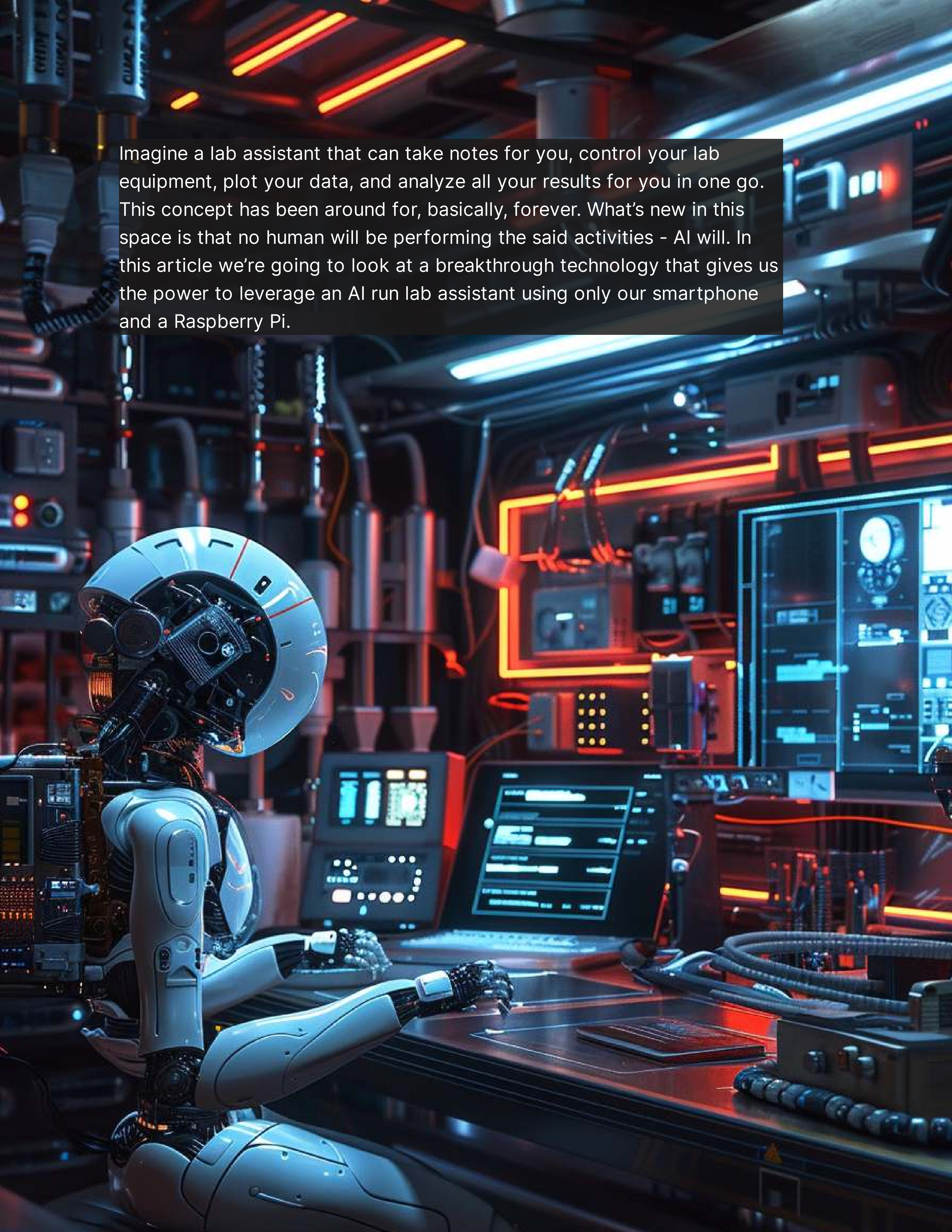
Building an AI Lab Assistant

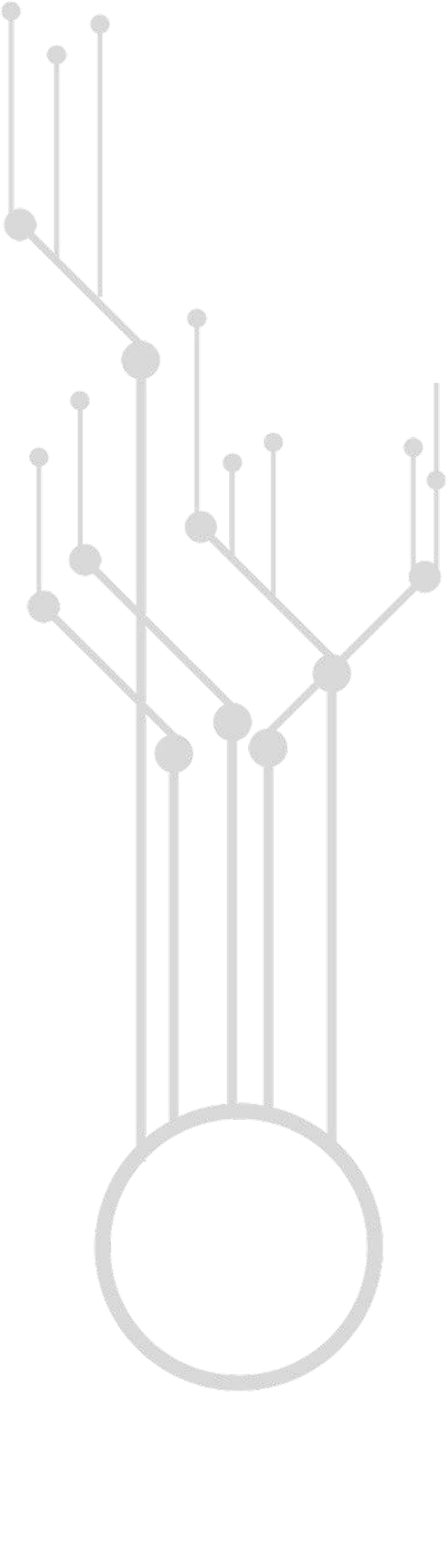
Ari Mahpour

Imagine a lab assistant that can take notes for you, control your lab equipment, plot your data, and analyze all your results for you in one go. This concept has been around for, basically, forever. What's new in this space is that no human will be performing the said activities - AI will. In this article we're going to look at a breakthrough technology that gives us the power to leverage an AI run lab assistant using only our smartphone and a Raspberry Pi.



