RELEVANT PROJECTS

Keyon Jerome

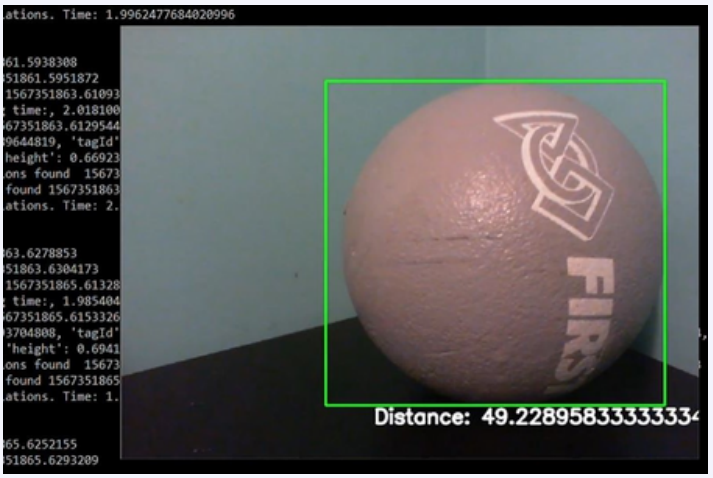
FRC COMPUTER VISION & ROBOT CODE

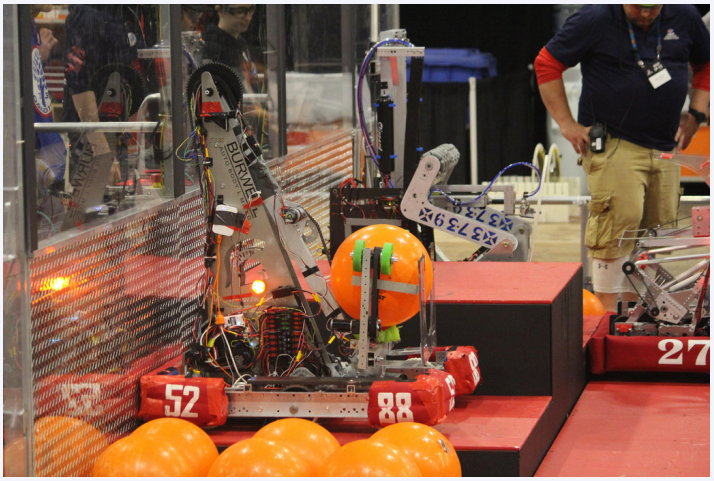
PYTHON | JAVA | TENSORFLOW

I worked on vision tracking for my robotic team's robot, as well as the robot's main code. My computer vision programs, running on embedded systems such as the **Jetson TX1** and **Raspberry Pi**, use video cameras and lights to detect field objects in real-time.

Robot Code | Computer Vision Code

By building a library of labelled images of each game field object, I was able to use **Microsoft's CustomVision.AI** on the **Azure** platform to develop a machine-learning model that could detect game objects. The machine-learning model exported to **Tensorflow** in **Python**. From there, I used computer vision techniques and frameworks such as **OpenCV** to find out where the game object was in 3D space, relative to the robot's camera.





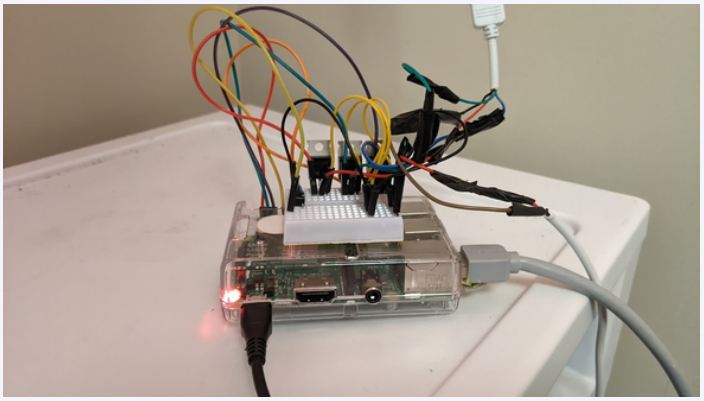
Finally, the result of the real-time image processing and calculations is sent over to the robot's control system via ethernet. From there, the robot's **Java** code is written to take in the result and use **PID** motion control to drive itself to the target.

