Hi, this is Jonathan. Put this together because my report was a bit scuffed and probably doesn’t give you a good overview of the over-arching project.

I’ve put this together to the best of my knowledge, but a lot of the information is just based on my time fiddling with the project – I don’t know everything! So just take some of this info with a grain of salt.

If you need more information you can contact me at [jonjjt2000@gmail.com](mailto:jonjjt2000@gmail.com) or +44 7365916933.

### Things

Velocity control?

Max out the current/voltage used by the motors – it’s currently maxed at 70A – maybe can set to 100A

The motor is rated for 100A at 4400W (44V) -

<https://hobbyking.com/en_us/turnigy-sk8-6374-192kv-sensored-brushless-motor-14p.html?___store=en_us>

# Miscellaneous Information

## System Overview

Graphical user interface, diagram

Description automatically generated

## Coordinate Systems

Diagram

Description automatically generated

## H-bot System Start-up

A picture containing text, indoor, box, worktable

Description automatically generated

1. Plug in the blue PSU, ensure Estop is disengaged (by rotating it)
2. Ensure the PSU is switched off (the slider switch on the front cover)
3. Plug in the Fork connectors to the PSU (red to red and black to black)
   1. WARNING: do not plug in the forks while the PSU is on as there will be many sparks due to the large capacitors.
4. Manually move the striker to the middle of the task space on the x axis and near to the goal
   1. WARNING: on boot, the Odrive board will perform encoder calibration, if the striker is too close to the wall it will crash, and the Odrive will error code after putting significant strain on the table.
   2. GRAPHIC
5. If not already done, plug in the ODrive USB cable into the tower.
   1. In the 2020-2021 version of the project, the USB cable had a light blue USB isolator attached to avoid damaging the tower in the event of a short circuit. This was not used in the 2021-2022 version of the project.
6. Turn on the PSU
   1. WARNING: the ODrive start up encoder calibration will make the striker move forward and then back in the task space (ONLY Y AXIS).
      1. If the striker moves on a diagonal, the ODrive is probably throwing an encoder error (see Debugging Info (ODrive)). Best to solve this problem before proceeding further, otherwise the H-bot system will only operate with one of the two motors.
7. Table is ready, next step is to run the control software

## Control Software Start-up

### ODrive Motor Controller Server (Python) (RUN THIS BEFORE THE C++ PROGRAM)

In a terminal, run:  
  
*cd ~/Desktop/Air\_Hockey\_Project/Motor\_Control*  
*sudo python3 odrive\_server\_main.py*

The Control Logic side of the C++ Program assumes that the striker is in the top left corner of the task space (0,0 in the Motor Controller coordinate system) The program will try to move the striker to this position on start-up. If this fails, close the program and run it again.

There are two other Python programs in this directory

1. *odrive\_config.py*
   1. Running this program sets the ODrive parameters – you only need to run this if you reset the ODrive board
2. *odrive\_set\_idle.py*
   1. Running this program tells the ODrive to set both motors to idle. When *odrive\_server\_main.py* crashes, sometimes the motors are not set to idle, so you can’t move the striker manually. This was useful before as *odrive\_server\_main.py* did not move the striker to the top left corner on start-up, so this needed to be done manually each time.

### Machine Vision and Control Logic Program (C++) There are two versions of this program that use two different types of control logic.

See code breakdown for more information.

#### Hsinhua Lu Version

In a terminal, run:  
  
*cd ~/Desktop/Air\_Hockey\_Project/Machine\_Vision\_and\_Control\_Logic/Lu\_Hsinhua\_Version/build*  
*./cameracode*

#### Jonathan Tang Version

In a terminal, run:  
  
*cd ~/Desktop/Air\_Hockey\_Project/Machine\_Vision\_and\_Control\_Logic/Tang\_Jonathan\_Version/build*  
*./cameracode*

## Debugging Info (ODrive)

The control code usually won’t throw errors on its own. When it stops responding, usually there is an error being thrown by the ODrive motor controller board.

