January Status Report

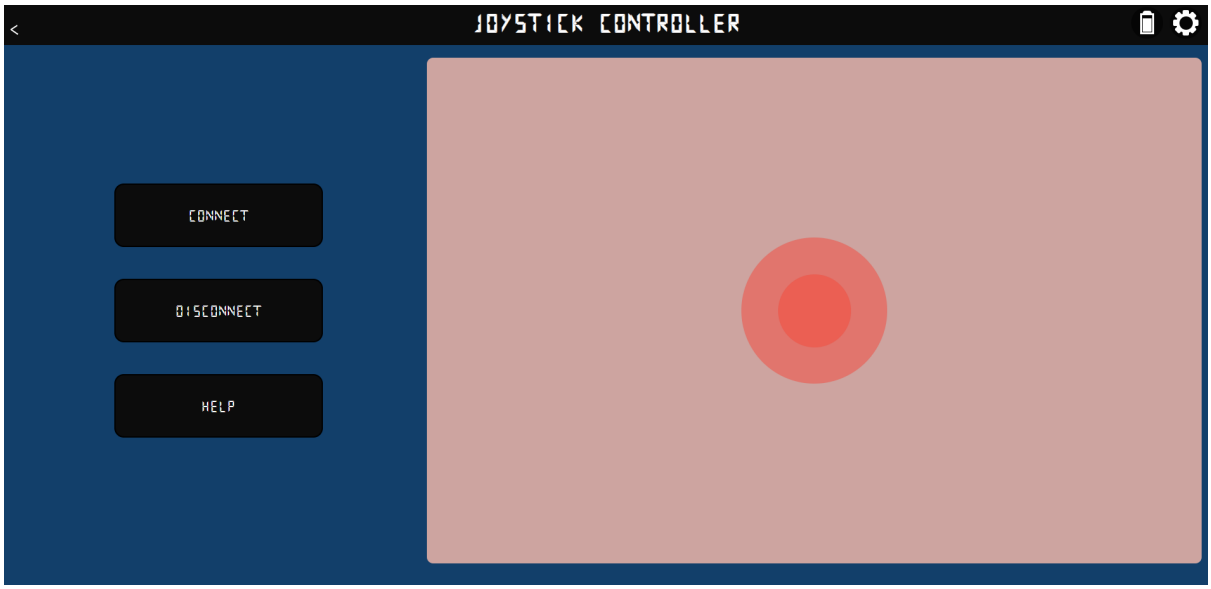
Robot Controller

I've been working on developing a joystick controller for the robot.

My most up-to-date implementation can be found here: https://lewistrundle.github.io/L4-Individual-Project/index.html

- Note that this does not currently load images (or have the developer mode mentioned below). For these, please see the old version here: https://lewistrundle.github.io/robot-controller/index.html

This can be tested on a mobile device (best to use landscape).



Joystick Implementation

To summarise how the current implementation of the joystick works:

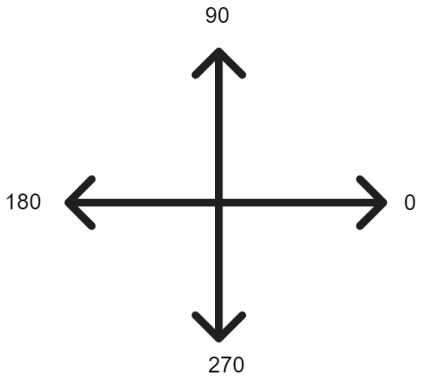
Every **0.6** seconds (this is as fast as I could get it without overloading the robot), the controller sends to the robot the angle the joystick is at. The code on the robot then takes this angle and uses a mapping to turn on the left and right motors to some degree.

The current angle-motor mappings are as so:

Angle	Left Motor	Right Motor

Note that the joystick has this directional system:

0	1	0
45	1	0
90	1	1
135	0	1
180	0	1
225	0	1
270	-1	-1
315	1	0
360	1	0



The only time the motors moves in reverse is when the angle is 270 degrees. This is because changing the direction of the motor is not always reliable and may fail due to a number of reasons.

There is also the problem that the left-motor on the robot seems to be faulty - it will only turn on if the right-motor is moving first. This has made it difficult to test the robot and is the main reason the joystick has not been fully implemented.

Developer Mode

While I was playing around with all of the possible functionality with the robot, I decided to keep everything I implemented in a separate page. This page can be accessed by clicking on the cogwheel in the top-right of the joystick controller page.

A possible future improvement to the project could be to include this page as some sort of “developer mode” as it allows the user to test different joystick implementations and functionality of the robot, such as uploading code. However, I believe this is out of scope of the current project.

Other Features of Joystick

Some features that still need to be implemented fully:

- Checking the battery of the robot: the battery percentage can be gotten from the robot; however I still need to display this number to the user.
- Help button: this will be done once I’ve written some documentation for the controller.

Track Design

I re-designed the track on Figma, this time including measurements.
https://www.figma.com/file/p5jf9avmuazf1W6wpArfBC/Track-Design?node-id=0%3A1&t=cJNbUIgawCNELzbC-1

I’ve designed it using the dimensions of an A0 sheet of paper so that it can be printed. I also included the design with A4 sheets of paper as well, incase people would like to assemble the track themselves.

I should note that this is not the final design. First a few things must be done:

- I would like to first implement and test a basic version of line tracking with the robot. This would allow me to understand more what thickness and colour the edge piece and middle line should be.
- I need to test how the Aruco markers can be detected by the robot (I played around with them a while ago but didn’t do any proper testing). Once this is done, the marker can be added to the design

Once these are done, the track can be exported and printed.

Next Steps

The first two things I'd first like to do are:

- fix the problems with the joystick controller, these are:
  - The input delay with the controller
  - Cannot quickly change direction of motors
  - The left-motor being possibly faulty
- Implement rough-version of line-tracking

There could quite possibly be many more problems with the joystick controller, however I would like to at least get a "somewhat" usable controller before moving on from it.

The line-tracking is also important to implement as this will allow me to test the track design.

Once this is done, I'd like to finalise the track design and get a usable version printed off so I can test the current implementation of the robot. As already previously mentioned, I would first need to:

- test the line thickness and colour of track
- decide the best way how the robot will self-navigate (i.e. will it simply just follow the central line or are there other ways?)
- test the Aruco markers

The next jobs are to:

- make documentation for the controller so that is can be easily used by anyone.
- make general improvements to both the joystick and line-tracking controllers.

During this I shall be working passively on my dissertation, making sure I have a draft copy for review at least 3 weeks before the deadline.

It is difficult to tell when I'll be able to implement object avoidance for the robot. However, this can be investigated whilst I am implementing line-tracking for the robot.