

**INTRO TO ROBOTICS JOURNAL**  
**COMP 20170**  
**GROUP 4:**

**TEAM MEMBERS:**

Mynah Bhattacharyya  
Luke Lynam  
Adam Lacey  
Jon Eckerth  
Ben McDowell

—

**Assignment 1: Obstacle Course Challenge**

- Use sensors to get to goal area
- Begin at start area
- Touch wall/lego box
- Follow a line
- Get to calibration area using the sonar, touch, light sensor
- Get to goal

General Roles:

Adam - ROBOTC code, group session notes

Mynah - Project manager, ROBOTC code

Jon - Build, EV3 kit management, parts resupply

Ben - EV3 Build(main)

Luke - EV3 Build, Robot C code

Timeline:

- Rough build- Complete(Ben+Jon)
- First Pass Code- In progress(Adam + Mynah)
- Start testing Monday
- Code & Build Adjustments

Code Main Procedure:

1. Move forward + k-turn (90ish degrees)
2. Find red line (move forward + turn 10)
3. Follow line (Course correct)
4. Calibration area (k-turn, park, reverse, u-turn)
5. Move forward till goal(Button press)

**Gantt Chart (Note: IP = In Progress)**

<u>Tasks:</u>	<u>Status</u>	<u>Person</u>	<u>Date Finished</u>
Base EV3 build	Complete	Ben, Jon	Wed 25/1
Try move fwd code	Complete	Adam	Thurs 26/1
Try manual turn code	Complete	Adam	Thurs 26/1
Attach touch sensors	Complete	Ben	Thurs 26/1
Software setup	Complete	Adam, Jon Mynah	Thurs 26/1
Initial code pass	Complete	Mynah, Jon	Sun 29/1
Test and debug/break up code tasks	Complete	Everyone	Mon 30/1
K-turn code	Complete	Adam, Mynah	Mon 30/1
Install gyro sensor	Complete	Ben, Jon	Mon 30/1
Find line code	Complete	Luke, Mynah	Tue 31/1
Follow line code	Complete	Adam, Mynah	Tue 31/1
U-turn code	Complete	Adam, Ben	Mon 30/1
Calibration area code	Complete	Adam, Ben	Tue 31/1
Final goal code	Complete	Adam, Ben	Thur 9/2
Put code together	Complete	Adam, Luke, Mynah	Wed 1/2
Test and debug	Complete	Everyone	Mon 13/2

**DAILY LOG 1:**

Mynah, Wed 25/1

We set up our groups after the lecture and started familiarising ourselves with our LEGO boxes. Ben & Jon started working on the basic build instructions. We got the base build set up quickly. Adam and Luke started working on setting up the ROBOTC software. I was able to assist between build and code while also dividing up tasks between the two main project pipelines- build and code.

**Obstacles:**

- I have a Mac so I can't use ROBOTC. I have a Windows laptop at home, which is bulky but will do- I will set up ROBOTC at home.
- Luke is on holiday for the next few days. We'll have to catch him up when he returns.

Total work time - 2hrs

## **DAILY LOG 2:**

Mynah, Wed 26/1

We met up 2 hrs early before class to get a head start on finishing our EV3 build. Ben and Jon stabilised the EV3, added the touch sensor and light sensor. Meanwhile, Adam helped me with setting up ROBOTC on my windows laptop. Together, we were able to write basic code for moving the EV3 forward and also responding to a button press by turning. However, we don't have control over the angle of the turn yet or motor synchronisation. We tried to get this problem resolved in class by asking both TA's and peers, but were unable to fix it- problems to be solved when we get more familiar with the code documentation.

### Obstacles:

- Angle turn control
- Motor synchronisation
- Preventing stopping drift towards the left
- How to use button presses as stopping conditions in loops

Total work time - 4 hrs

## **DAILY LOG 3:**

Mynah, Sun 29/1

Adam had taken the robot home for testing writing code, but was unfortunately occupied this weekend, so I and Jon decided to try and write some code over a discord call. We tried to get the nMotorSync method working, but unfortunately it was not supported in our current version of ROBOTC. We were able to instead find the setMotorSync method, as well as some useful documentation online on line following and colour sensors, so we used that to write a first pass of the whole code. In the process we learnt a lot, mainly:

- How to break up the obstacle course into discrete code parts/procedures
- How to modularise code into functions using void(although this later did not work)
- Increasing familiarity with ROBOTC methods and functions

We will further test our code on Monday and see if the bot accepts functions using "void" instead of "task".

