03

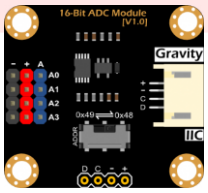
FUNGI KIT

Plotting Guide

THE UNIVERSITY OF QUEENSLAND
AUSTRALIA
CREATE CHANGE

Step 1: Saving the Data from the Raspberry Pi Pico

! Disconnect the **analogue-to-digital converter (ADC)** from the Raspberry Pi Pico by unplugging the wires before continuing.



Reopen Thonny, then plug the Raspberry Pi Pico back into the computer and press the red **Stop** button to tell **Thonny** to find it.

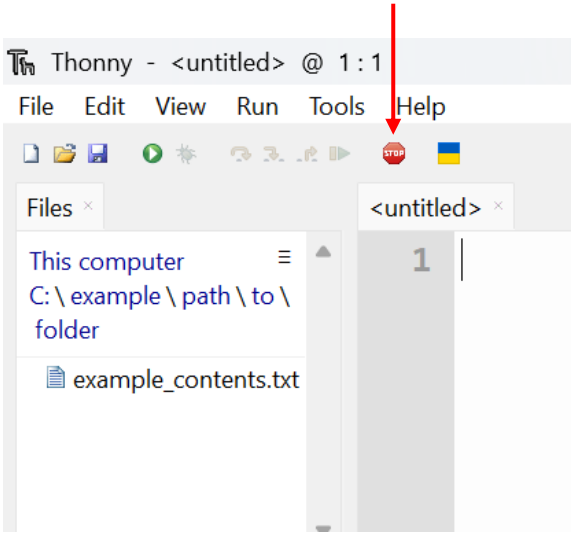


fig. 1

We want to save the data file from the Raspberry Pi Pico to a simple location. In the **Files** tab on the left of **Thonny**, click on the **C:** (fig. 2).

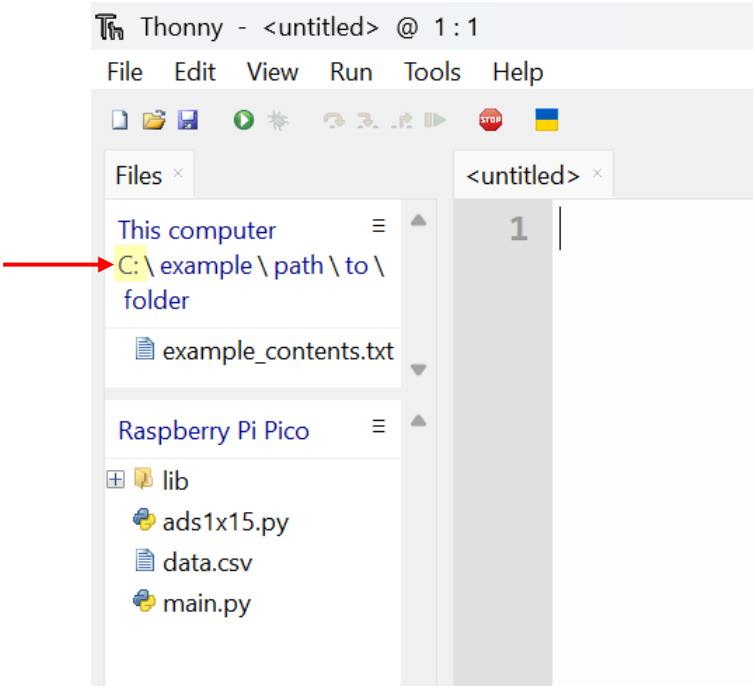


fig. 2

In the Files window, navigate to **Users → [Your Computer Name] → Documents → FungiKit**. You should now see a file named **fungi_kit_plotter.py**.

Note: You won't see the text **"Your Computer Username"** — it will show your actual computer name instead.

- Right-click on the **data.csv** file and click **"Download to C:\Users\[Your Computer Name]\Documents\FungiKit"**.
- Also open the file **fungi_kit_plotter.py** file by double clicking it.

