

示波器自動化和 Python 入門

技術簡介



目前已有許多產業的工程師使用自動化來擴展其示波器的功能，並且許多工程師也正在使用 Python。免費的程式設計語言 Python 在自動化方面具有顯著優勢：

- 多功能性
- 易教易學
- 程式碼可讀性
- 廣泛可用的知識庫和模組

自動化有兩個主要使用案例：

- 模仿人類行為，自動化前面板操作並節省時間的例程式，例如自動化相容性測試。工程師無需在每次需要測試新零件時坐在示波器前新增適當的量測值並寫下結果，而是開發一個指令碼來完成所有這些並顯示結果。
- 擴展儀器功能的用途，例如量測記錄、驗證或品質保證。自動化使工程師能夠執行複雜的測試，而沒有這些測試固有的許多缺點。操作員無需設定示波器或手動記錄結果，每次皆能以相同的方式進行測試。

本技術簡介將涵蓋開始使用 Python 程式設計示波器所需的內容，包括程式設計介面的基礎知識，以及如何下載和執行範例。

什麼是程式設計介面？

程式設計介面 (PI) 是兩個計算系統之間的一個或一組邊界，可以透過程式設計來執行特定的行為。就我們而言，PI 是執行在每台 Tektronix 測試設備中的電腦與最終使用者編寫的應用程式之間的橋樑。更精確地說，PI 是一組命令，可以遠端傳送至示波器以及該示波器上處理和執行 PI 的系統。

PI Stack (圖 1) 顯示了從主控制器到儀器的資訊流。最終使用者編寫的應用程式程式碼定義了目標儀器的行為。這通常是使用業界常見的開發平台之一編寫，如 Python、MATLAB、LabVIEW、C++ 或 C#。

此應用程式會將 SCPI 命令 (幾乎所有測試和量測設備的標準格式) 傳送至 VISA 層。在某些情況下，應用程式將叫用驅動程式，然後驅動程式會將一或多個 SCPI 命令傳送至 VISA 層。

