

# Context Is All You Need: AI Engineering with the Peto 'Bittle X' Robot Dog

---

## Overview

In this workshop, students will learn the fundamental concepts of [AI Engineering](#) by connecting Large Language Models (LLMs) to the documentation for programmatically controlling the [Peto 'Bittle X' Robot Dog](#). This hands-on experience demonstrates how AI becomes truly powerful when given the right context, transforming from a general-purpose tool into a specialized assistant capable of controlling hardware through natural language.

## Key Objectives

By the end of this workshop, students will:

1. **Understand AI Engineering:** Learn what distinguishes AI Engineering from traditional programming and why it's becoming the future of software development.
2. **Experience Context Engineering:** Discover how giving AI specific context (like API documentation) dramatically improves its capabilities, making it more useful than general-purpose AI.
3. **Build an AI Context Pipeline:** Connect the Peto API documentation to an AI model, enabling natural language control of the robot.
4. **Practice Responsible AI:** Implement transparency features that show exactly what commands the AI sends to the robot before execution.
5. **Develop Problem-Solving Skills:** Use AI tools like GitHub Copilot, Microsoft Copilot, and ChatGPT to overcome coding challenges, regardless of prior programming experience.

## Materials

- [Peto Bittle X Robot Dog](#) (1 per 5-6 students)
- Laptops with Python and VS Code installed
- Python packages: [openai](#), [python-dotenv](#), [pyserial](#)
- OpenAI API key, preferably loaded from a [.env](#) file
- Peto API documentation ([sending-skills.md](#) and [serial-protocol.md](#))
- Pre-configured starter code

## Rough Draft

DELETE EERYTHINV BELOW WHEN DONE.

(Short intro here)

(Following bullet points are rough outline of where I'm going with this workshop. Maybe needs to be an Overivew section then another section called Key Objectives than another section called Mateirals then another section where we do the 1,2,3 exact steps to execute the workshop including all required code in a

sub folder called "code" of this same folder that we link to if it's a lot of code, or if not just inline as code fenced)

(Find some place in this to link to <https://www.philtschmid.de/context-engineering> where it talks about how the new skill in AI isn't really just prompt engineering it's context engineering where you have to put in work to give AI the right context. You as a human connect the dots)

- Teach students the fundamental concept of AI Engineering, the importance of connecting AI, specifically in this case large language models (LLM's), with specific stores of contextual knowledge to empower them to actually provide value. This is part of the growing field of [AI Engineering](#), as anointed by Andrej Karpathy, (speak here about who he is, his prestige, role at OpenAI, and contributions to the space).
- We assume that the students have played around with ChatGPT. They've used it. AI is in their face, it's everywhere. They've likely had experiences where it kind of sucked. Why? Well, the model was trained on a set of data, and that's all it knows. It isn't alive. It isn't constantly retraining itself. The training process itself is very expensive and takes time and massive compute.
- To combat this, in 2020 Meta released to the world a strategy called Retrieval Augmented Generation where they basically gave the ability for an AI model to go and look up more information live from data sources that someone wires up to the model. Now it can know things beyond all of what it was trained on, which is good, but how much important data is there out there that isn't publicly available for OpenAI or Google or Meta to train their models on or simply the fact that there's just too much out there to train a given model on literally it all, so you got to have people using these models go and put in some work to connect the model to their rich source of information that can make it a lot more useful for them. Use analogies that are relatable to Gen Z high school students to drive this home.
- Emphasize what a model is. use the analogy "hey when you see ChatGPT do you ever see that dropdown menu? Why are there different options?" and the point of defining this is to get to them mentally "oh wait they trained each of these things on data, on information, and tweaked them to be good at different things, but if I want to use one for my own thing, let's say have it help me write, and all my writing is in my private Google Drive, well then I'd have to do some work connecting a model to my data so it can really help me". They have tools these days to help you do that and those tools are built by software engineers, product managers, business analysts, data scientists, data analysts, all of these people are hopping on the boat and becoming AI Engineers, people connecting AI with rich complicated sources of knowledge.
- Okay, now we have set the stage, now we have explained where we're going, now we're going to put this live into action using a fun tool, the Peto 'Bittle X' Robot Dog.
- The robot dog has a manual. The company that makes it, Peto, has done all the hard work of doing the math and putting the robot dog together and all that really challenging stuff of figuring out the low level code that's required to control it. On top of that, using a high level programming language, they've given us the ability to control the robot dog with Python or C++. (Try to help the students understand low level versus high level very simply. Layers of abstraction)
- But here we have an opportunity! We could write code in Python or C to do all sorts of fun things with the robot dog, have it do dance routines, and backflips, even write a program to have it be a guard dog and all that. That's traditional software development/engineering, that's what programmers do, but these days with AI, we have another path. With those same coding skills, instead of writing all the code yourself to make the robot dog do something, you can assume the role of an AI Engineer!
- Since we have really good documentation about what kind of code you have to write to make the robot dog do something (We have an API, application program interface, provided by petoi, basically a set

of really detailed instructions. a manual, a guidebook, for anyone writing code to interact with their product, companies do this all the time for things they give to the world. For example, TikTok and Instagram have API's that let programmers write their own code to control stuff. Ever seen bot accounts on TikTok or IG? That's how people build them, they write code against the TikTok or IG API!). Since we have a robot dog API, we can connect it to an AI model and then write natural language (prompts, prompt engineering!) to control the robot dog. That's being an AI Engineer!

