UNIVERSITY OF TECHNOLOGY SYDNEY

FACULTY OF ENGINEERING

Subject:	***48623 – Mechatronics 2***
Assessment #:	***4***
Assessment Title:	***System 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

Display	/3
Wall follow	/4
Find Water	/5
Put out Fire	/5
Return home	/7
Fastest Navigation	/6
TOTAL	/30
Assessment III Mark	/30

Aim

The aim of this assessment is to integrate the Assessment 3 tasks with the robotic simulation tool to carry out advance 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 robot is placed at a starting pose = [2 4 0] in the **map4_2a.png** map. It will have 2 main goals coloured in blue and red. The blue goal symbolises a water bucket and the red goal symbolises a fire. The robot needs to wall follow to **first find the water bucket** and then find the fire to put it out using the water bucket as quickly as possible. Once the fire is extinguished, the robot must take the shortest path back to the starting position.

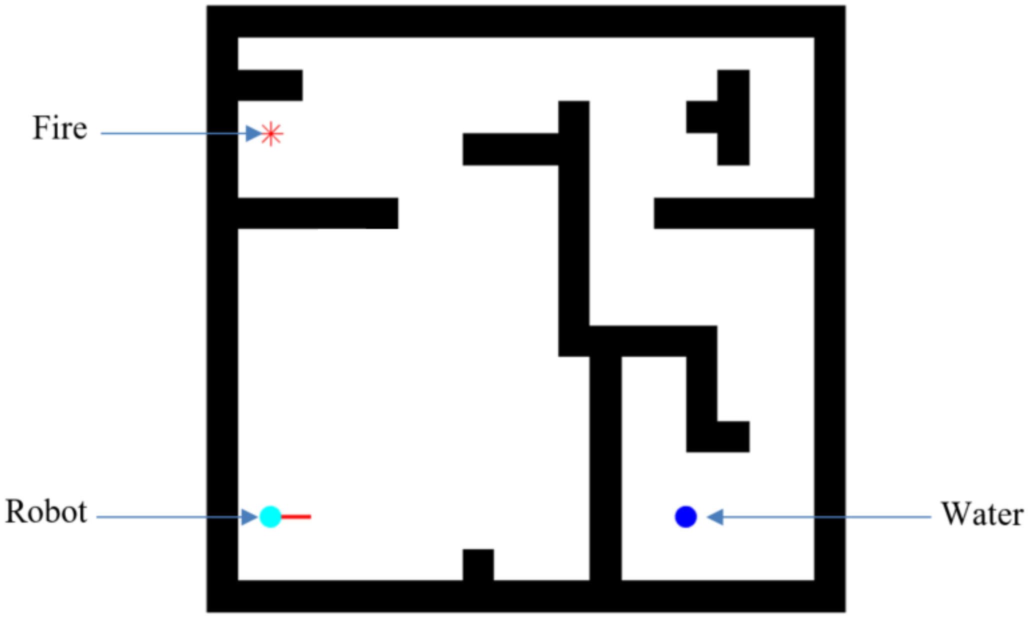


Figure 1: Map (map4_2a.png)