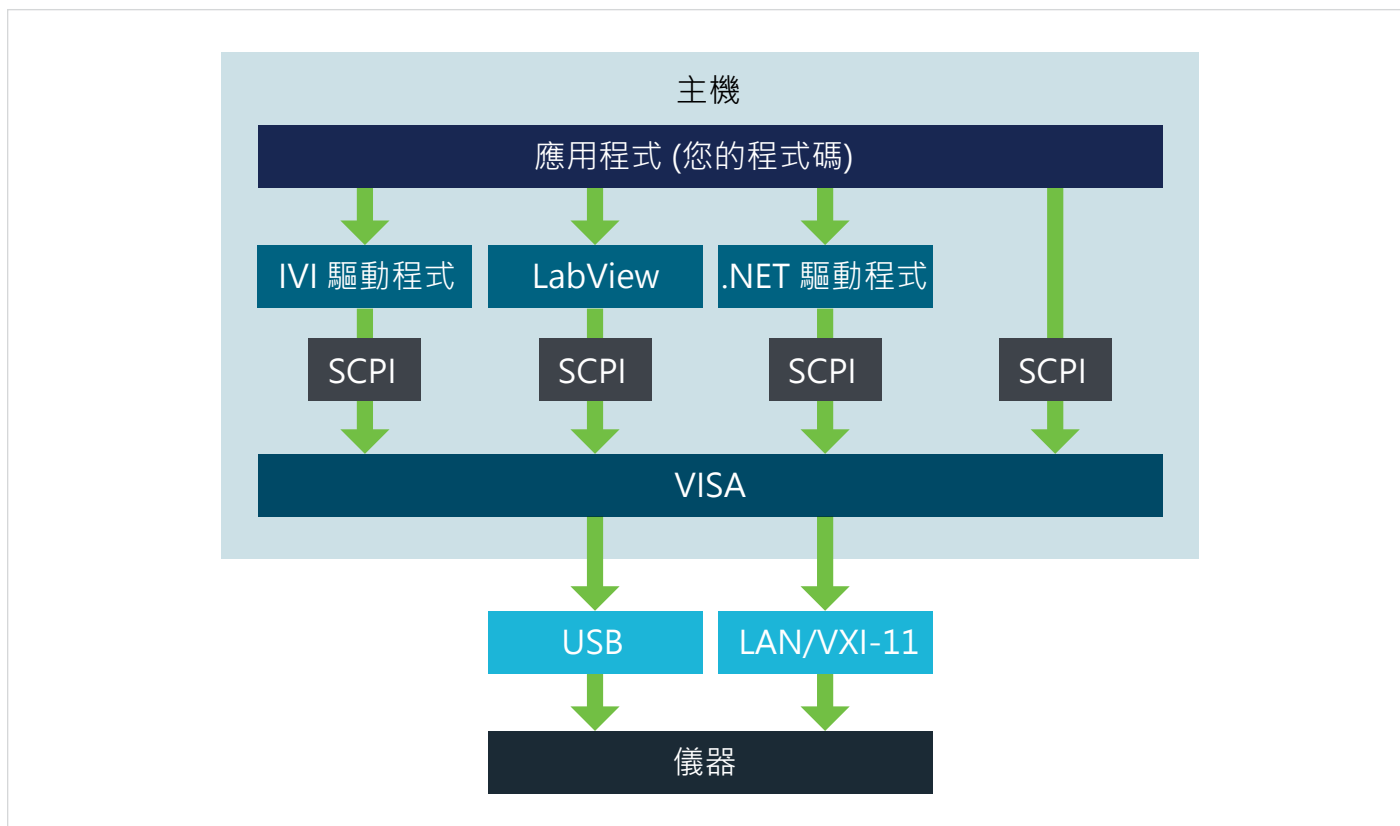


圖 1. 程式設計介面 (PI) 堆疊顯示主機控制器和儀器之間的資訊流

VISA 層透過 USB 或 LAN 介面協商與儀器的連接。VISA 可開啟和關閉與儀器的遠端工作階段，在匯流排上讀寫資料，並透過同步、排清和其他雜項功能維護連接的健康狀況。這一層將許多通訊所需的技術細節抽象化，如資源位址和緩衝區管理。

然後儀器將直接在 USB 或 LAN 介面上讀取和寫入資訊。根據發出的命令，可能會變更示波器應用中的設定或提供對查詢的響應。

實體通訊介面之上的所有系統也在示波器內鏡像，但我們可以將其視為「黑箱」，因為我們無法修改這些系統。每個主要的儀器供應商均提供了 VISA 標準的實作以與其產品搭配使用。所有基本功能都包含在標準要求中，因此任何供應商的 VISA 皆應能妥善地控制儀器。

如何開始使用自動化

為了快速開始，您應該在您的 PC 上安裝一些開發工具：

1. 安裝 NI VISA：
 - 從 National Instruments 下載安裝程式。URL 可能會變更，但可以透過搜尋「NI VISA」找到安裝程式。
 - 使用預設設定。
2. 安裝 Python：
 - 從 <https://www.python.org/> 下載安裝程式。
 - 使用「Install Now」(立即安裝) 安裝 Python，並啟用「Add Python 3.x to PATH」(將 Python 3.x 新增至 PATH)，將可更輕鬆地安裝模組。
3. 安裝 pyvisa：
 - 開啟 Windows cmd (命令行)
 - 按 Win+R 開啟「執行」
 - 鍵入「`py -m pip install pyvisa`」

下載並執行範例

為了示範如何執行 Python 程式，我們使用 Tektronix GitHub 頁面 (<https://github.com/tektronix/Programmatic-Control-Examples>) 上的檔案，位於 Oscilloscopes – Touchscreen Oscilloscopes & High Speed Digitizers 下。程式稱為 SimplePlotExample。

