有電容器和一對電阻的電阻式感測器。

許多訣竅需要用免焊接麵包板和公對母跳線（參閱訣竅 10.9）。

14.1 使用電阻式感測器

問題

你要連接可變電阻到 Raspberry Pi，測量電阻以在 Python 程式中決定可變電阻旋鈕的位置。

解決方案

你能在 Raspberry Pi 上只用電容器、一對電阻和兩個通用輸入 / 輸出 (GPIO) 針腳測量電阻。在此例中，你將藉由測量微型可變電阻 (trimpot) 滑片接點兩端的電阻值來評估它的旋鈕位置。

要完成此訣竅，需要以下材料：

- 麵包板和跳線 (參閱第 596 頁的「原型設備與套件」小節)
- 10k Ω 微型可變電阻 (參閱第 597 頁的「電阻與電容器」小節)
- 兩個 1k Ω 電阻 (參閱第 597 頁的「電阻與電容器」小節)
- 330 nF 電容器 (參閱第 597 頁的「電阻與電容器」小節)

圖 14-1 圖示麵包板上零組件的排列。

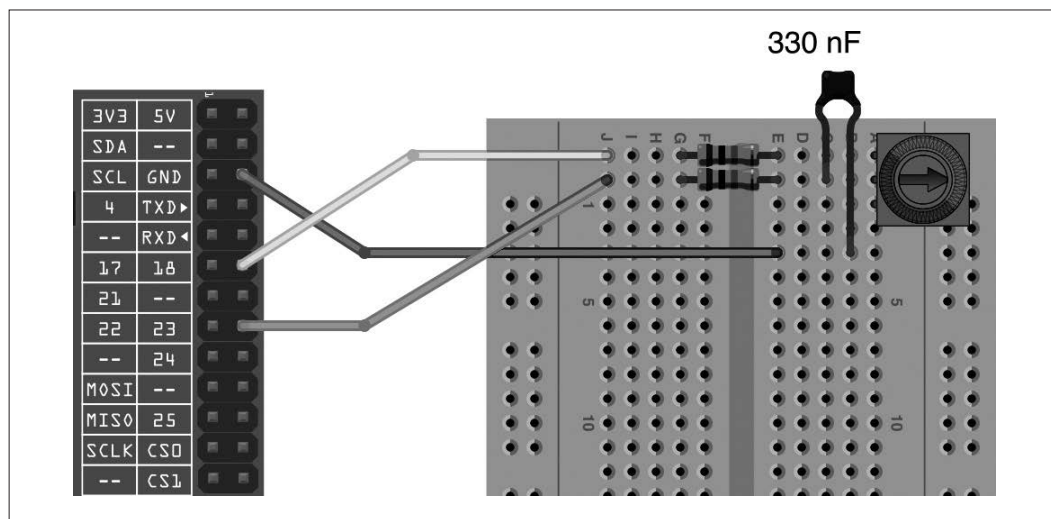


圖 14-1 在 Raspberry Pi 測量電阻值

本訣竅使用一個 Python 程式庫，其作者為了讓類比感測器易於使用而開發。要安裝它，請執行下列指令：

```
$ git clone https://github.com/simonmonk/pi_analog.git
$ cd pi_analog
$ sudo python3 setup.py install
```

開啟編輯器並貼上以下程式碼 (`ch_14_resistance_meter.py`)：

```
from PiAnalog import *
import time

p = PiAnalog()

while True:
    print(p.read_resistance())
    time.sleep(1)
```

如本書所有程式範例一樣，你可以下載本程式（請參閱訣竅 3.22）。

執行此程式時，你應該會看到像這樣的輸出結果：

```
$ python3 ch_14_resistance_meter.py
5588.419502667787
5670.842306126099
8581.313103654076
10167.614271851775
8724.539614581638
4179.124682880563
267.41950235897957
```

