Boot Time Graph Execution Process

# Running boot\_time\_graph.py

* To run the boot\_time\_graph.py script (located at: test\_scripts\tester\boot\_time\_graph\boot\_time\_graph.py), use the following command:
* python boot\_time\_graph.py
* This script is used to present Excel file measurements from both the Bootloader and Linux Loader in a graphical format.

# Requirements

* Install the following packages using the command: python -m pip install <pkg-name>

• Matplotlib – Used for interactive and animated visualization.

• Seaborn – Provides a high-level interface for drawing attractive and informative statistical graphics.

• SciPy – Required by Seaborn for various internal operations.

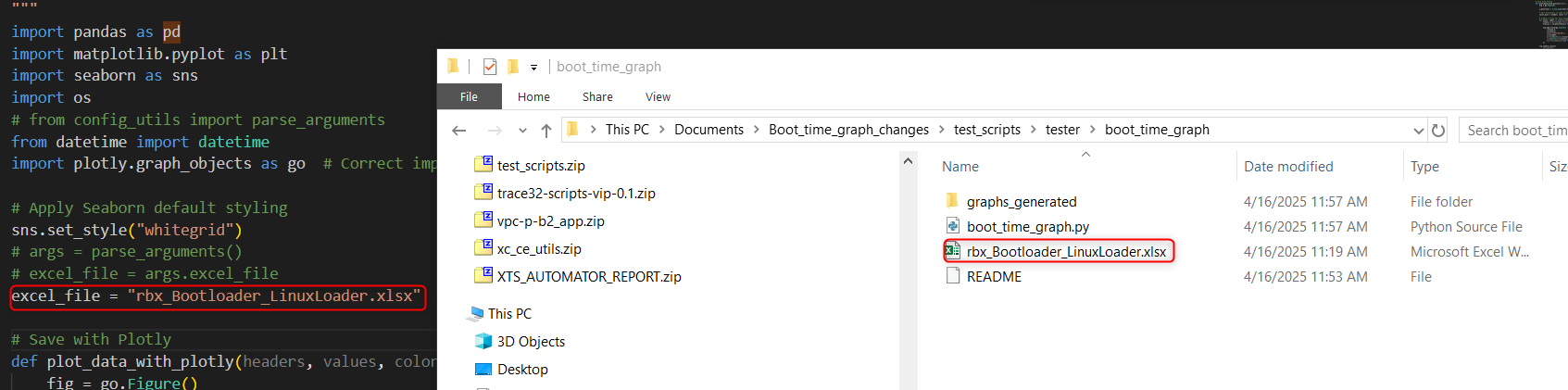
• Plotly – Supports exporting graphs to standalone HTML files.

• Pandas – Used to read Excel files.

* If you encounter an error like “ImportError: DLL load failed while importing qhull: The specified module could not be found,” run the command: pip install --upgrade scipy

# Running Conditions

• When running boot\_time\_graph.py as a standalone script, ensure you provide the Excel file name and comment out the parse\_arguments section.



• If running boot\_time\_graph.py from the boot\_time\_graph folder, comment out the import of config\_utils.

• Ensure the Excel file contains two sheets named either 'Boot Loader' and 'Linux Loader' or use the names available in your file.

