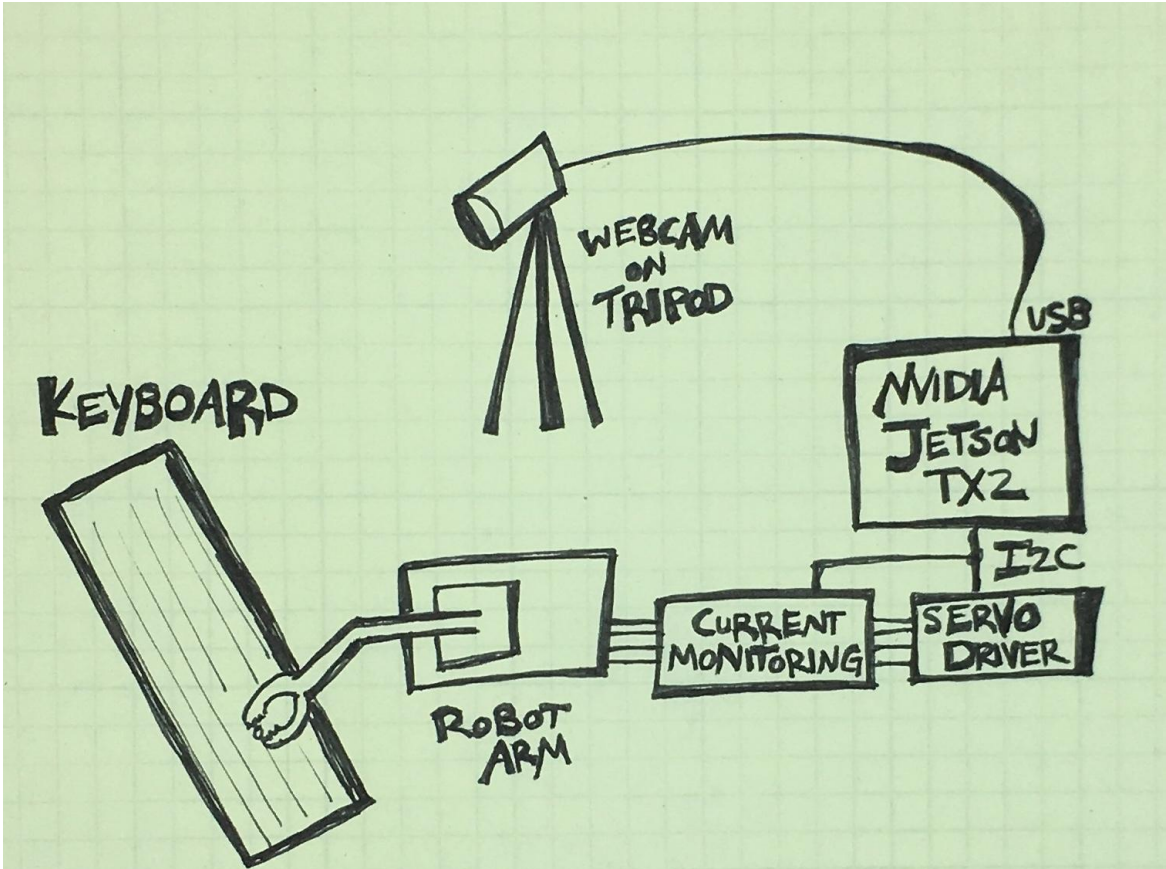
Work Still To Do

- Refine typing
  - Temporarily removed current control but need it to minimize repeated letters.
  - Better motion control, ideally via reinforcement learning on ROS Gazebo simulation.
  - Better visual feedback, likely by placing the camera in a more overhead position instead of off to the side
- Better speech detection parameters
- Object identification to typing: type what you see
  - Modify NVIDIA’s imagenet-camera to send the most likely result to python so Ty can type the object it sees.
  - Possibly do with with the multi-object identification system YOLO as well to build a word-story of the current view.