1. 從 GitHub 取得檔案。一種方法是開啟原始程式碼，然後以滑鼠右鍵按一下 Raw (原始) 並選擇「Save link as」(將連結另存為)。
2. 開啟 IDLE (預設的 Python 編輯器) 並開啟剛剛儲存的檔案。IDLE 將開啟文字編輯器，顯示程式碼。
3. 執行程式碼看看會發生什麼情況。在 IDLE 中有兩種方法：按一下 Run (執行) ->Run Module (執行模組) 或直接按鍵盤上的 F5。圖 3 中的畫面顯示程式碼存在問題。

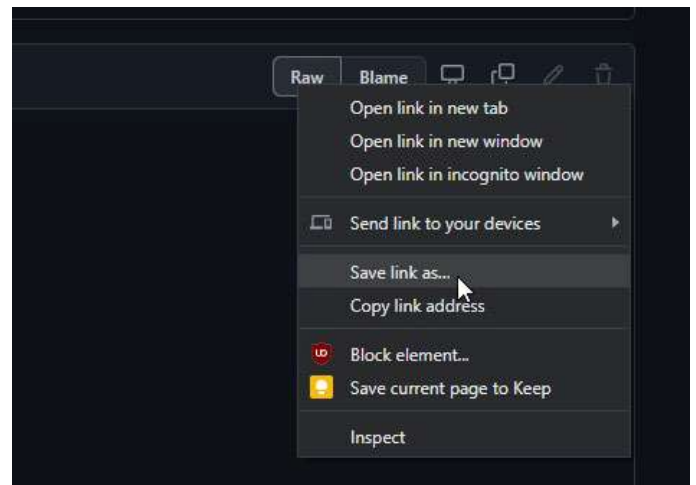


圖 2. 使用 Chrome Web 瀏覽器從 GitHub 儲存原始程式碼。

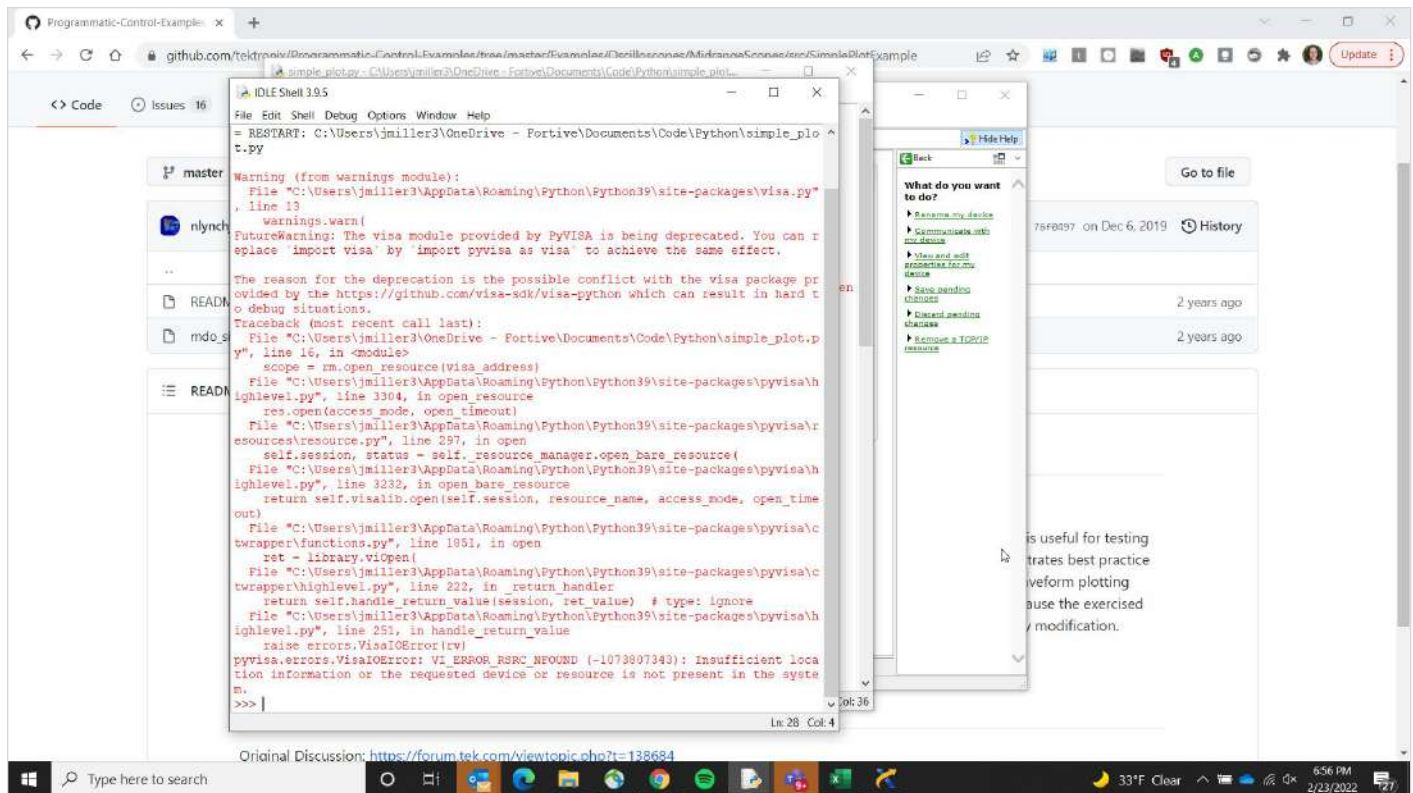


圖 3：錯誤的 VISA 位址

使用任何範例指令碼時要做的第一件事是查看 VISA 位址，這是告訴 VISA 在哪裡可以找到儀器的網路名稱。此範例程式碼指定了一個 USB 位址，這意味著使用者需要將完全相同的示波器連接至其 PC，就像此指令碼的作者編寫指令碼並將其發佈到 GitHub 時所執行的步驟。

若要更正 VISA 位址，請開啟作為 NI-VISA 的一部分安裝的 NI MAX。NI MAX 將允許使用者連接至示波器，對這些連接進行疑難排解，並根據需要傳送單個單獨的命令。

對於此範例，我們使用的是 LAN 連接。在圖 4 中，您可以在左側的樹形結構中看到網路上的任何示波器。位址為 192.168.1.17 的儀器是 4 系列混合訊號示波器，將作為本技術簡介的範例（您的網路上將會顯示具有不同位址的不同示波器）。