讀數會根據轉動可變電阻旋鈕的位置而有不同。理想上，電阻值讀數會在 0 和 10,000 Ω 之間變動，但是實務上，會有一些誤差。

討論

要解釋 `PiAnalog` 類別的原理，我需要先解釋測量可變電阻電阻值的步階響應 (*step response*) 技術的原理。

圖 14-2 示範此訣竅的電路圖。

這麼做的方法稱為步階響應法，因為它是看輸出由低電位到高電位切換的步階改變之電路響應來運作。

可以將電容器想成一個電力的容器，當它充滿電荷時，兩端的電壓會增加。因為 `Raspberry Pi` 沒有類比數位轉換器 (ADC)，所以你無法直接測量電壓。不過你可以計算電容器充滿電荷至高於 1.65V，或達到高電位數位輸入所需的時間。電容器充滿電荷的速度取決於可變電阻值 (R_t)。電阻值越低，電容器充滿電荷和電壓上升得越快。

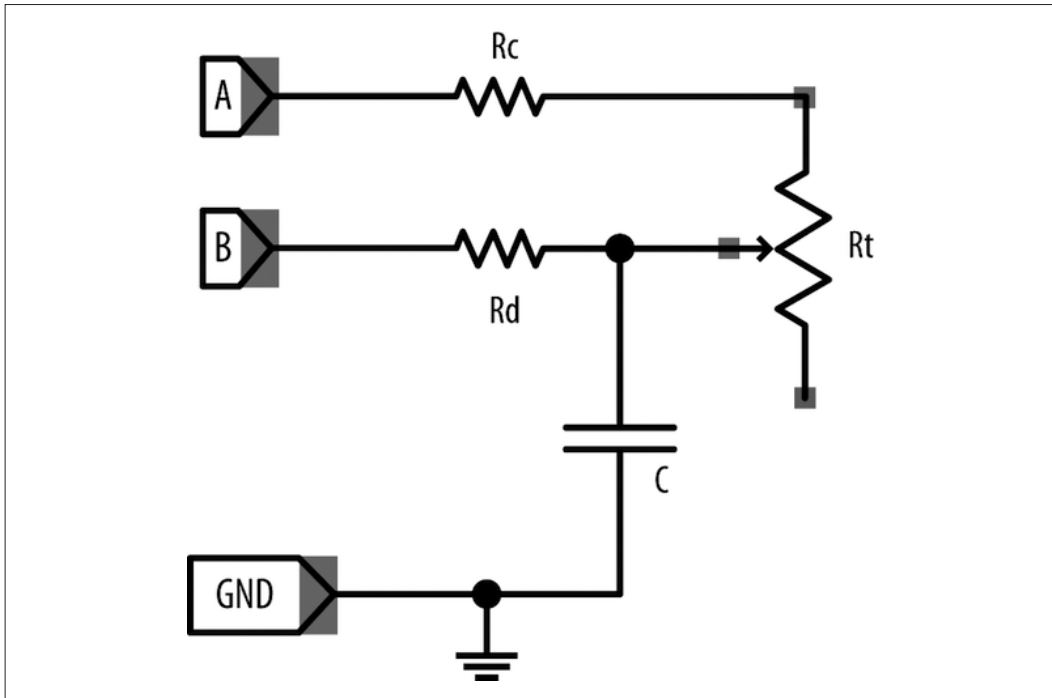


圖 14-2 使用步階響應法測量電阻值

要取得好的讀數，你也必須於每次讀取讀數前清空電容器。圖 14-2 中，A 接線經由 R_c 和 R_t 將電容器充電，B 接線則經由 R_d 來將電容器放電（清空）。電阻 R_c 和 R_d 用以防止電容器充放電時過多電流流經 Raspberry Pi 相對脆弱的 GPIO 針腳。