Hardware Diagram

- Robot ARM is the inexpensive MeArm.
- Servo driver is the PCA9685 16-channel, 12-bit PWM/Servo Driver.
- Current monitoring consists of ADS1115 16-bit, 4-channel ADC and four 4.7Ω resistors.
- The USB Webcam is a Microsoft Lifecam and provides both video for arm feedback and audio for speech detection.

For more about the hardware, see related blog post.



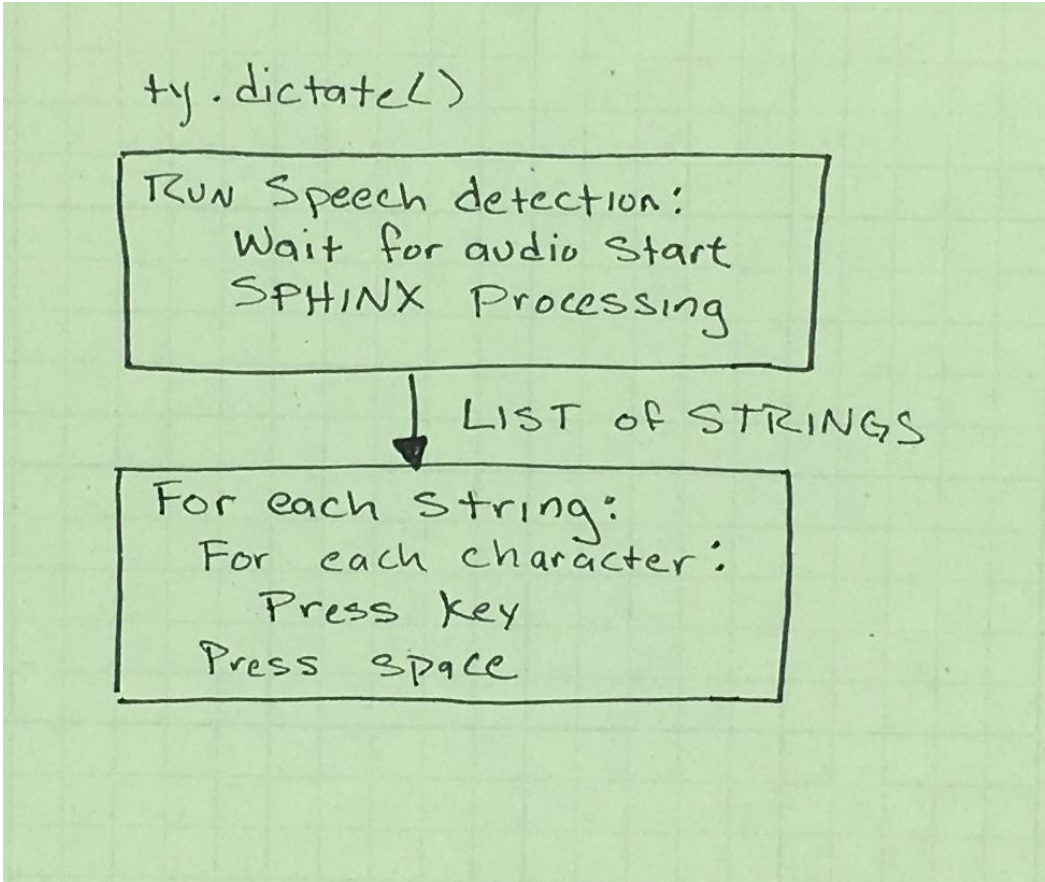
Software Diagram: Overview

At its core, the code is simple.

But it explodes quickly into complexity.