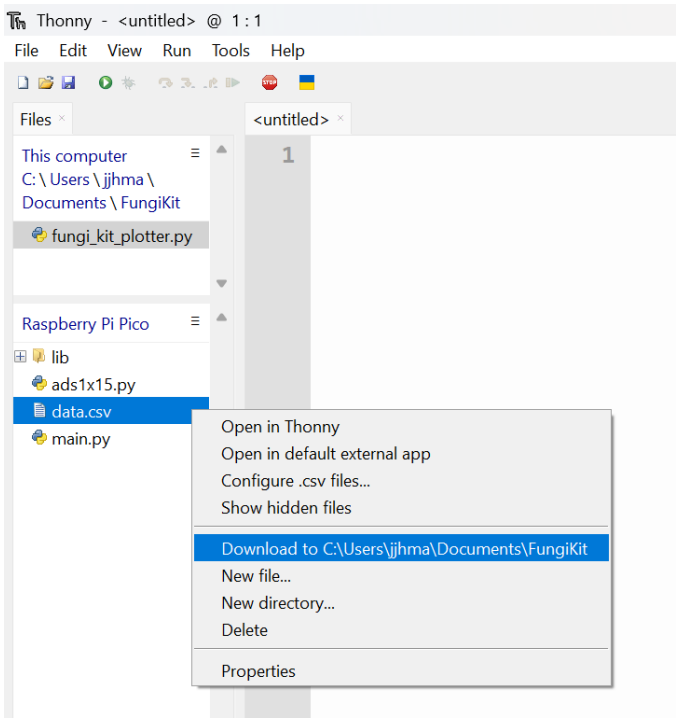


fig. 3

Step 2: Configure the Interpreter

- Go to the top toolbar and select **Run > Configure Interpreter...**

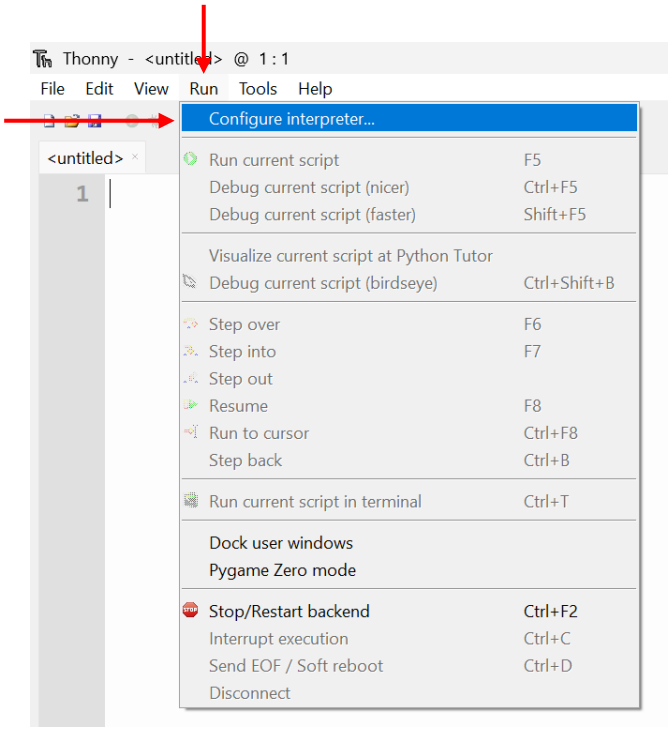


fig. 4

- In the first drop-down, select **Local Python 3** and then click **OK.**

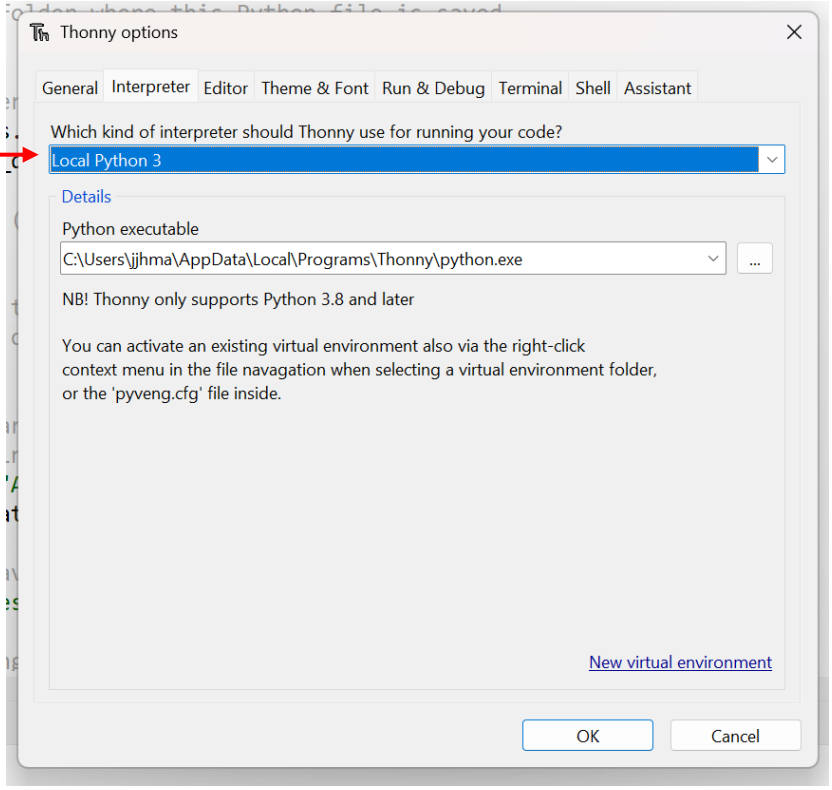


fig. 5

Step 3: Plotting the Data

- In the **fungi_kit_plotter.py** file you opened earlier, scroll down and rename the **“TITLE”**. Make sure to include the date of the recording taken in your title.

```
28
29 # Plot each of the sensor readings (A0–A3) over time
30 plt.plot(data["timestamp"], data["A0"], label="A0")
31 plt.plot(data["timestamp"], data["A1"], label="A1")
32 plt.plot(data["timestamp"], data["A2"], label="A2")
33 plt.plot(data["timestamp"], data["A3"], label="A3")
34
35 # Label the x-axis (horizontal) and y-axis (vertical)
36 plt.xlabel("Time")
37 plt.ylabel("Value (V)")
38
39 # Give the graph a title
40 plt.title("TITLE")
41
42 # Show a small box that explains which line is which
43 plt.legend()
44
45 # Display the graph on the screen
46 plt.show()
```

fig. 6

- Click the **Green Play Button** in the top left corner of **Thonny** to run the program.

