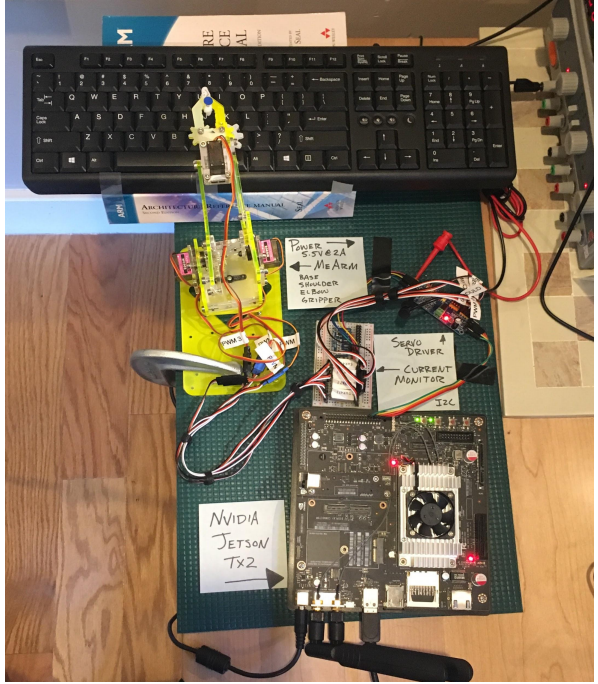


Ty P. Eypt

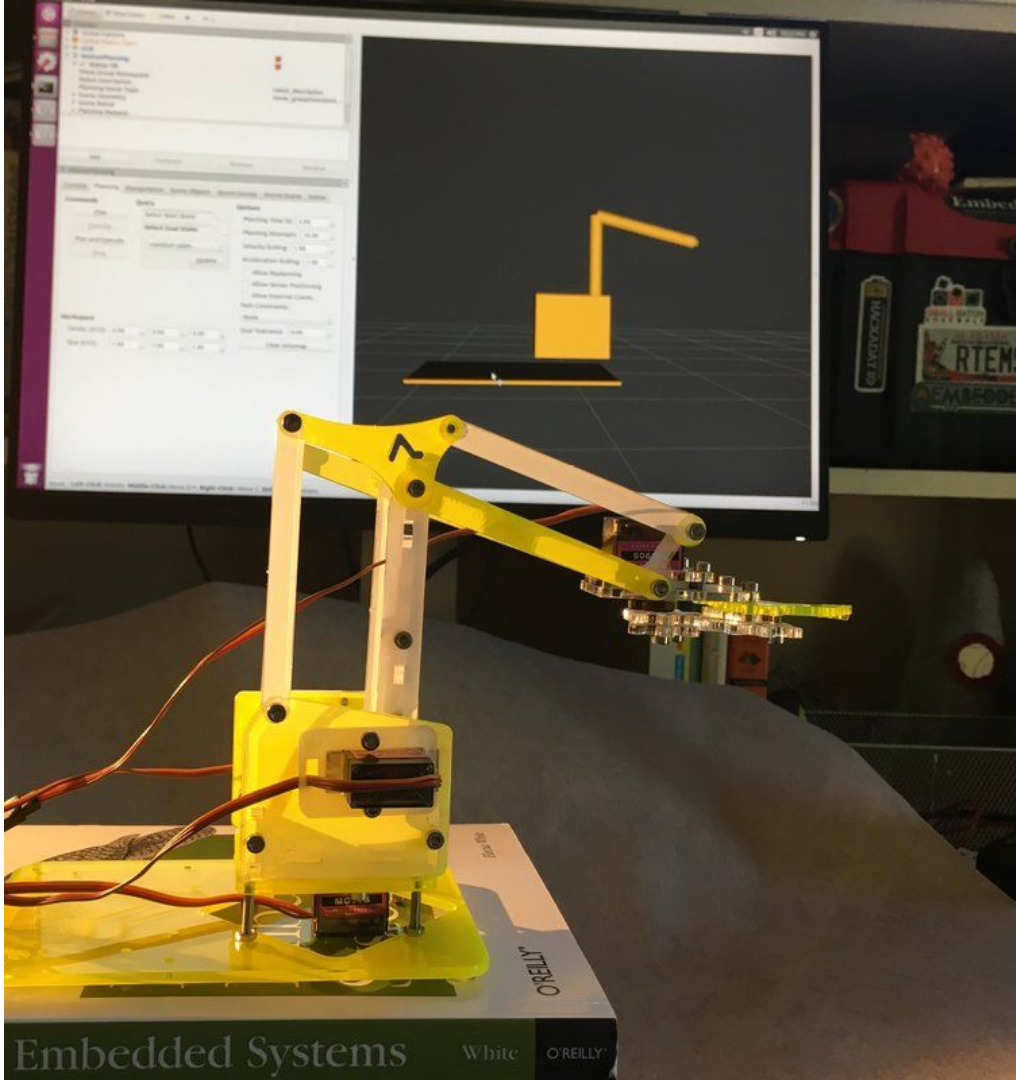
An entry into the
NVIDIA Jetson Developer Challenge
By
Elecia White
@logicalelegance, @embeddedfm