涉及讀取讀數的步階會先由 R_d 將電容器放電，然後經由 R_c 和 R_t 將它充電。要放電，接線 A（GPIO 18）會被設定為輸入，有效地斷開電路中的 R_c 和 R_t 。接線 B（GPIO 23）則會被設為低電位輸出。會維持該狀態 100 毫秒以清空電容器。

現在電容器已清空，你可以開始設定接線 B 為輸入（有效地斷開電路），然後設定接線 A 為高電位 3.3V 輸出以讓電荷流入。電容器 C 現在會開始經由 R_c 和 R_t 充電。

圖 14-3 圖示電阻和電容器在這樣的排列下，於電壓高低電位切換時的充放電。

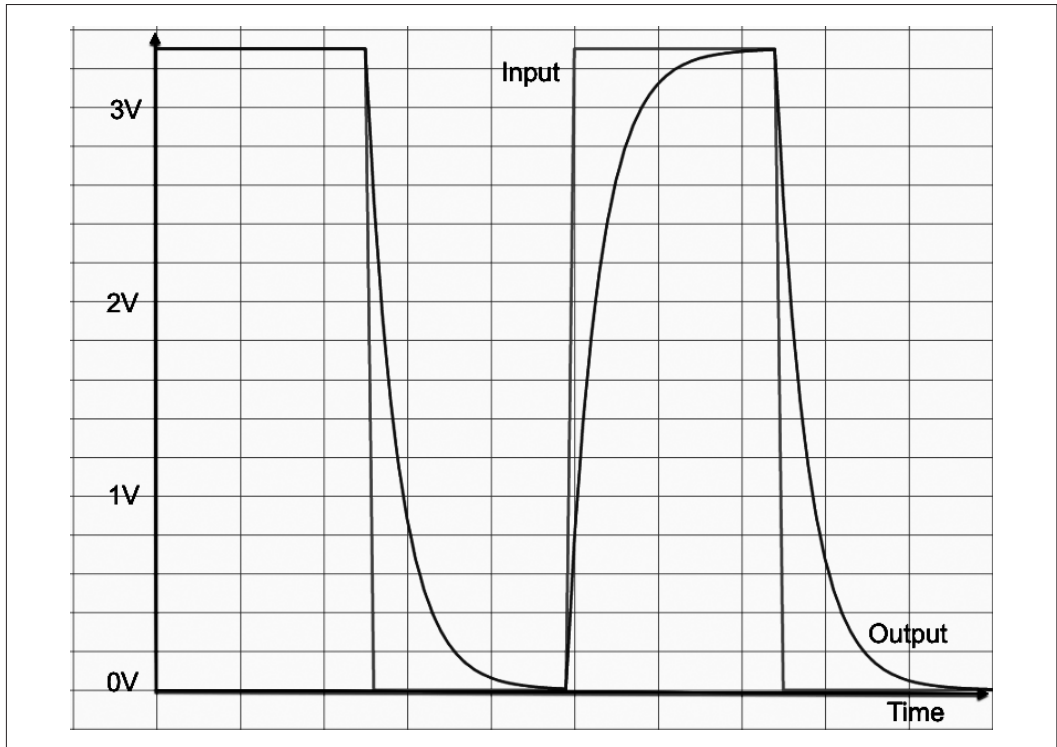


圖 14-3 電容器的充放電

你可以看到電容器的電壓先快速增加，但隨後當電容器滿了之後逐漸降低。幸運的是，你有興趣的是曲線向上直到電容器達到約 1.65V 的部分，它是相當直的線，表示電容器電壓上升到此點所花的時間粗略地和 R_t 的電阻值即旋鈕的位置成比例。