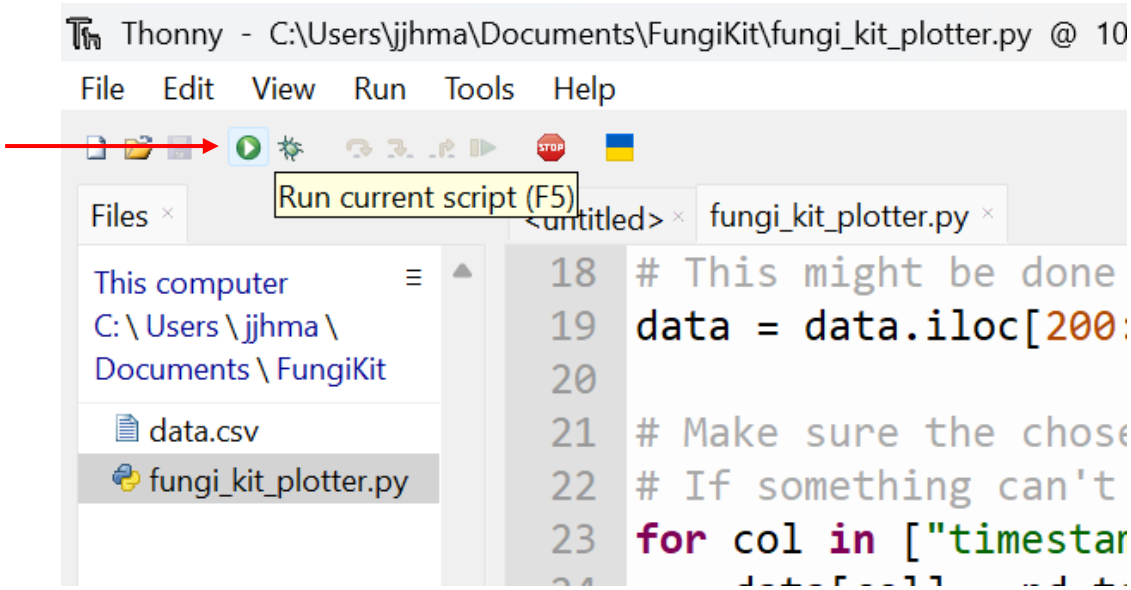


fig. 7

- After a few seconds, a new window will open with your data plot

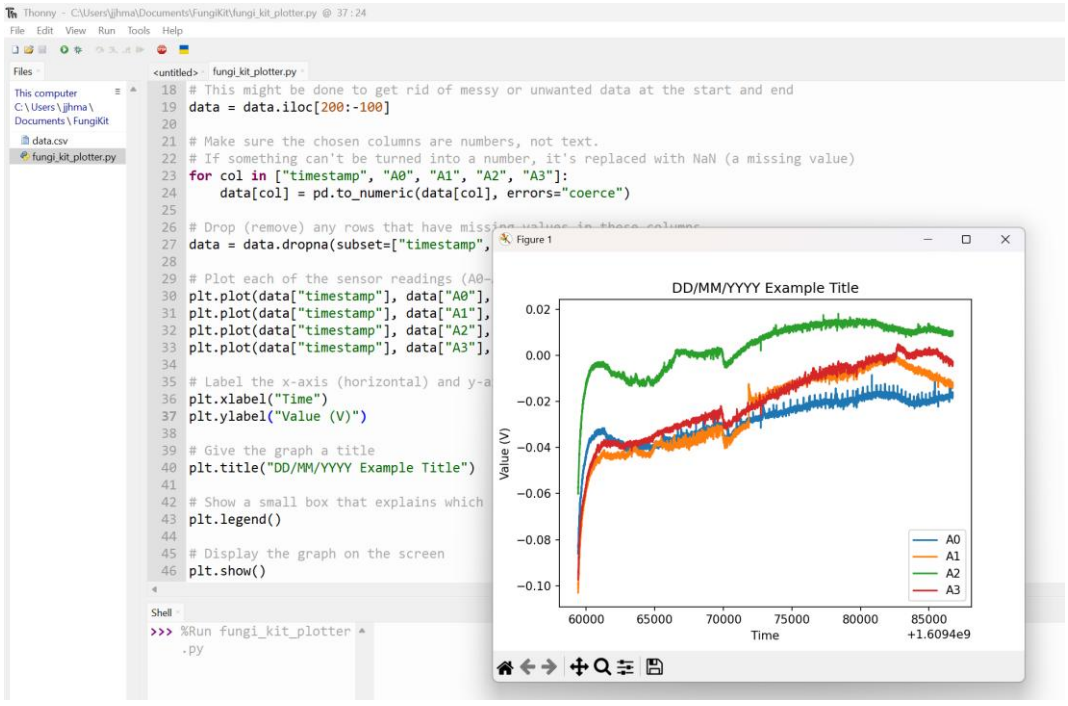


fig. 8

- Congratulations! You have now finished recording and plotting your data. Your teacher and the tutors will now explain how to analyse the recording.
- If you have finished this task quickly, flip over to the next page for an extension task.

Extension Task: Smoothing the Data with Filtering

- You may notice that the data you plotted is quite **noisy**, which means it jumps up and down a lot instead of following a smooth trend. Noisy data can happen for many reasons, like small measurement errors or natural variations in the mushrooms. To better see the overall pattern, we can use a method called **filtering** to smooth the data. Filtering reduces the random ups and downs, making it easier to spot trends and understand what the data is really showing.

- To filter the data, we need to add a few more lines of code **before plotting the data**. Add the following highlighted lines of code (fig. 9).