- So, what we're going to do is challenge you to write the kind of code that allows AI to know how to control the robot dog without you having to write it all yourself in terms of writing a routine
- (At this point I need to have a specific goal for what they should accomplish after connecting the AI model to the petio documentation/instructions. Like if they get it connected what should they type into the chat window to try and get the robot dog to do some sort of specific routine and then I come around and validate they did things correctly, or have challenge so what should be possible based on how they wrote their code)
- Students should be writing Python code, and using AI (they can use Microsoft Copilot, ChatGPT, whatever) to help them connect the dots.
- The challenge here is that this workshop needs to be approachable to students of all backgrounds. Python is known for being syntactically similar. The code they're editing should give them enough hints and clarity about where to start and what to change to get things working. Even if a student has zero coding background, a part of this workshop is opening their eyes to the world we live in with AI. Don't tell me you have no idea how to code, tell me you know how to go to ChatGPT or Copilot, copy and paste in all the code, ask it to explain to you what it's doing because you don't know jack squat about coding, even give it the petio specs and the challenge, and have it tell you exactly what to do to figure this out. That's what I want to see. I want to see problem solving students. I want to see their minds open to the world we live in. Don't sit there and say you can't do it, use every tool available to you to solve the problem.
- So the goal is for them to successfully build a sort of RAG pipeline that connects the specification for the petio bittle to a model and then the chatbot (this part we got to figure out, the chatbot piece, maybe just the terminal? Make it simple, have them run the python code and it turns into a live terminal that's their chatbot) where they should be able to just chat and control the robot dog that way. They should have semantic level meaning, just kind of describing things and having the chatbot use AI and send commands to the robot dog.
- One key piece of the code they write is the challenge that every single command the chatbot sends to the robot dog it should return to you in a nicely formatted structured list like "I sent the following code commands to the robot dog". Drive home Responsible AI here. Don't let this thing do whatever it wants, that's like a black mirror episode.

## Logistics

Logistics on how to pull off this workshop.

- Students are working in browser based <https://vscode.dev/> so that we don't have to install VSCode on every laptop.
- Challenge is checking if from VSCode dev you can connect to the petio that's linked via bluetooth or manually plugged in. Or do you have to install all sorts of local stuff to enable coding with the petio bittle

## Challenge Station Idea

Could I have a condensed version of this workshop where I don't think of it as a whole thing delivered to students but I have a 1 page print out that's set next to a laptop all set up with the local stuff required to be connected to the Petoit Bittle and the 1 page print out is a challenge sheet like "Can you solve this challenge!" or "Can you be an AI Engineer?" or something like that and it has python code in VS Code and it very succinctly says stuff like

- The program you're looking at is called Visual Studio code! (briefly explain what it is)
- The robot is the Petoit 'Bittle X' robot dog!
- In VS Code, there's a file called (name of the file with all the specific API call instructions for the Petoit)
- Also in VS Code, there's this window to the right that has a tool called GitHub Copilot (or maybe I do this using a terminal or a prebuilt chatbot web interface I don't know)
- They are encouraged to use GitHub Copilot inside of VS Code to solve this. Specifically we want them to learn how to use that since it's a coding based AI tool. Explain it as it's like ChatGPT but specifically for coding, it can see all the file so you can ask it some questions! Sort of thing
- Can you figure out how to change the code in (python code file name here) such that the AI chatbot's model is able to understand how to control the robot dog and you can write natural language commands to make the dog do things?
- This is called AI Engineering? Do you have what it takes to be an AI Engineer?

So this way instead of having to scale the robot dog to a classroom that might have 40 students and given the \$300 price tag of the dogs it's going to be hard having 5+ students huddled around 1 laptop trying to do this 1 thing at the same time, it becomes a challenge sort of thing. Maybe there's a main workshop where there's other AI related teaching going on and students are released in small groups, or 1 by 1 to see if they can solve the challenge, and always have some easy way to reset the challenge (maybe a reset button or script that just uses git to reset to the local version, revert all changes, use version control)

I like this idea. Then any student that solves the challenge has to call an instructor over for validation and we give them an AI Engineer certificate and maybe some candy and have them take a picture with the robot dog and the laptop and praise them for problem solving. The challenge can be on tables during lunch or an open networking session or something like that.

(The Challenges need to be each a 1 page document separate from this overall Context is all you need AI Engineering workshop, but this document can link to them)

Here are the challenges in short I want.

1. The challenge mentioned above, can you be an AI Engineer with the robot dog.
2. A variation of the same challenge that uses the Robot Dog with the Robot Arm and specifically is asking students to write code to have the robot dog pick up and move an item (keep references to the item generic, examples would be a tennis ball, small cup, etc.)

Both could be in the same 1 page document that you just clearly say something like "if you're working with the robot arm dog then this is your challenge else if you're working with the standard dog" sort of thing.