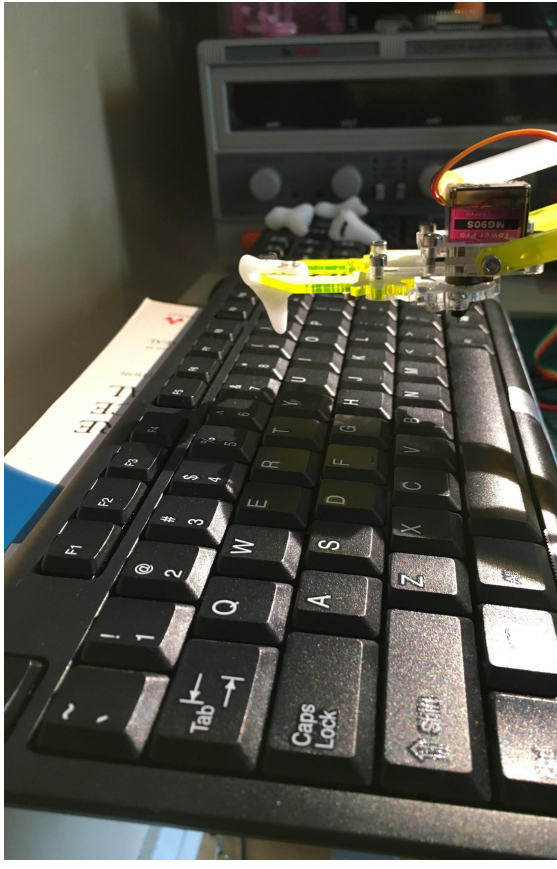
Judging Criteria: Design of Your Project

I fell in love with embedded systems the first time a motor moved under my software's control. No longer was I pushing invidiable bits from one place to another, my code was interacting with the physical world. It was magical.

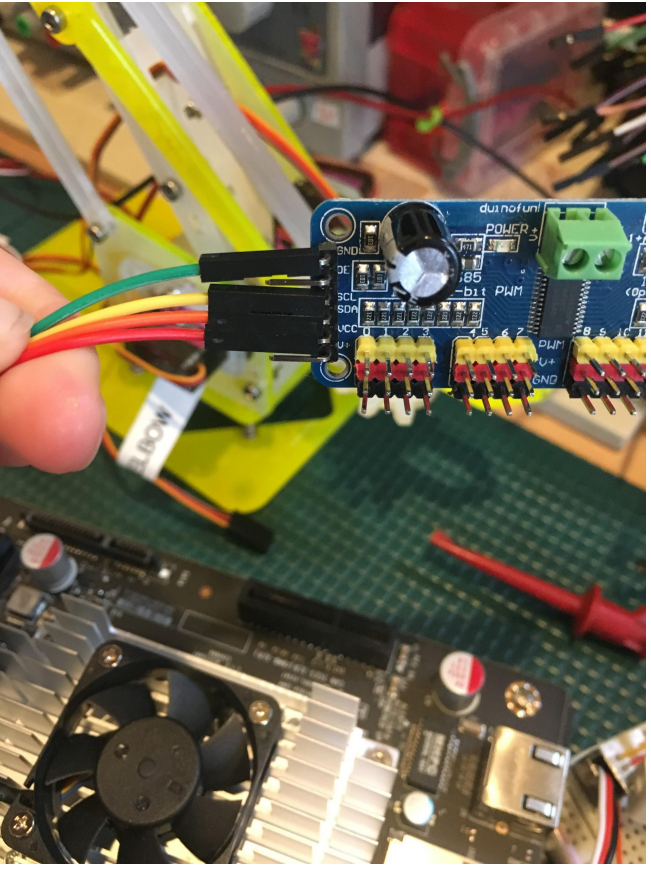
When people see the robot arm, they want to play with it. When it was chasing lasers, people wanted to control the laser, to try to push the arm into strange positions, to turn off the laser and watch its butt wiggle like a cat. Now, as it types, the system creates a sense of wonder and curiosity. The frailness of the robot arm enhances this wonder, generating laughter. That is my design. I'm pleased with it.

From a more engineering point of view, I'd like to think my project is well-designed, particularly for its primary purpose of exploration, learning, and teaching. The code and associated blog posts cover a wide range of deeply technical topics that I've tried to demystify.

At each stage, I documented what I did, including the struggles. My code is open and the system repeatable.