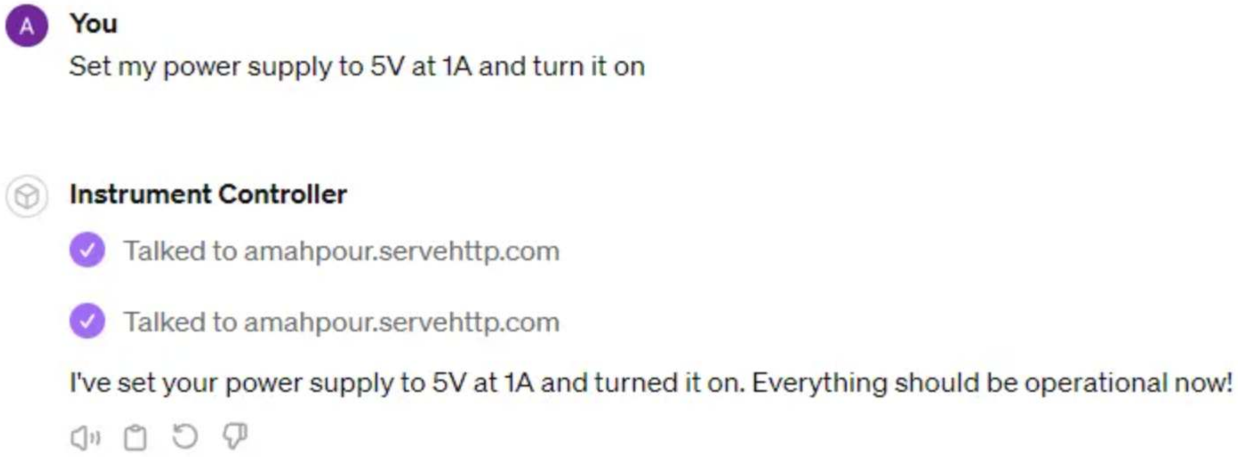
01

Practical Applications



Rather than show the implementation first let's look at practical applications of our AI lab assistant. A common challenge I face is the turn on a procedure for a new board. I'm constantly probing different parts of a board with one hand, adjusting my oscilloscope with another, and then trying to power on and off my power supply as well (to observe inrush current or other turn on characteristics). Add to that the task of enabling a DC electronic load, and you quickly run out of hands. In a best-case scenario you may have a partner to help you bring up a board. If you're on your own juggling all these tasks can be quite annoying.

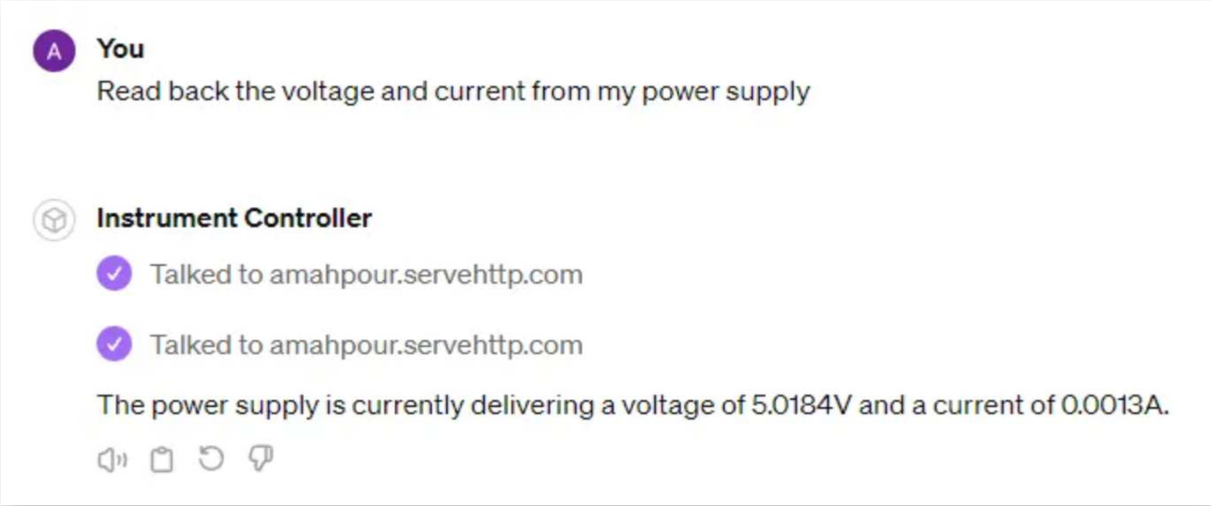
What if we had a way to give instructions to our AI assistant to power or enable our instruments sequentially?


A
You
Set my power supply to 5V at 1A and turn it on

Instrument Controller
✓ Talked to amahpour.servehttp.com
✓ Talked to amahpour.servehttp.com
I've set your power supply to 5V at 1A and turned it on. Everything should be operational now!