從 NI-VISA 中取得 VISA 資源名稱並將其貼至程式中，然後再次執行程式碼。

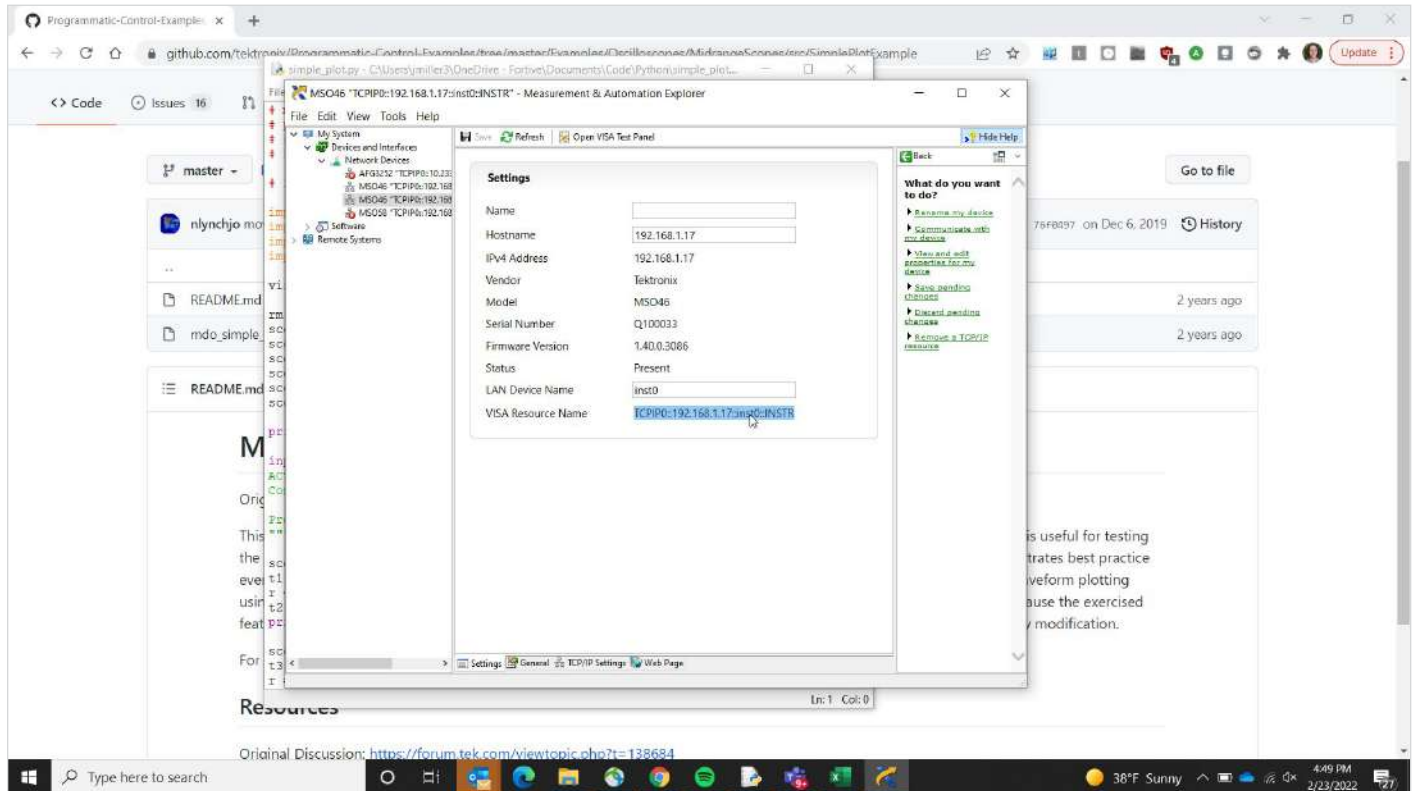


圖 4. 使用 NI MAX 確定正確的 VISA 資源名稱。

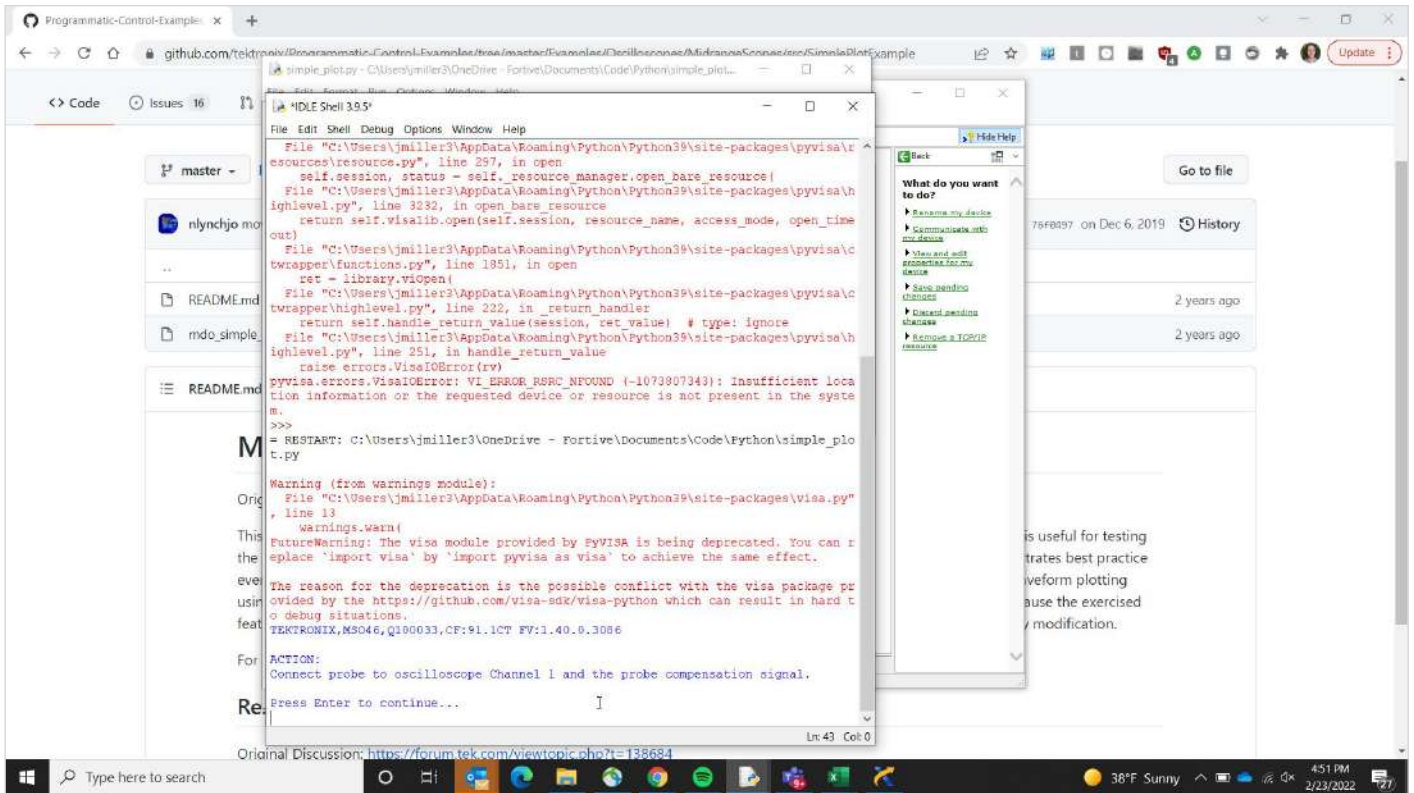


圖 5. Simple Plot 範例程式提示使用者連接探棒。

範例程式碼現在可以正常執行。程式會暫停片刻，要求使用者在通道 1 上安裝探棒並將其連接到示波器上的探棒補償訊號。

連接探棒後，按 Enter 繼續。這會在 PC 上產生示波器軌跡。

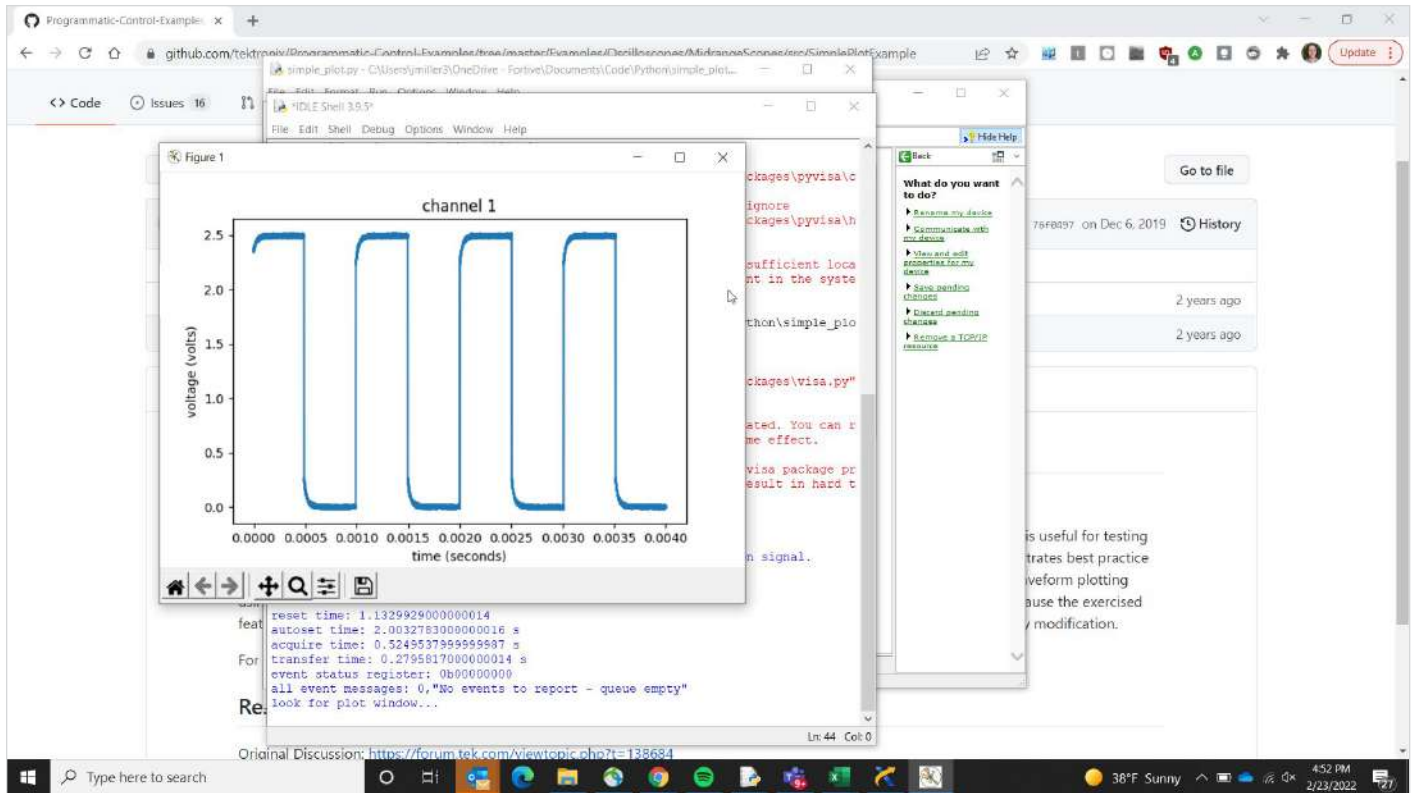


