Using Creole Forth to control a Picar-X

Joseph M. O'Connor

December 2024

What is the Picar-X?

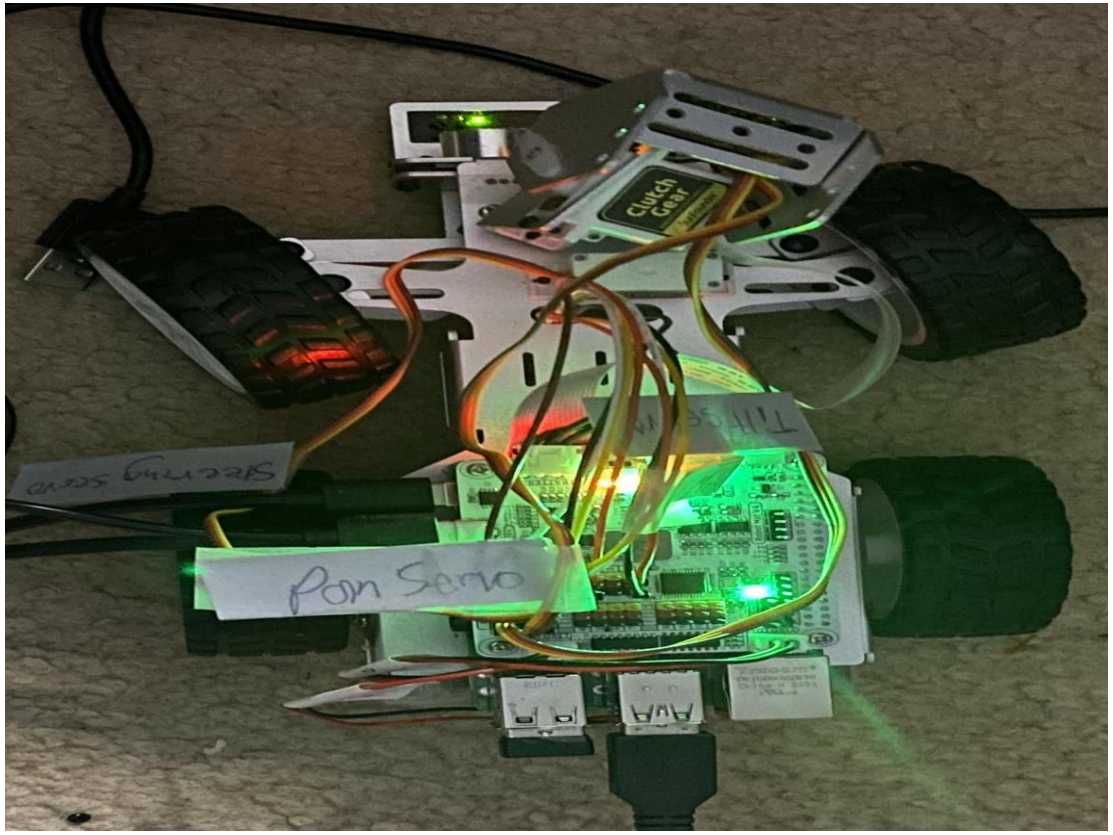
- A robot car kit (can be ordered on Amazon).
- Requires physical assembly – can be done in about a day.
- In the car for this presentation, it's controlled by a Raspberry Pi 4B with 8 GB of RAM and 64 GB SD card.
- This is a very generously proportioned system by the standards of most Forths.
- For those who want to save money and some power, the Raspberry PI Zero with 512 MB of SDRAM can be used instead ($22 vs $85).
- Interfaces with a microcontroller "hat" which sits on top of the PI.
- Programmable by default in Python and Ezblock
- Programmable in Python means it can be used with Creole Forth for Python.

What is Creole Forth

- A Forth-like scripting language that is intended to sit on top of a base language. Can be used as a DSL (domain-specific language) or macrolanguage to customize an application.
- Idea is derived from Norman Smith's UNTIL, which is a Forth interpreter built in C/C++.
- There are versions of Creole Forth available in several different base languages, such as Delphi/Lazarus, Visual Basic, JavaScript, and Python.

Appearance of Picar-X (this one is not mine)

This one is mine



Picar-X – what can it do

- Documentation is at the Sunfounder web site.
- There is a list of sample scripts in the Play with Python link such as line tracking, cliff detection, and obstacle avoidance.

Steps

- (1) Put together the physical components of the car.
- (2) Wire the ”hat” (controller interface that works with the Raspberry PI)
- (3) Download the Python libraries and code.
- (4) Test with Python code examples.
- (5) Use Creole Forth to make commands based on the examples.

Creole Forth Commands in current setup

- NOD – nods head yes
- SHAKE – shakes head no
- FWD – goes forward
- BACK – goes back
- LEFT – turns left
- RIGHT – turns right
- STRAIGHTEN – straightens front wheels out
- SAY – says some words

Some challenges

- Can't install many of the machine learning math libraries in default "managed" environment. Must set up a virtual environment.
- In Jupyter Notebook, some of the libraries installed didn't appear to be usable on the Raspberry PI even after installation. Possible workaround: Set up a script file in Python, wrap a primitive in the notebook to execute it.
- Picar-X TTS library didn't appear to work – possibly because the speaker on the robot hat was damaged. Workaround was to use espeak, a Raspberry PI TTS utility.

Intentions/Goals?

- Shorter term : explore capabilities of Picar-X with Creole Forth.
- Longer term : Learn to use Raspberry PI together with microcontrollers such as arduinos.
- Raspberry PI side – offers potential of access to machine learning/deep learning/statistical analysis facilities such as linear/logistic regression, matrix manipulation, and information management facilities such as Pandas dataframes.
- Microcontroller side - can handle sensors and data collection.
- In some cases, it may be possible to swap out functionality from one to the other.
- Example of the above: using Raspberry PI code for TTS instead of the Robot Hat.

Questions?

Demo

- Commands are executed in Jupyter Notebook on the Picar-X Raspberry PI
- They can be seen through the viewer available at connect.raspberrypi.com.
- We'll watch the Picar-X itself through an iPhone when commands are issued from the notebook.
- We use the Jupyter Notebook facility of 'magic commands' to interface with the Creole Forth code.