UNIVERSITY OF TECHNOLOGY SYDNEY

FACULTY OF ENGINEERING

Subject: ***48623 – Mechatronics 2***

Assessment #: ***3***

Assessment Title: ***Embedded integration***

Student Number:	
Student Name:	

Declaration of Originality

The work contained in this assignment, other than that specifically attributed to another source, is that of the author(s). It is recognised that, should this declaration be found to be false, disciplinary action could be taken and the assignment of the student involved will be given zero marks. In the statement below, I have indicated the extent to which I have collaborated with other students, whom I have named.

Statement of Collaboration

Signature

Marks

LCD	/2
Control	/2
Sweep	/3
Wall follow	/7
Navigation	/10
Quiz	/6
TOTAL	/30
Assessment II Mark	/30

Aim

The aim of this assessment is to integrate the Assessment 2 with a robotic simulation tool to carry out basic robotic navigation tasks in a maze.

Requirements

You will be required to write source code for the micro-controller to carry out robotic tasks in a hardware in the loop simulation set up. The map is 20x20m and will be given to you (See Figure 1). The white spaces are empty (obstacle free) and dark parts are walls (obstacles). You will be provided with MATLAB interface and full documentation of the simulator.

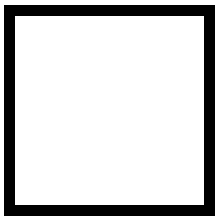

Figure 1: Map (box.png)

You are required to carry out the following operations through the Arduino kit.

- **Main menu**

  When powered ON, the first line of the LCD should display the Student ID. The second line should display "Main menu". The robot should be designed to perform the following operations: Control, Sweep, Wall Follow and Navigation. User should be able to select the above options through the second line of the LCD. The cursor should be in the second line. The Menu should be able to cycle through the 'DOWN' button. A particular mode should be able to select through the 'SELECT' button.

- **Control**

  In the 'Control' mode, the system should take in the user input from the button press to drive the robot in the simulator map. LEFT and RIGHT buttons should rotate the robot in left and right directions respectively. UP and DOWN buttons should drive the robot forward and backward respectively. The robot should drive/rotate continuously on the chosen direction while the button is being pressed. The second line of the LCD should display "Control". By pressing the SELECT button, the users should be able to exit this mode and return to the main menu.

- **Sweep**

  Once in the Sweep mode, the second line of the LCD should display "Sweep". Pressing the UP button should start the sensor autonomously to do a 360^{o} clockwise (CW) scan of the surrounding area. The robot should be stationary. The robot then should find the nearest object and rotate itself counter clockwise (CCW) to face towards the identified shortest distance to the object. The robot will be tested in the given map with some random obstacles (see Figure 2).

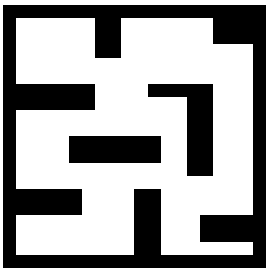

Figure 2: Map (map3_1.png)

Once the task is completed, pressing SELECT button should allow the users to exit this mode and return to the main menu.

- **Wall follow**

  Robot should be in the box map (Figure 1) and when this option is triggered the robot needs to find the nearest wall using the 'Sweep' functionality. The robot should then autonomously move towards the wall and stop at 2 meters from the wall. The robot must then travel parallel to the wall while maintaining the 2m distance. The robot needs to keep moving while maintaining the 2m distance from all the walls until the user press the UP button to stop the robot (wall follow as well as sensor scanning). The second line of the LCD should display "Wall follow". By pressing the SELECT button, the users should be able to exit this mode and return to the main menu.

- **Navigation**

  The second line of the LCD should display "Navigation". The robot will be placed in a map with a given start and goal locations (in global coordinates). The start and goal coordinates can be given in MATLAB simulator as in the following example (Follow the Maze Colliders simulator guide).

  ```
  randomGoal = false; %% Spawn random goals on map
  randomPose = false; %% Random beginning pose of robot

  g1 = [17 15]; % GOAL [x y]
  g2 = [17 15]; % GOAL [x y]

  Pose = [2 4 15]; %Pose [x y theta]
  ```

  Note: Use the same coordinates for g1 and g2 to simulate one single goal in the map.

  There will be objects placed randomly in between the 'start' and 'end' points based on simulator map so that a straight-line navigation is not feasible (Figure 3).

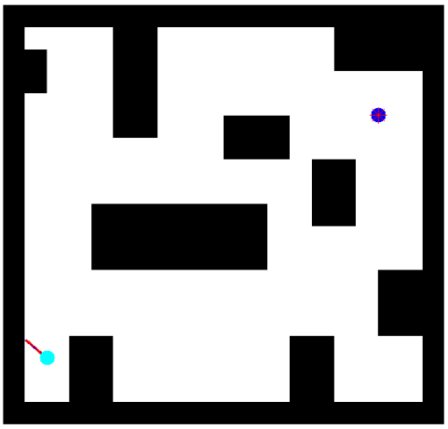

Figure 3: Example Map (map3_2.png)

Then the robot needs to:

- perceive the area through sensor scanning to locate any obstacles.
- move towards the goal avoiding obstacles using the shortest/fastest path.
- stop when it reaches within 0.5m of the goal (you will have access to the global coordinates of the robot through the simulator) and display 'Finished' in the first line of the LCD.
- **at any time**, by pressing the SELECT button, the users should be able to exit this mode and return to the main menu.