Judging Criteria: Functionality of Your Project



You've seen the video, I hope. You can see it works. The system types. It can hear you. Sometimes it types what you say, sometimes it types something hilarious.

So as a dictation device, it gets maybe a barely passing grade of a C- with the possibility of moving it up to a B with some additional work.

As an educational resource for myself, I'd give the project an A. I have learned so much in so many different areas. The TX2 has more to teach me. The robot arm gives me a direction and keeps me engaged.

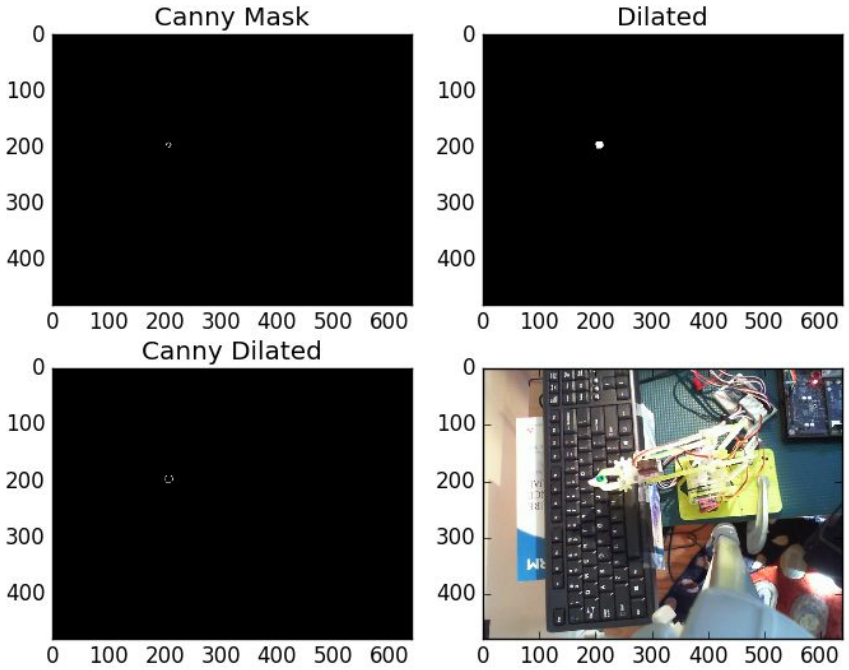
I'm optimistic that this project will be an educational resource for others. I hope it is inspiring others to try and to learn.

Judging Criteria: Impact and Scalability

We need more people to be interested in STEM. And then we need them to level up in machine learning, computer vision, and robotics.

This pathway is hard, showing the possibilities is the only way to make people interested, to get them engaged.

I hope that showing the TX2 connected to a cheap robot arm shows opportunities. Further, showing the struggle of getting there may inspire people to persevere in their attempts as well.



Resources Used to Make This Project

- Two Days to a Demo and the imagenet-camera
- Udacity Self-Driving Car term 1
  - TensorFlow
  - Keras
- Python OpenCV 3, book and tutorial
- Robot Operating System book
- YOLO: you only look once object identification
- Adafruit PCA9685 PWM Servo and I2C Python libraries
- MeArm library (with some modifications)
- CMU Sphinx
- Sophi Li's CMU PocketSphinx for Python post

Resources I Created Before Contest

- Github for Typeypt: test and laser_cat_demo
- My Arms! They Are Here!: Planning my project and exploring TX2's object identification
- The Sound of One Arm Tapping: Building the MeArm from a flat pack
- A Robot By Any Other Name: Breaking the project into pieces and goals
- Imagine A World of Robots: Exploring Robot Operating System
- On Cats and Typing: A status update on laser following and machine learning
  - What If You Had a Machine Do it (related podcast episode)
  - Related video of talk at Hackaday DesignLab Meetup
- Completely Lacking Sense: Adding current monitoring to the motor control

Resources I Created During Contest

- Github for Typeypt: prototyper
- What's Up With Ty: Trying to learning enough machine learning concepts to apply them to typing, adding YOLO to the TX2
- Pressing Buttons: Exploring end effectors and calibration problems due to using a $50 robotic arm
- Video of Ty typing "hello" for the first time
- Hand Waving and OpenCV: Using computer vision to solve calibration problems
- The Good Word About AI: Podcast with NVIDIA's Dusty Franklin about TX2
- NVIDIA Jetson Developer Challenge Video