### Obstacles:

- No bot for testing!
- Old documentation-> not sure what functions to use

Work time -

Jon - 5 hrs

Mynah - 6 hrs

#### **DAILY LOG 4:**

Mynah, Mon 30/1

The whole team was able to reconvene today to work on the bot together. When we put the first pass of code through the EV3, it rejected both the functions with parameters and the setMotorSync methods, so we are due for a major code rewrite. Instead of modularizing code into discrete tasks called by main, we are instead doing everything procedurally in main, with commenting and pseudo code construction for clarity. We put in the gyrosensor as well and now we have much more control over the angle of our turns, which simplifies a lot of the k-turn & U-turn functionality. Adam and I knocked out the code for that quickly. We still have to figure out how the light sensor works properly - the first pass code had many logical errors. I will be taking the bot home and reading documentation on light sensors and reading documentation on light sensors and reading in colours for the code. Adam & I will be on a discord call together for the first part and I will work on it afterwards.

#### **Obstacles:**

- Old code + function on Brightspace documentation not working
- Colour sensor methods finicky, should not rely on ambient light

#### **Total work time**

- 2 hrs in school
- 4 hrs (Mynah) at home
- 4 hrs (Adam) at home

#### **DAILY LOG 5:**

Mynah, Tue 31/1

Most of the team (besides Jon, who had work), were able to meet today. We managed to get a major part of the code rewrite done. Luke and I worked on the followLine and findLine code, and we were able to get that working semi-successfully with some bugs on the turns and stopping conditions. Adam and Ben worked on the code for calibrationArea, and were able to test it against the walls of where we were working to debug. Breaking up the procedures and making up a template of the pseudocode, while testing discrete parts on independent source files was very useful. Right now, since we couldn't quite figure out functions, we are working off of the assumption that we will compile all our code into a main method in one file later, after working on discrete parts and getting them working. Later on, we can modularise it further. Later this evening, we did a group Discord call, and were able to get all the bugs sorted in our followLine code, and get some of findLine out of the way as well.

#### **Obstacles:**

- Lack of understanding of functions affecting modularisation
- Lack of actual board for testing exactly how the bot will respond

#### **Total Hours Worked:**

- 3 in person (Mynah, Luke, Adam, Ben)
- 3 on call (Mynah, Luke, Adam)

**DAILY LOG 6:**

Luke, Wed 01/2

The full team was able to make it to the lab today. In the lab we were able to find out how to use functions. Adam and Ben then worked on getting all the commands for the robot into one file. They then converted the file to run them using functions instead of just having everything in main. Meanwhile Mynah, Jon and I tested the files on the robot that we've been coding. Because we had access to the actual board we were able to more thoroughly test the program and find problems in code we previously thought were perfect. When testing the robot we found that one motor moves faster than the other causing it to turn into the wall when commanded to move straight. Another thing we found was that when following the line it turns too much for just following the line. For these problems we'll have to increase/decrease the speed of one of the motors and reduce the angle the robot turns when following the line.

Total Hours Worked: 2 hrs (in lab)

**DAILY LOG 7:**

Luke, Thur02/2

The full team was in. We all worked on the code for the robot in the lab. We ended up with a lot of issues with the code for the robot such as only giving it commands to turn left and it moving forward. We ended up losing track of code that worked and code that didn't so we set up a git repository so we wouldn't have this problem again. Along with this the robot itself was starting to act up. We at first thought it was just out of battery but ended up having to reset it as the screen remained frozen even when charged for a while.

Total Hours Worked: 2 hrs(in lab)

**DAILY LOG 8:**

Jon, Fri 03/2

Over the weekend I worked on the code for the robot. To improve its performance I rewrote the line finding algorithm.

The improvements are based on an alternating search pattern. Before, the robot was searching for the line in an area of 120 to -120 degrees from the current heading.

It first searches the entire area to its right, before taking a look to the left. This has the effect that when the robot is to the right of the line, it loses a significant amount of time before it can even find the line to its left.

The new algorithm instead implements an increasing alternating search pattern. It first looks 10 degrees to the right, before turning -10 degrees to the left before turning 20 degrees to the right and so on. This has the effect that the robot can find the line almost equally as fast in both directions, the tradeoff being that the robot has to turn more often. But since the turns it has to make during the line following are quite small, the overall time it takes to follow the line is reduced.