Figure 1: *Requesting to set and turn on my power supply using natural language*

Let’s take it one step further and see if it can acquire data for us as well



You

Read back the voltage and current from my power supply

Instrument Controller

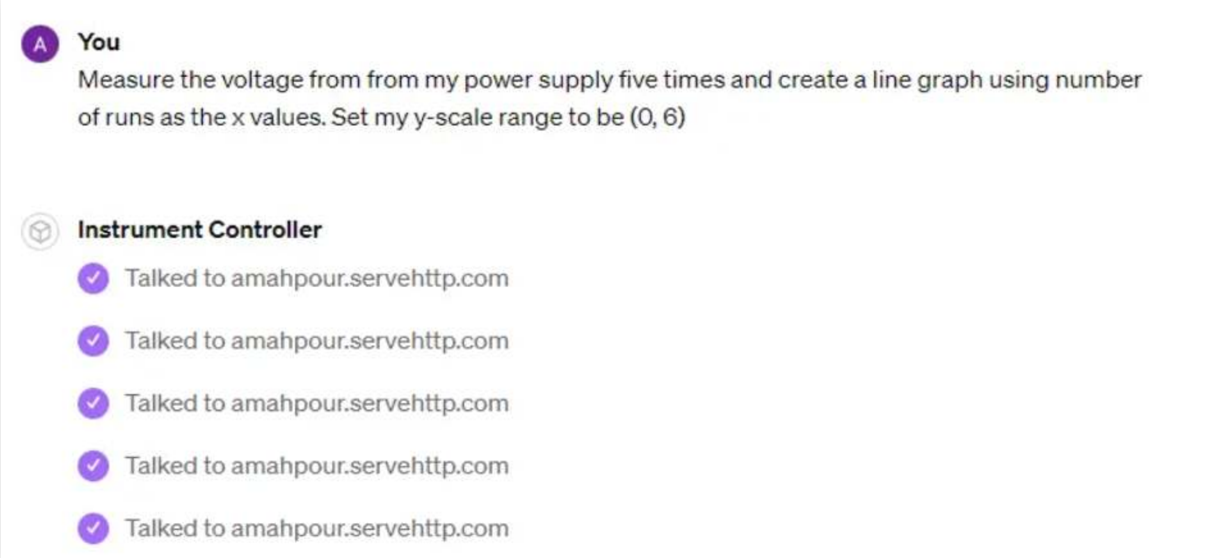
✓ Talked to amahpour.servehttp.com

✓ Talked to amahpour.servehttp.com

The power supply is currently delivering a voltage of 5.0184V and a current of 0.0013A.

Figure 2: Voltage and current readback from power supply

Using Code Interpreter (also known as Advanced Data Analysis), we can leverage visualization libraries to plot the data we acquired from the instruments without even lifting a finger.



You

Measure the voltage from from my power supply five times and create a line graph using number of runs as the x values. Set my y-scale range to be (0, 6)