這種方法不太精確，但是成本很低且易用。不準確的主要原因是合適值的電容器準確度僅為 10%。

參閱

使用步階響應於各種光線（訣竅 14.2）、溫度（訣竅 14.3），甚至氣體（訣竅 14.4）偵測的電阻式感測器都運作得不錯。

對於更正確地微型可變電阻位置測量，請參閱訣竅 14.7，可變電阻會和 ADC 一起使用。

14.5 測量空氣品質

問題

二氧化碳（CO₂）濃度是空氣品質指標。你想用 Raspberry Pi 以特別的感測器測量空氣品質。

解決方案

請用低成本（好吧，相對低成本）的 MH-Z14A CO₂ 感測器模組，圖 14-10 示範它連接於 Raspberry Pi。

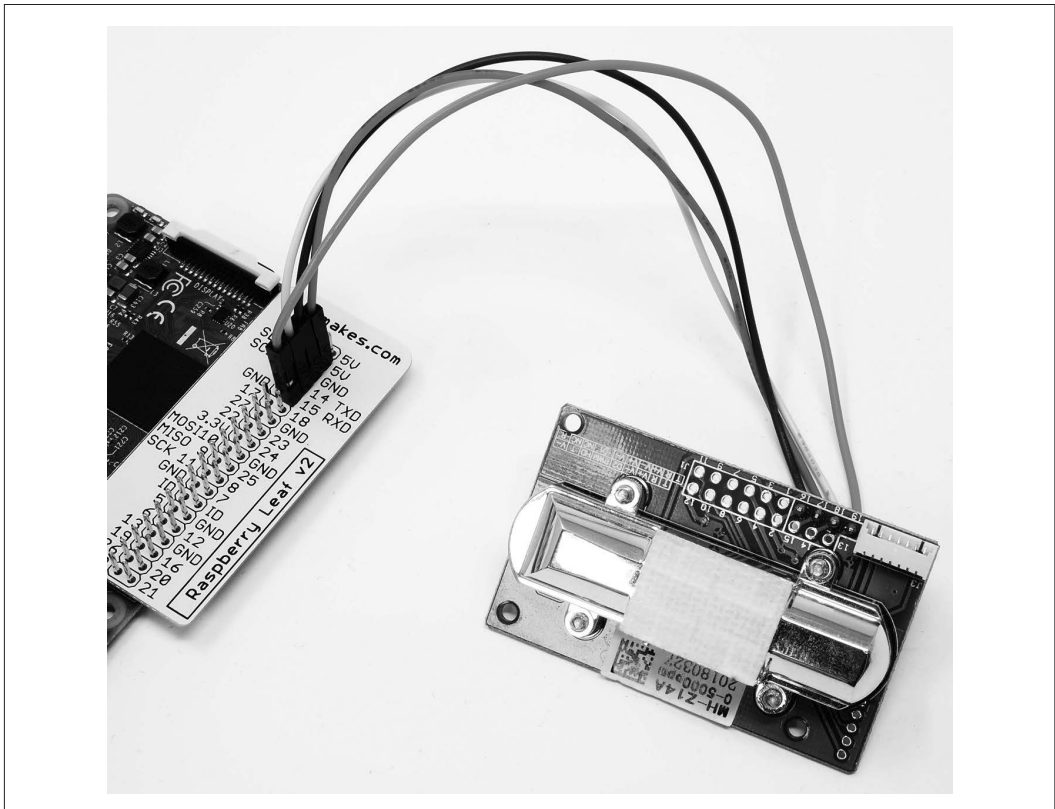


圖 14-10 連接至 Raspberry Pi 的 MH-Z14A CO₂ 感測器

要製作此訣竅，你需要以下材料：

- 一個 MH-Z14A CO₂ 感測器模組（參閱第 598 頁的「模組」小節）
- 母對母跳線（參閱第 596 頁的「原型設備與套件」小節）

將 Z14A 感測器以下列方式和 Raspberry Pi 連接：

- MH-Z14A 針腳 16 對 Raspberry Pi 的 GND
- MH-Z14A 針腳 17 對 Raspberry Pi 的 5V
- MH-Z14A 針腳 18 對 Raspberry Pi 的 GPIO 14（TXD）
- MH-Z14A 針腳 19 對 Raspberry Pi 的 GPIO 15（RXD）