Winner of the BOS Innovations 2019 Raspberry Pi Programming Contest

keyon.io
linkedin.com/in/keyonjerome
github.com/keyonjerome
keyon.jerome@uwaterloo.ca

ROS2 SMART HOME

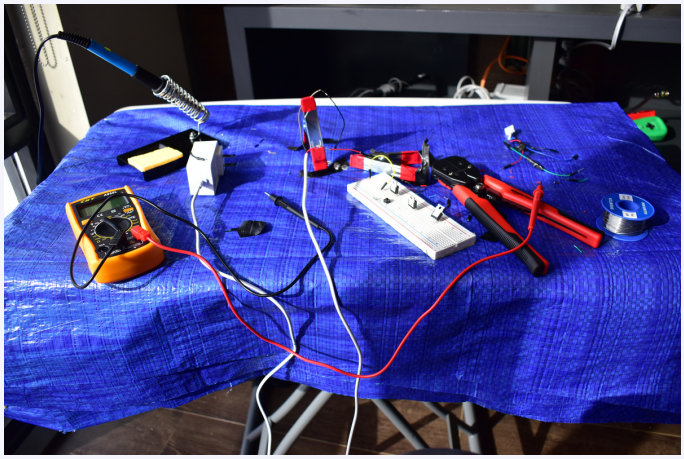
ROS2 Galactic | C++ | pigpio | rosbridge.js



I built an LED strip controller using ROS2 Galactic in C++ and the pigpio library. Running on a Raspberry Pi with the appropriate Ubuntu 20.04 RPi, I used pigpio (a Raspberry Pi GPIO C library) with PWM to control the LED strip's color intensity, with 3 logic-level MOSFETs for all three colors (RGB). Finally, I created a web app with rosbridge.js and Node.js to publish messages to the LED strip.

GitHub Link

I used two separate ROS2 nodes: one for defining a subscriber to `new_color`, which immediately sets the LED strip color (via pigpio) to the received color, and one "button_pub" node that defines and reads from a GPIO pushbutton. Depending on which color is next in the queue, a button press will immediately publish a new color.



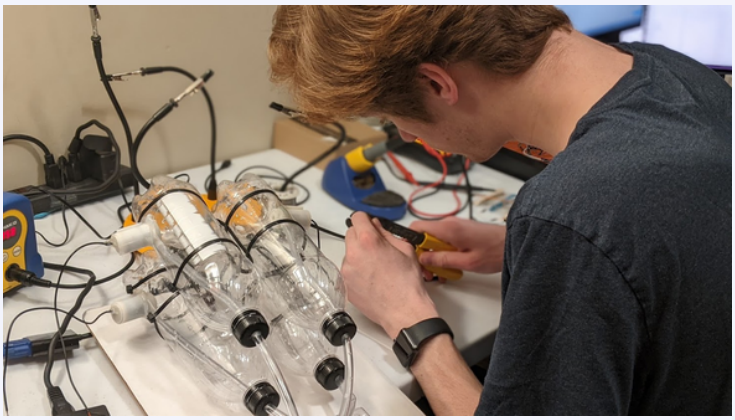
DrinkMaster

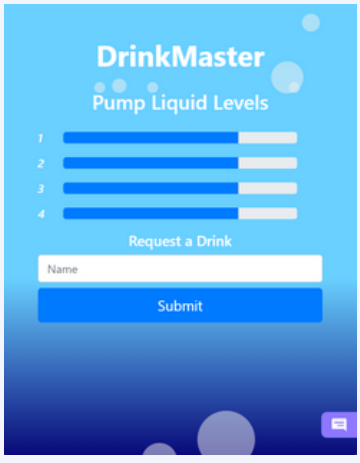
Raspberry Pi GPIO | Flask | Python
MakeUofT 2022

The DrinkMaster is best explained via our Devpost or demo video, but here's the 30-second version: it's a wearable machine that dispenses drinks at your fingertips. It's complete with a companion web app hosted directly from the machine itself, a real-time LCD screen for displaying the device status, and dispensing action from any of four different beverages at the touch of a button.

GitHub Link

The DrinkMaster is comprised of four 1L bottles connected to liquid pumps with tubes running down the length of your arm. These pumps are relay-activated and controlled by a Raspberry Pi 3B+. Each individual button on the DrinkMaster's glove controls one of these four water pumps. This Raspberry Pi simultaneously runs a DrinkMaster web app in Flask on the Raspberry Pi's local network.





keyonjerome.co
linkedin.com/in/keyonjerome
github.com/keyonjerome
keyon.jerome@uwaterloo.ca