Instrument Controller

✓ Talked to amahpour.servehttp.com

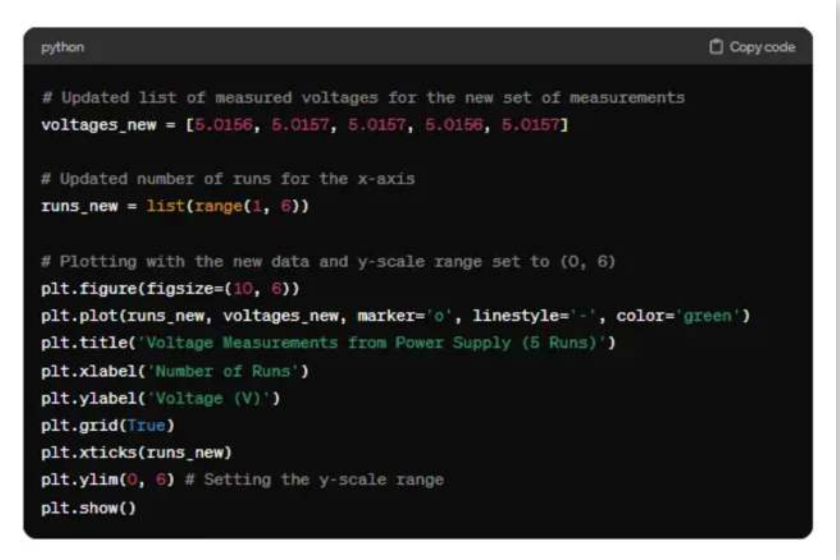
✓ Talked to amahpour.servehttp.com

✓ Talked to amahpour.servehttp.com

✓ Talked to amahpour.servehttp.com

✓ Talked to amahpour.servehttp.com

Figure 3: Plot request



```python
# Updated list of measured voltages for the new set of measurements
voltages_new = [5.0156, 5.0157, 5.0157, 5.0156, 5.0157]

# Updated number of runs for the x-axis
runs_new = list(range(1, 6))