此感測器使用序列埠。這表示要讓它運作，你必須遵循訣竅 2.6。請注意雖然需要連接序列埠硬體，但是你不該啟用 `serial console` 選項。

下列測試程式會從感測器讀取 CO₂ 的量，並每秒回報一次（`ch_14_co2.py`）：

```
import serial, time

request_reading = bytes([0xFF, 0x01, 0x86, 0x00, 0x00, 0x00,
                        0x00, 0x00, 0x79])

def read_co2():
    sensor.write(request_reading)
    time.sleep(0.1)
    raw_data = sensor.read(9)
    high = raw_data[2]
    low = raw_data[3]
    return high * 256 + low;

sensor = serial.Serial('/dev/serial0')
print(sensor.name)
if sensor.is_open:
    print("Open")

while True:
    print("CO2 (ppm):" + str(read_co2()))
    time.sleep(1)
```

如本書所有程式範例一樣，你可以下載本程式（請參閱訣竅 3.22）。

程式執行時，CO₂ 的量應該（除非你在很小又不通風的房間）會是大約 400ppm。如果你對感測器呼吸數秒鐘，讀數會慢慢開始升高，然後接下來幾分鐘會再回到正常讀數：

```
$ python3 ch_14_co2.py
/dev/ttyS0
Open
CO2 (ppm):489
CO2 (ppm):483
CO2 (ppm):483
CO2 (ppm):481
CO2 (ppm):491
CO2 (ppm):517
CO2 (ppm):619
CO2 (ppm):734
CO2 (ppm):896
CO2 (ppm):1367
```

感測器使用請求／回應通訊協定。所以當你要從感測器接收讀數時，需要先送出含在 `request_reading` 中的 9 位元組訊息。感測器會立刻以 9 位元組訊息回應。我們感興趣的只有位元組 2 和 3，它包含以 ppm 為單位之 CO₂ 讀數的高低位元組。

討論

正常 CO₂ 的量大約是 400 到 1000 ppm。高於此就會開始感覺空氣混濁，你可能會昏昏欲睡。研究顯示通風不良導致的高 CO₂ 濃度會降低心智表現。將程式徹夜執行後，我現在都會將臥室門窗微開。

參閱

您可學習更多關於 Z14A 協定的資訊（<https://oreil.ly/Zc6is>）。

你可以找到更多 CO₂ 安全濃度的相關資訊（<https://oreil.ly/h0QGj>）。

14.6 測量土壤濕度

問題

你要測量植物周圍的土壤濕度。

解決方案

請使用 MonkMakes 植物監控板（MonkMakes Plant Monitor board）。不像其他大多數的土壤監控板，這塊板子有序列介面，所以你可以先遵照訣竅 2.6 啟用序列介面。請如圖 14-11 所示連接板子。