圖 6. Simple Plot 範例將從示波器傳輸波形並在 PC 上繪製。

用於示波器自動化的關鍵 Python SCPI 命令

可程式設計儀器的標準命令 (SCPI) 指令用於變更設定並從儀器取得量測值。如示波器之類的多功能儀器可能支援數百個命令。即使是不同的製造商和儀器，SCPI 命令的一般語法皆非常相似，進而更容易與不同儀器進行通訊。大多數 SCPI 命令都有一個用於變更設定的「set」(設定) 表單，以及一個用於從儀器取回資訊的「query」(查詢) 表單。指定儀器的精確語法通常記錄在程式設計員手冊中，您通常可在儀器製造商的網站上取得該手冊。[4、5 和 6 系列 MSO 的程式設計介面手冊](#)就是一個很好的例子。

SCPI 命令的完整描述超出了本技術簡介的範圍，但您應瞭解下列幾個命令：

- IDN?：告訴您連接哪個儀器，並提供握手和完整性檢查。
- OPC?：僅在操作完成時傳回的同步命令。
- ESR?：告訴您發生錯誤。這必須在 ALLEV? 之前傳送。
- ALLEV?：解釋錯誤。

採取下一步措施

開發人員從範例中複製和貼上程式碼是很常見的動作；這不僅可以節省時間，還可以幫助他們一路學習。瀏覽 [Tektronix Github](#) 上的程式碼範例以獲得完整的解決方案和靈感！

由於 Python 提供的可攜性、易用性和可擴展性，Python 是使用自動化擴展示波器功能的首選軟體環境之一。如果您要使用示波器執行重複或複雜的任務，則使用 Python 並從預先編寫的範例程式開始可以提供快速解決方案的途徑。