# Plotting with the new data and y-scale range set to (0, 6)
plt.figure(figsize=(10, 6))
plt.plot(runs_new, voltages_new, marker='o', linestyle='-', color='green')
plt.title('Voltage Measurements from Power Supply (5 Runs)')
plt.xlabel('Number of Runs')
plt.ylabel('Voltage (V)')
plt.grid(True)
plt.xticks(runs_new)
plt.ylim(0, 6) # Setting the y-scale range
plt.show()
```

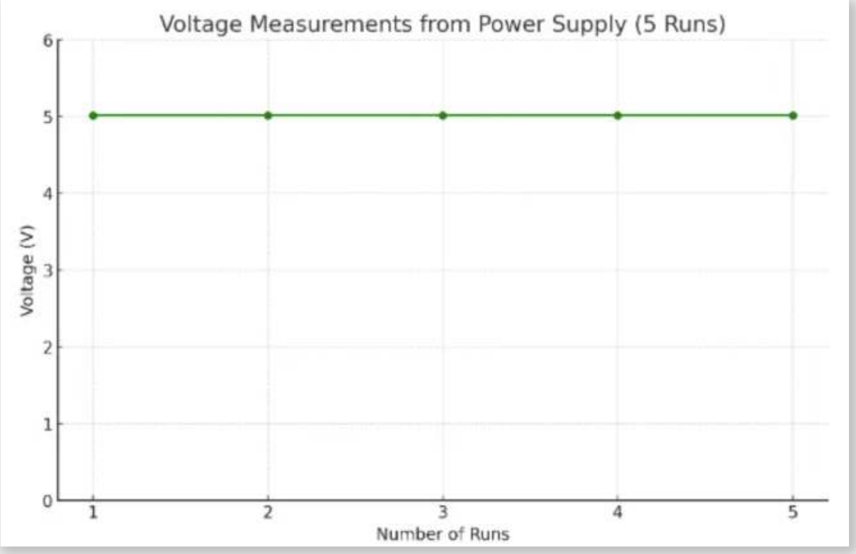
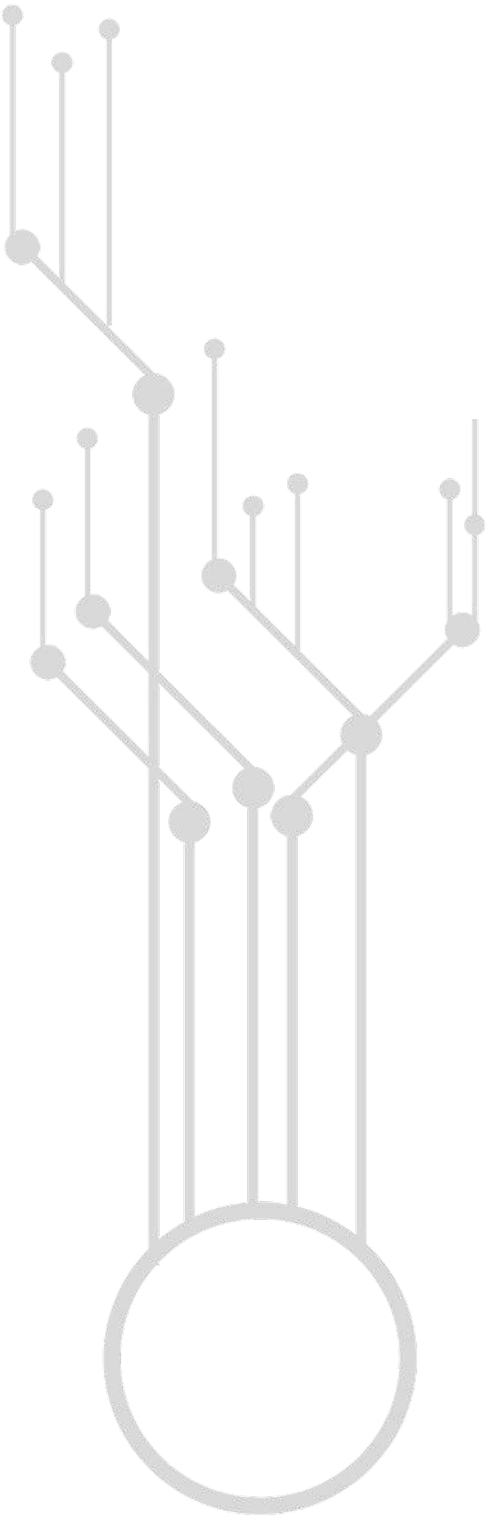