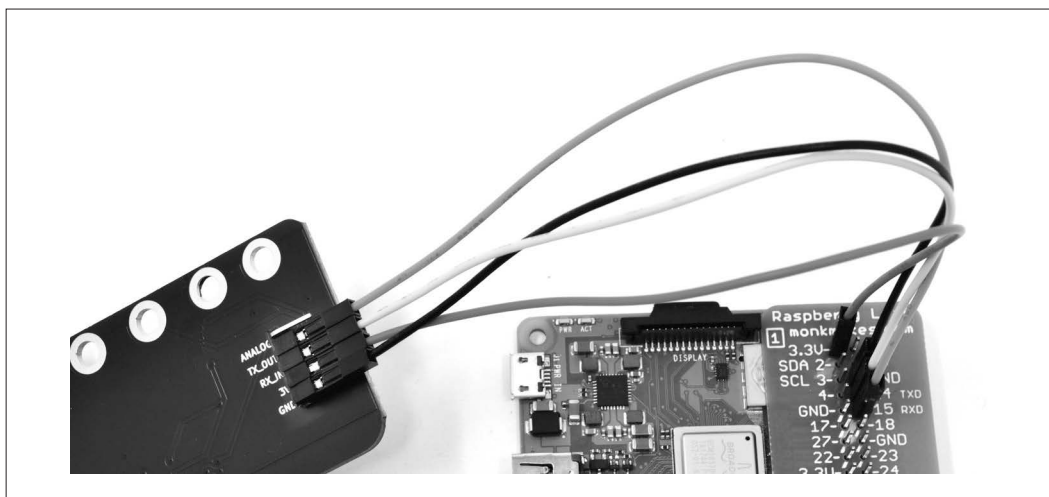


圖 14-11 連上 Raspberry Pi 的植物監控板

此連線有：

- GND 對 GND
- Raspberry Pi 的 3.3V 對植物監控板的 3V
- Raspberry Pi 的 14 TXD 對植物監控板的 RX_IN
- Raspberry Pi 的 15 RXD 對植物監控板的 TX_OUT

再使用此指令下載板子用的 Python 軟體：

```
$ git clone https://github.com/monkmakes/pmon.git
```

這是另一個使用 `guizero` 的程式，所以如果你尚未安裝 `guizero`，請使用以下指令安裝：

```
$ pip3 install guizero
```

然後切換至範例目錄，並以下列指令執行範例程式：

```
$ cd pmon/raspberry_pi
$ python3 01_meter.py
```

這會開啟如圖 14-12 所示的視窗：

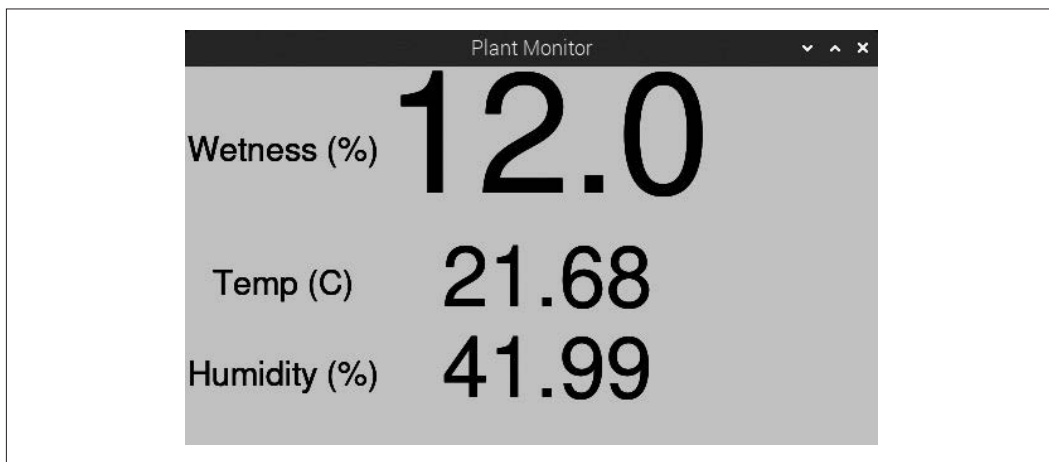


圖 14-12 以 Raspberry Pi 監控土壤水分、溫度和濕度

討論

大部分土壤濕度感測器需要類比輸入，所以無法直接連接 Raspberry Pi。MonkMakes 植物監控板使用序列介面，讓它很適合連接 Raspberry Pi。它會回報植物的溫度和濕度，也有 RGB LED 指示土壤水分。

前一個範例程式的程式碼如下：

```
import threading
import time
from guizero import App, Text
from plant_monitor import PlantMonitor

pm = PlantMonitor()

app = App(title="Plant Monitor", width=550, height=300, layout="grid")

def update_readings(): # 以新的溫度和 CO2 讀數更新欄位
    while True:
        wetness_field.value = str(pm.get_wetness())
```