Total Hours Worked: 1.5 hrs (at home)

**DAILY LOG 9:**

Jon, Wed 08/2

Implemented a function to move the robot from position 2 to the line, after it has done its k-turn.

It is just a simple move function with a stop condition, that checks if the robot has reached the line.

Total Hours Worked: 1/2 hr (at home)

**DAILY LOG 10:**

Luke, Thur 09/2

Only Adam didn't make it today. We just fine tuned the code today. We tested the robot on the actual board multiple times, tweaking the code slightly each time. By the end the robot made a successful full journey around the board. The only thing we didn't fix because we ran out of time was one of the last turns. The angle the robot turned was too wide and the robot ended up knocking over the book at point 4. However despite this it still made it to the end. Later we will work on fine-tuning this so we can complete the course.

**DAILY LOG 11:**

Jon, Sun 12/2

Now that the code is completely integrated with little more to do, we continued fine tuning the parameters. We met up and recreated the board as close as possible utilising pictures we made of the board during the lessons. The robot is extremely sensitive to the initial conditions, so when it starts with a slight angle at position 1, this angle will compound and results in the robot hitting the wall at position 2 at an unexpected location, sometimes making it miss the Line entirely.

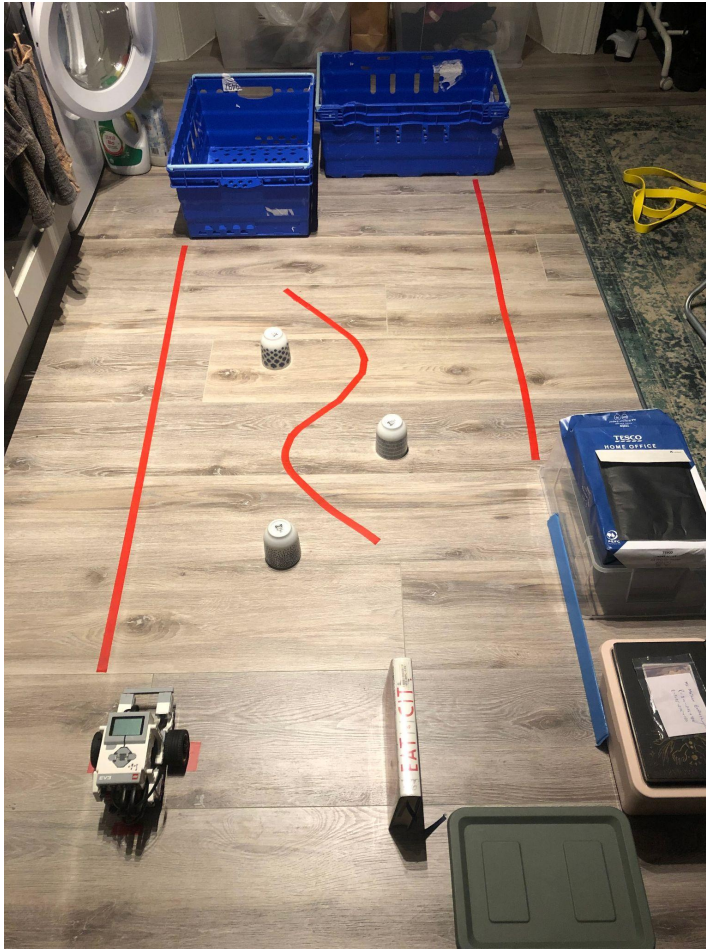
To mitigate that, we exactly marked to position of the robot to be able to recreate the initial conditions reliably.

This enables us to better fuzz the parameters and find the optimal values for the robot to complete the board reliably.

We tweaked the speed of the motors, the angle the robot turns when following the line and distances the robot moves when reaching the different positions.

In the end we were able to get a video of the robot completing the board reliably, without any issues.

The only thing left to do now is the recreation of this on the official board in the lab.



Picture of board recreation at home for testing.

We were able to get one video of it working at home, but will hopefully be able to test it on the board in school as well.

Total Hours Worked: 1 hrs (at home)

Mynah, Jon

### **DAILY LOG 12:**

Luke, Wed 15/2

Everyone was there today. We got the board and recorded the video of the robot completing the course.