### Using odrivetool

More info here (it’s not amazing but this is the ODrive reference): [https://docs.odriverobotics.com/v/latest/fibre\_types/com\_odriverobotics\_ODrive.html?highlight=cpr\_polepairs#](https://docs.odriverobotics.com/v/latest/fibre_types/com_odriverobotics_ODrive.html?highlight=cpr_polepairs)

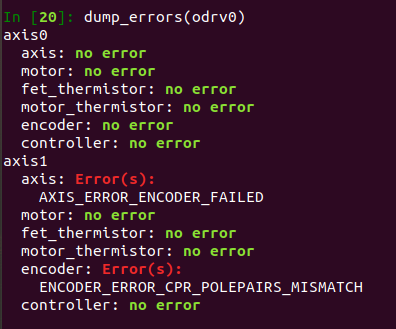
In a terminal, run:

*odrivetool*

and wait for the tool to find the ODrive motor controller board USB device.

#### Useful Commands

##### dump\_errors(odrv0)



Example of what dump\_errors(odrv0) Returns

This command shows what errors the ODrive has thrown, if any.

##### odrv0.reboot()

This command tells the ODrive board to reboot – This is equivalent to turning the ODrive board off and on again, hence it will perform the encoder calibration sequence as described in step 6 of the H-bot System Start-Up section. The same warning applies.

This is useful as flicking the PSU off won’t immediately turn off the ODrive board due to the high-voltage capacitors between it and the ODrive board. Using this command saves you the ~1-2 mins it takes for the capacitors to discharge if you need to restart the ODrive board.

#### Common Errors and Possible Causes

##### ENCODER\_ERROR\_CPR\_POLEPAIRS\_MISMATCH

This error is thrown when the encoder CPR (counts per rotation) and motor pole pairs are not matched up in the ODrive configuration to the appropriate values for each device.

This should be set up properly already –

CPR: 4000

Pole pairs: 14(? Maybe 7…. Can’t remember. But this is the number of pole pairs in the motor.)

For this system, the error tends to be caused by the motor and encoder bracket having been shaken loose – causing the encoder and motor to not be aligned properly. Check if either bracket is loose and tighten the fasteners. You need to remove the spools to get to those fasteners.

## Turning off the H-bot System

Turn off procedure, turn off the PSU or press the Estop. (Possible future work: move estop between capacitor and ODrive board in the circuit)

WARNING: DO NOT disconnect the forks from the PSU until the PSU voltage readout has dropped under 2V. There is a capacitor discharge resistor in the Odrive board but it is slow. Disconnecting the capacitors while they are charged may result in a short that will dissipate 80J of energy in very little time.

## Camera Software TCam Capture

This is an application that you can open with the white and blue icon in the taskbar.

You can adjust the camera focal length, exposure time etc. here.

Would avoid doing so unless the lighting conditions change or the puck/striker tracking or ArUco marker detection is consistently failing. You could also tune the HSV values in the code.

You ***DO NOT*** need to open this software to run the machine vision code.

## Pucks and Strikers

Use the 3” Pucks

# Known Issues

## Reduced Workspace (H-Bot Mounting Location)

## Camera and Table Shake

## Curved Striker Path

Due to the H-shape motor layout and position control mode, when the command for striker to move in diagonal direction is sent, two motor will both move in fixed acceleration curve (trying to reach max velocity). And the one moving less distance will stop before the other, causing the curve striker path.

## ArUco Marker Detection on C++ Program Start Up

### Camera Doesn’t Start on First Run of C++ Program

After turning on the PC, the first time you run the machine vision code, it will say 0 ArUco markers detected. Don’t know why, don’t know how to fix it – I just CTRL + C and run it again haha.

## Glare

## Lost Puck Tracking

## Trajectory Prediction Uncertainty on Bounces

Due to the smoothening effect of Kalman filter,

# Code Structure and Breakdown