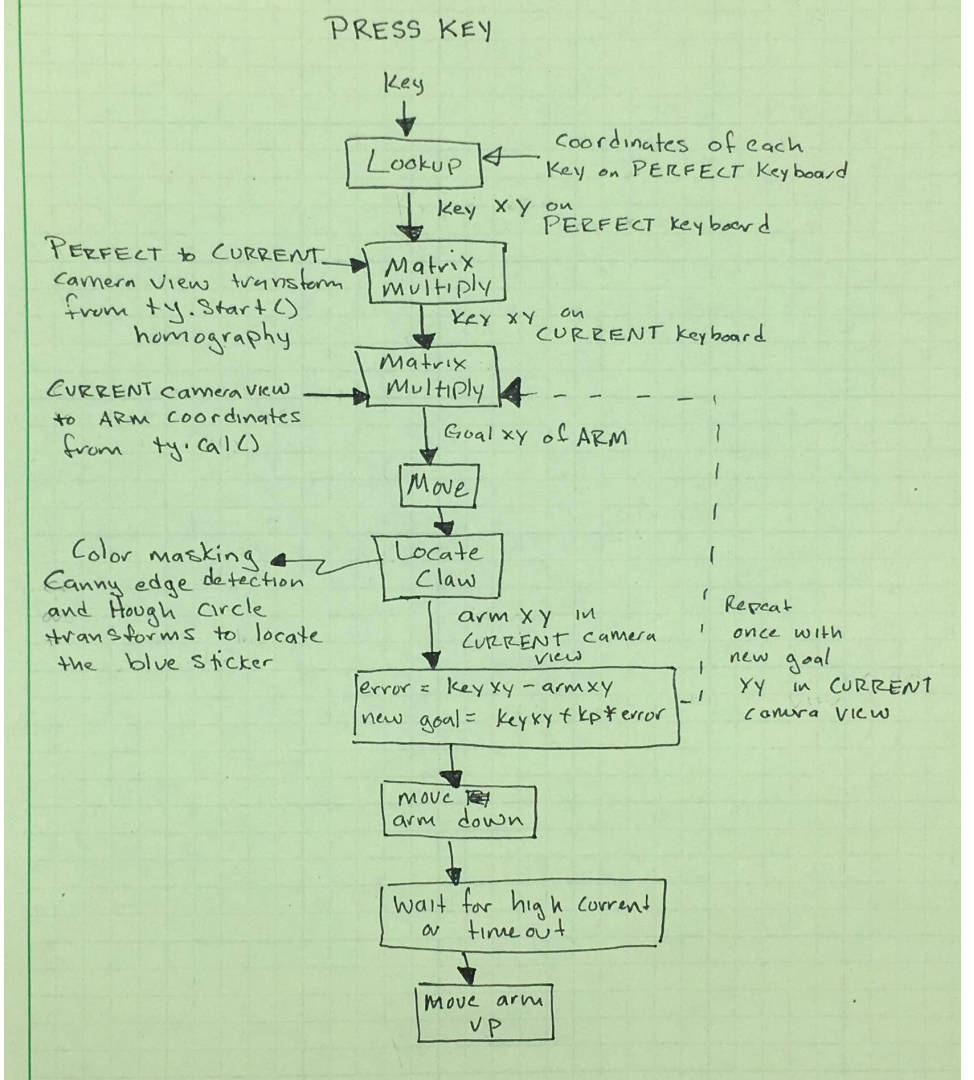
The speech detection piece is relatively off the shelf, using CMU pocketsphinx for python and set up according to [Sophi Li's blog post](#).

Speech detection provides a list of strings but things get interesting when you press a key.