- **Important:** In all scenarios, the robot must not collide with any walls.

- **Quiz:** You will be given a quiz to complete as part of the assignment which you need to answer based on your system implementation.

Notes

- The only equipment permitted for this assignment is:
  - 1x Arduino Uno
  - 1x16x2 LCD Shield
  - MATLAB robot simulator
- The MATLAB simulator will be provided with the user guide for you to test your implementation and complete the assignment. Make sure to test your implementation in different Maps (Set of maps are included in the simulator).
  - MazeColliders simulator: https://github.com/cpu191/MazeColliders
  - MazeColliders user guide: https://github.com/cpu191/MazeColliders/blob/master/MazeColliders_User_Guide.pdf

- **YOU ARE NOT PERMITTED TO USE THE ARDUINO LIBRARIES AND FUNCTIONS (Refer to ban list on UTS online)**; however, you are permitted to use the LiquidCrystal.h library for the LCD, Serial.h for serial communication, and progmem library (including F()) for better memory handling. You are permitted to use the C standard libraries and AVR i/o and interrupt libraries.
- You may also use the C++ String variable types for this assignment.

- You may also use these variable conversion function part of the C++ variable and Arduino function:
  - https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/tofloat/
  - https://www.arduino.cc/reference/en/language/variables/data-types/stringobject/
  - https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/toint/

- A well commented complete PDF of your source code should be submitted in TurnItIn by the due date. And the same code needs to be uploaded as buildable ".ino" file separately for testing the functionalities.
- In the PDF document you must include the specific serial monitor setting you used. By default, we will be marking with the baud rate at 9600 and line ending as newline.

Marking Scheme

	Item	Mark Allocation	Marks Awarded
LCD	• Correctly displays the student ID in the first line • LCD displays correct information on second line allows user to choose specific modes of operation and the mode name clearly should display during operation.	0.5 1.5	
Control	The simulation robot should accurately move inside the map based on the user button inputs and it has to be easy to control.	2	
Sweep	The robot should scan (CW) and find the closest object The robot turns (CCW) to face the identified object (shortest distance)	2 1	
Wall follow	Robot should follow the closest wall without collision Robot should maintain the set distance while continuously navigating until the UP button is pressed	3 4	
Navigation	Obstacle detection and avoidance Reaching within 0.5m of the goal, marks are based on the completion time as below. Top 10 students with shortest times will receive 5 marks. The student who are within 20% more than the top band timing will receive 4 marks. The student who are within 21% and 50% more than the top band timing will receive 3 marks. The student who are within 51% and 80% more than the top band timing will receive 2 mark. The student who are above 81% more than the top band timing will receive 1 mark.	5 5	
Quiz	Answer the quiz question comprehensively based on your work	6	

Support and Assistance

Support and assistance for this assignment will be available by posting questions on the "Tutorials and Assignments" forum on UTSOnline. This forum is monitored electronically and as such will have the same response time as a direct email. Please use the forum so that other students may benefit from the answers given. Face to face support is available during the lecture and/or tutorial timeslot. Please email to make an appointment.

Rules for Submission

Due Date: 6th October 2020 at 11:59pm

Code

The code must be submitted through the UTSOnline submission area before **11:59pm** of the due date. Students must submit their main code in a single buildable ".ino" file. Students should also submit (UTSOnline) this code in a PDF document with the appropriate instructions attached. It should be in **PDF format only**.

Assessment

Your code will be accessed running it in the simulator with different maps and based on the performance marks will be given to each functionality completion. There will be an online quiz to assess your knowledge of the system design. This quiz will be on UTSOnline and must be completed on the day of the due date (00:00 am to 11:59 pm).

Students with difficulty meeting assessment requirements

Students who experience **significant** difficulty, or anticipate that they will experience significant difficulty, in meeting assessment requirements must submit an "Application for Special Consideration form" (available at http://www.sau.uts.edu.au/assessment/consideration/online.html) to the Registrar **before** the due date of the assessment item. Significant difficulty means

i. Serious illness or psychological condition.
ii. Loss or bereavement
iii. Hardship/trauma

Note also that students may apply for special consideration because of illness or other circumstances (**not work related**) beyond their control. The "Application for Special Consideration form" has a section that must be filled in by a doctor, counsellor or other relevant professional authority. A medical certificate alone is not adequate and will not be accepted.
Note that it is up to the students to provide adequate information about their circumstances. University staff will not chase additional information and the Subject Coordinator has the right to reject applications that lack sufficient information.

It is the student's responsibility to contact the Subject Coordinator to find out what action has been taken and to obtain details of any additional assessment required or learning and assessment special arrangements. For further details please refer to section 4.6 of the "Coursework Assessment Policy and Procedures Manual".