LAB 4 – Sensors and Distance 2

Catch-up Lab

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

TASKS

1. Sign in to the lab and sign out a robot. Attendance goes towards CA.
2. Read the Webcourses lab 3 material
3. Rebuild the Robot from last week and add Bump and Ultrasonic Sensors
4. Build the layout in diagram below.

Lego Box

Book

Book

1. Use this lab to catch up on the tasks from last week that you did not get completed.
2. Research the following statements to help control rotations (both for EV3 + NXT).
   * 1. nMotorEncoder[motorC]
     2. nMotorEncoderTarget[motorC]

**motor**[motorB] = (**rand**() % (max-min)) + min; // set motorC to a random number in the range: [min, max]

1. Display the power/speed and battery life on the LCD display. Research how to do this.
2. Write a program to make the **bump sensor** detect a collision and reverse back for 1 second. To test your program you should start it from different locations and ensure that it always goes back to the roughly place.
3. Write a program to make the **ultrasonic sensor** detect a wall and reverse back for 1 second. To test your program you should start it from different locations and ensure that it always goes back to the roughly place.
4. Write a program that goes from Red zone to point 5 using bump and ultrasonic sensors.
   1. Use bump sensor for the first obstacle, and ultrasonic sensors for the second two obstacles.
5. **New Task:** Write a program that uses the **Ultrasonic sensor only** to get from the Red zone to point 5. However, if at any time the **bump sensor** is touched, the robot will stop and reverse back for one second.
6. Upload the code and the logbook to Webcourses.
7. Tidy up and sign back in the robot.

GIT + GITHUB

1. Show your tutor a local Git repository and a pull and commit you have made to it. See this week’s notes on how to push and pull from a Git Repo using Windows commands.

IF YOU ARE FINISHED…

1. Write a program that uses the **motor encoder** to control the distance the robot covers. Research and write behaviour that makes the robot go forward for one full wheel rotation, then stops. Measure this distance.
2. Use this time to research some tasks from next week’s lab.
3. How might you use **functions** to make this code more modularized?
4. Write a program to make the **light sensor** detect a line and reverse back for 1 second.
5. Implement a simple line tracking program to follow a line forever.