Category: Geek Out

The origin of the term “geek” has its roots in the circus freaks who would eat anything. Keeping that in mind, my entry for the NVIDIA Jetson Developer Challenge could only fall under the under Geek Out category.

I didn’t build a laser finding cat, a typing robot, a dictation machine, or an educational tool to investigate machine learning and the Robot Operating System (ROS). I worked on all of them.

Sometimes you need to try everything on the buffet.



Goal: A Robot that Types

- Fine motor skills
  - Demonstrate a level of control over the robot and its surroundings; once you have a robot that can type, the robot can do a wide variety of other things.
- Keyboard independent
  - Use a non-fixed camera to locate the keyboard and find keys
- Refined motion control
  - Use the same camera to locate the end effector on the robot
- Machine learning to drive the system
  - Type words spoke by user
  - Type objects that are identified (not finished)

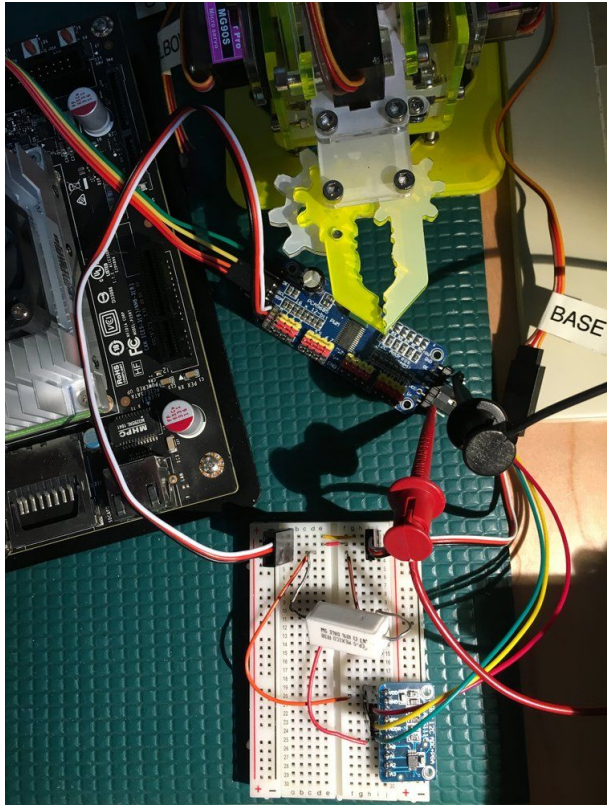
- Robotics
  - Arm control
  - Kinematics
  - Motion planning
  - Servo motor position calibration
- Machine vision
  - Homography
  - Object masking
  - Hough Circle Transform
  - Perspective transforms
- Speech recognition
  - CMU PocketSphinx
- Machine learning
  - ImageNet GoogleNet
  - YOLO
  - Tensorflow, Keras

Underlying Goal: A Robot that Types (and Teaches)

My real goal here is to learn and to teach.