SW Diagram: Press Key

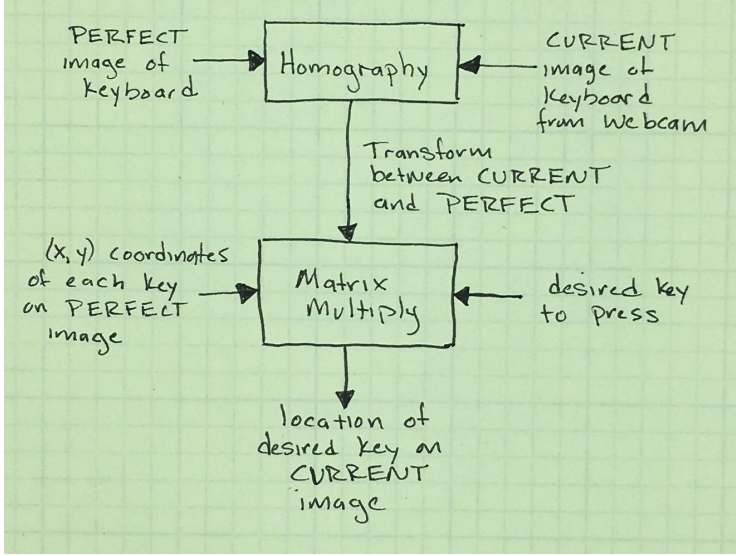
Upon starting the system, the camera system is calibrated with the command ty.start(). Then the arm is calibrated with ty.arm(). These feed into finding keys in the current camera view, locating the arm in the current camera view, and moving the arm correct distances given the current (and changing) system configuration.

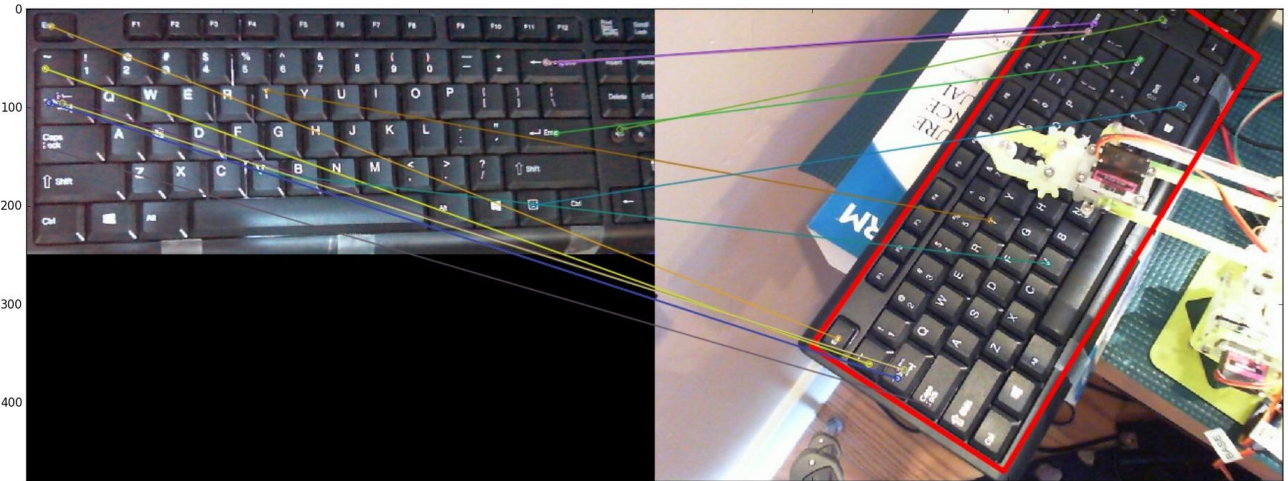


SW Diagram: ty.start()

The most important calibration step is ty.start. This finds the perfect keyboard (left) in the current view (right) via homography, a neat computer vision technique to locate objects in an image.

For more about the computer vision subsystem, [see my blog](#).





Judging Criteria: Use of AI and Deep Learning

I'll admit Ty doesn't currently use much deep learning directly in typing. I tried reinforcement learning to locate keys but the mechanical arm is too fragile for the training process (and my ROS model wasn't quite right). It does use a nearly absurd amount of computer vision, OpenCV is well optimized on the TX2. And it does use speech to text processing via PocketSphinx.

I intend to add the option of typing the objects reported by Dusty Franklin's Jetson Inference ImageNet-camera.

However, AI cannot be used alone, the system needs to *use* the intelligence. Plus, robotics is more fun anyway.