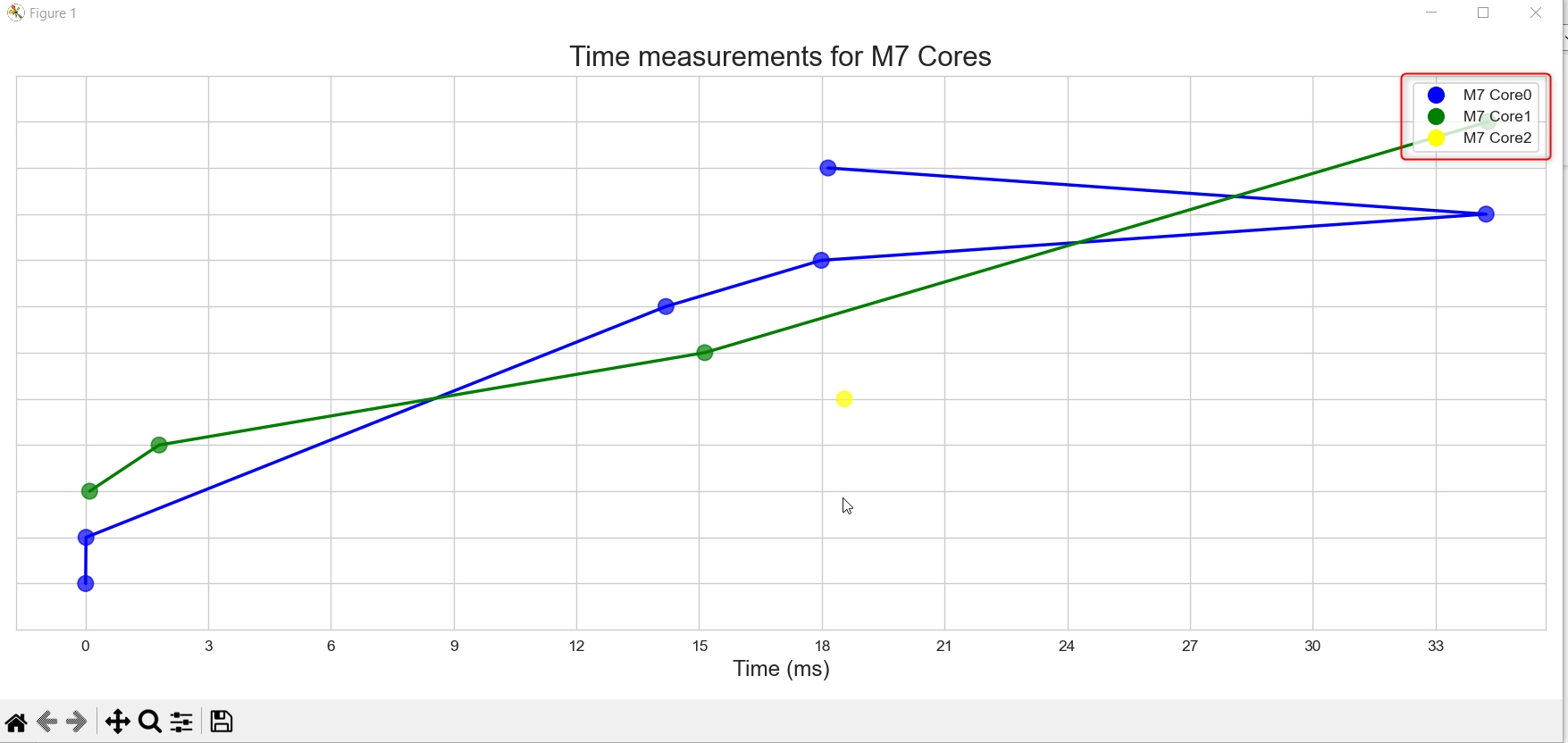
# Excel File Settings

• Define headers\_m7 and colors\_m7 for the Boot Loader sheet. 'Headers\_m7' represents all header/column names. Ensure all listed headers exist in the Boot Loader sheet; otherwise, remove or replace them.

• 'Colors\_m7' must have the same number of entries as 'headers\_m7' (e.g., if checking for 7 headers, provide 7 colors).

• Similarly, define headers\_linux and colors\_linux for the Linux Loader sheet.

• Legends on the right side represent what each dot means in the graph.

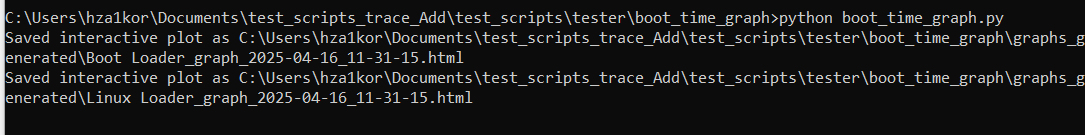


# Features of plot\_data()

• Creates an interactive scatter plot using Matplotlib, mapping measurement values to colored dots based on the provided headers and legend labels.

• Sets up an event handler so that clicking on any dot displays an annotation with the corresponding header and value.