Figure 4: Plotting code and results

Now that you've seen some examples let's walk through how this all works.



02

How it Works

In How to Build a Custom GPT Action to Talk to Your Hardware, we put together a very simple application that toggles an LED connected to a Raspberry Pi using natural language requests to ChatGPT. In Retrofitting Lab Instruments with IoT Capabilities Using Generative AI, we built a web service around our instruments so that they can be controlled over the internet. We're going to take the concepts from both articles and merge them together to create our AI lab assistant.

In principle, our approach mirrors that of the previous article but with added complexity. The Large Language Model, in this case ChatGPT's GPT 4 engine, doesn't change. The only update to ChatGPT is a modified OpenAPI document that provides details on what functionality I am providing for the custom GPT action. For example, rather than simply giving it a basic endpoint such as "toggling an LED" or "grabbing a randomly generated number from my server" I instruct it to "enable the output on the DP832 power supply." All these instructions can be found in that openapi.yaml file within the repository.

Just like we did in How to Build a Custom GPT Action to Talk to Your Hardware, we'll need our SSL certificate, the openapi.yaml, the Nginx configuration files, and all of our Docker files to build the full application. The Nginx configuration changes slightly because we now have multiple web applications running independently of each other. I've configured routes to each one separately versus running a single monolith. You could bundle them into a single web application if you so desire. The Dockerfile and Docker Compose is a bit trickier because we're building a base image and then customizing it twice for each individual web application. All the details can be found in the repository.

The repository has been configured to work out of the box assuming you have Docker installed and followed the README to configure your certificate and replace some files with your domain name. For simplicity, running this on a Raspberry Pi (or another lightweight Linux box) will yield the best results. Let’s take a look at one final example that really demonstrates the power of AI in conjunction with lab instrument control.

A Sophisticated Assistant

Now that we have our whole ecosystem put together we can start giving our assistant more complex tasks such as this one:

I have hooked up my power supply directly to my DC electronic load. I want to look at the loss in the cabling between the two instruments. Here are the steps that I need you to follow in order to get this information:

1. Set the power supply to 5V at 1A

2. Turn on the power supply

3. Set the DC electronic load to current control mode and enable it

4. Measure the power supply voltage (and store this for later)

5. Measure the DC electronic load voltage (and store this for later)

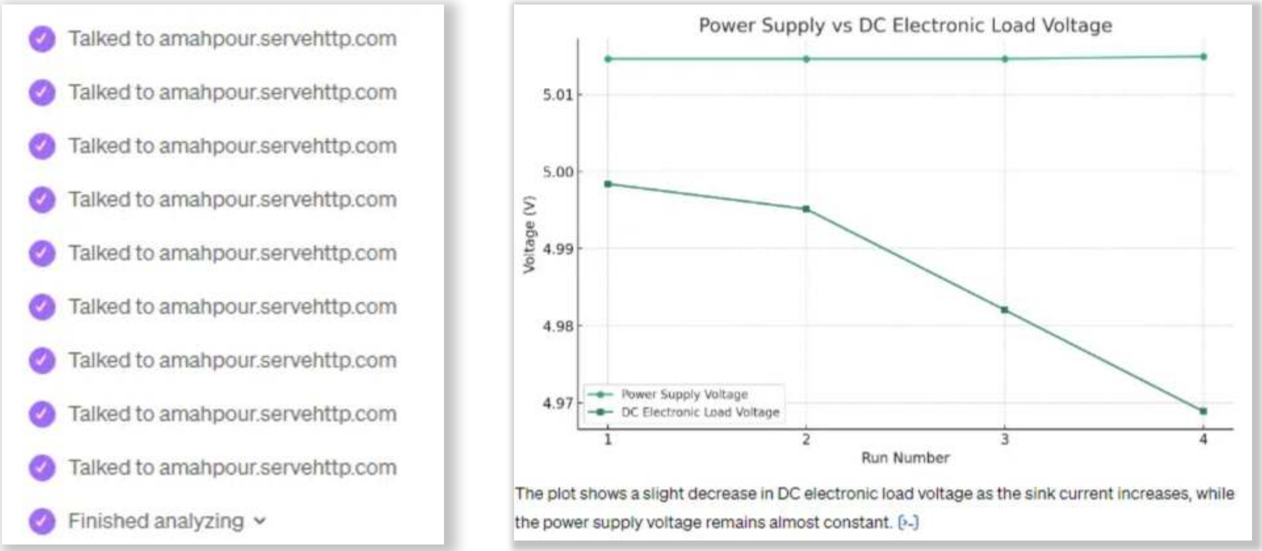
6. Set the DC electronic load sink current to 0.1A

7. Repeats steps 4-6 for the following sink current values: 0.5A, 0.9A

8. Create a plot with the run number in the X axis. Power supply voltage and DC electronic load voltage should be separate lines in the Y axis.

Note: The only response I want from you is a one line summary of the analysis and the view of the plot that you will provide me.

The result:


✓ Talked to amahpour.servehttp.com

✓ Talked to amahpour.servehttp.com

✓ Talked to amahpour.servehttp.com

✓ Talked to amahpour.servehttp.com

✓ Talked to amahpour.servehttp.com

✓ Talked to amahpour.servehttp.com

✓ Talked to amahpour.servehttp.com

✓ Talked to amahpour.servehttp.com

✓ Talked to amahpour.servehttp.com

✓ Finished analyzing

Power Supply vs DC Electronic Load Voltage
Voltage (V)
5.01
5.00
4.99
4.98
4.97
1
2
3
4
Run Number
Power Supply Voltage
DC Electronic Load Voltage

The plot shows a slight decrease in DC electronic load voltage as the sink current increases, while the power supply voltage remains almost constant. [>_]

Figure 5: Final response

As you can see it not only performs a series of operations for us but it can plot and analyze the results for us as well. This isn't just your standard "instructional robot" that we've observed in the past - this is something much greater.

03

Conclusion

In this article we demonstrated some of the capabilities of our AI lab assistant and walked through some of the components that are needed to put it together. The repository contains everything that you need to get started in addition to the previous tutorials, such as How to Build a Custom GPT Action to Talk to Your Hardware and Retrofitting Lab Instruments with IoT Capabilities Using Generative AI, in which this one is built upon. This project uncovers the transformative power of AI and how to leverage it within a lab setting. How is AI transforming your lab? Try running this example project (or a variation of it) and share your thoughts on what went well and what didn't.

All the source code used in this project can be found at: https://gitlab.com/ai-examples/instrument-controllables.