```
26 # Drop (remove) any rows that have missing values in these columns
27 data = data.dropna(subset=["timestamp", "A0", "A1", "A2", "A3"])
28
29 for col in ["A0", "A1", "A2", "A3"]:
30     data[col] = data[col].rolling(window=30, center=True).mean()
31
32 # Plot each of the sensor readings (A0-A3) over time
33 plt.plot(data["timestamp"], data["A0"], label="A0")
34 plt.plot(data["timestamp"], data["A1"], label="A1")
35 plt.plot(data["timestamp"], data["A2"], label="A2")
36 plt.plot(data["timestamp"], data["A3"], label="A3")
37
38 # Label the x-axis (horizontal) and y-axis (vertical)
```

fig. 9

- Run the script again using the **Green Play Button**.

- You should see that the data on the new plot is smoothed, and the spiking activity is more clearly visible.

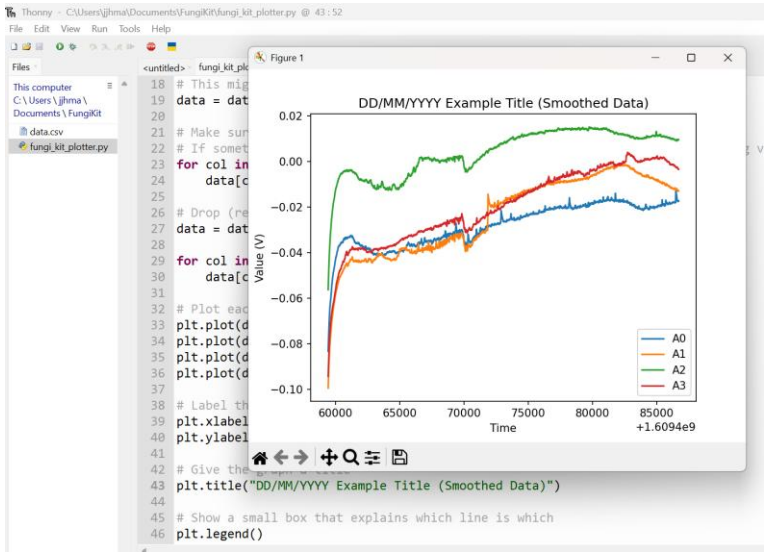


fig. 10

- Experiment with different window sizes by changing the number in **[window=VALUE]** and observe how it changes the graph.