I want to learn more about deep learning and machine vision. I want to understand robotics, from the electronics to complex motion control. I want to see if speech recognition works for me. I want to practice with embedded Linux systems.

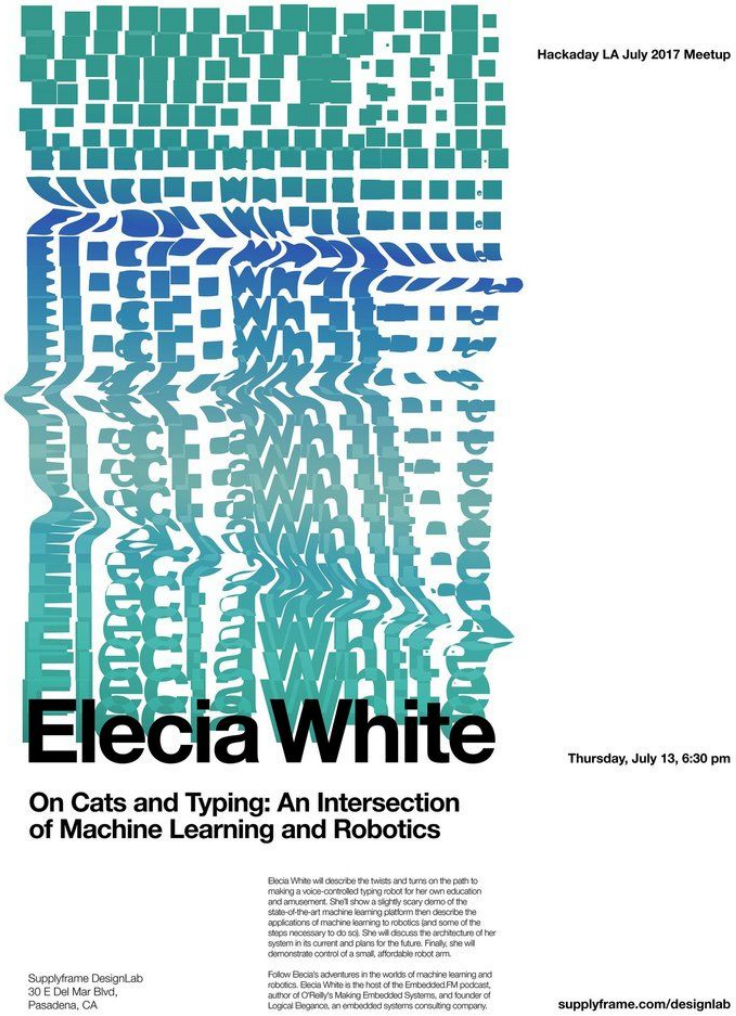
I also want to demystify these topics for other people. I want to make them easier to approach, putting up warning signs when implementation becomes difficult but showing why persevering is worth the effort. I place a fair amount of effort on making the project fun, even silly, in an effort to make both robotics and machine learning approachable. By combining a powerful board and a ridiculously underpowered robot arm, I'm showing that software is the critical part of the puzzle.



Pre-Existing Work

Before the challenge started on October 23, 2017, I had the NVIDIA Two Days to a Tutorial object identification working. I also built a small robot arm from a kit and got it to act cat-like as it chased a laser pointer. I setup ROS and got the motion planner and visualization system working. I even gave a presentation showing these off, called [On Cats and Typing](#).

Note that no typing actually occurred at that time.



Work Done During Contest

- Installed and experimented with YOLO, a multi-object recognition (I wrote about this.)
- Took Udacity Self-Driving Car, term 1. Attempted to run my advanced line finding and vehicle detection labs on the TX2 (sadly, failed to install miniconda).
- Implemented robot arm typing, experimented with end effectors. Determined fixed calibration was untenable. (I wrote about it.)
- Implemented computer vision project using homography to find the keyboard with a non-fixed camera and Hough Circle Transform to find the end effector. (Wrote about it.)
- Added python-based speech recognition based on Sophi Li's related blog post using CMU Sphinx's PocketSphinx project.