• Saves an interactive HTML version of the plot using Plotly and then displays the Matplotlib plot.

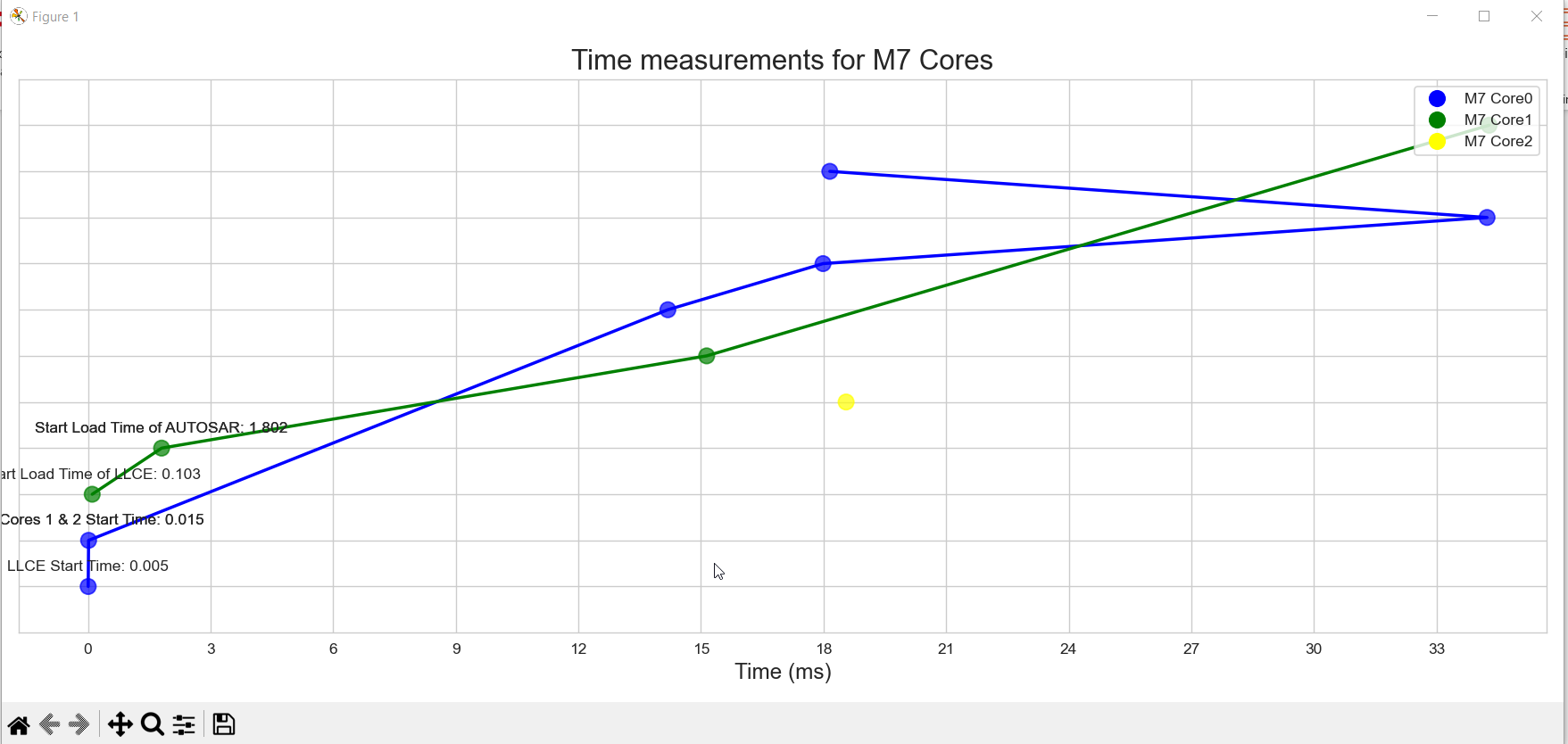


# Graph Saving and Demo

• After closing the graphs, the code saves them in the 'graphs\_generated' folder with timestamps.

• Graphs Demo:

– Bootloader



– Linux Loader