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

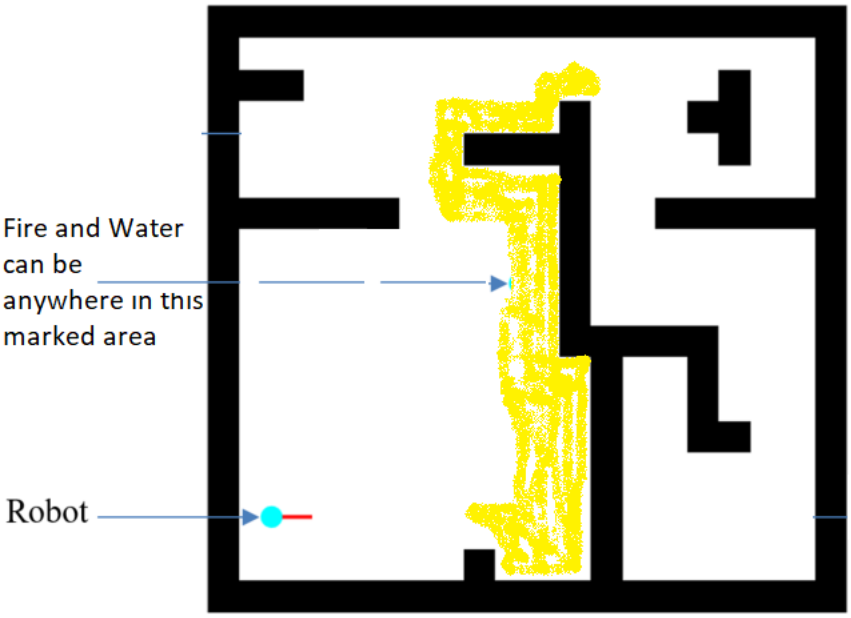
- **Display**

  When powered ON, the first line of the LCD should display the Student ID. The second line should display a timer with "00:00" (first 2 digits represent minutes and second 2 digits represent seconds). Once the SELECT button is pressed the autonomous **right wall follow** should start and the time should start to update by counting the mission time (the timer needs to update every second). Once the mission is completed the timer should stop counting and display the total mission time.

  - The current objective of the robot should also be displayed on the second line. This will change based on whether or not the robot has completed certain tasks of the mission. The objectives are: Finding water: the LCD should display a "W" character to indicate it is searching for the water.

- Finding fire: the LCD should display a "F" character to indicate it is search for the fire.
- Returning to start position (home): the LCD should display a "H" character to indicate it is returning to start position.
- Complete mission: the LCD should display a "C" character to indicate it has completed the mission.

- **The mission:**
  1. The robot must start when SELECT is pressed, transmit the start command to the simulator.
  2. The robot **must find the wall on the right** and wall follow to find and reach the water bucket.
  3. The robot must then wall follow to find and reach the fire.
  4. Once the fire is reached within 0.5m, the fire is considered to be extinguished. Then the robot must return to the starting position (within 2m) following the shortest path back.
  5. Once the robot has reached the starting position (within 2m range), the robot needs to transmit the close command and update the LCD to indicate it has completed its mission.
  6. **The whole mission needs to be finished within 6 minutes**. If the simulator is still running at 6 mins, markers will stop the simulation to evaluate the performance. The students will receive two attempts and the higher mark of the two attempts will be recorded as the final demonstration marks.
  7. **In order to make the simulation simpler, fire and water will be placed only in the area highlighted in the following figure**.



- **Note:** If the robot discovers the fire before collecting the water bucket nothing should happen until it discovers the water bucket and puts out the fire. Once the robot goes to the correct goal the goal will disappear from the map and will receive a feedback message from the simulator to the Arduino.

- **Important:** In all scenarios, the robot must not collide with any walls. Collisions will result in loss of marks.

- **The robot will need to be within 1 - 2 units (m) away from the walls at all times.**

Notes

- The only equipment permitted for this assignment is:
  - 1x Arduino Uno
  - 1x16x2 LCD Shield
  - MATLAB robot simulator
- Use the sensor on the robot to avoid obstacles and navigate through the map to find the goals. The MATLAB simulator will be provided with the user guide for you to test your implementation and complete the assignment.
  - 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 Canvas)**; 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++ 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. There shouldn't be any difference in the 'ino' and 'pdf' submissions. **In case they have dissimilarities, it may incur heavy penalties.**

Marking Scheme

	Item	Mark Allocation	Attempt 1	Attempt 2
Display	• Correctly displays the student ID in the first line. • Counting the timer and updating it every second smoothly. • Correctly displaying the current objective	0.5 0.5 2		
Wall follow	The simulated robot should accurately navigate a substantial distance inside the map without colliding on walls. Each collision will result in a loss of 1 mark The robot should be at least 1m away from the walls/obstacles.	3 1		
Find water	Exploring and reaching the water bucket. One mark will be penalised for every metre (one unit) away from the water bucket, in case the robot fails/crashes.	5		
Put out fire (Must have first found water to receive this mark)	After finding the water, exploring and reaching the fire position. One mark will be penalised for every metre (one unit) away from the fire position, in case the robot fails/crashes.	5		
Return to Start Position **(Must have found both water and fire to receive the full marks)**	Two options: Return to start through a shortest path navigation algorithm (shouldn't be the wall follow algorithm) For the students who chose to return to start using the wall follow algorithm can only receive a maximum of 3 marks. The robot signals completion within 2m of the starting position. Two marks will be penalised for every metre (one unit) away the robot is from the starting position. Maximum distance is 5m away from start position. **You must transmit the close command to count as complete.** Other navigation scenarios will be considered by the discretion of the marker.	7		
Fastest navigation	Mission completed within 3 minutes – 6 marks After 3 minutes, every 30s is penalised with 1 mark	6		

**** If any student could not complete the whole task, they have the option of demonstrating parts those they have completed with reduced marks. As it is a complex mapping, the marks are decided by the discretion of the markers.**

Support and Assistance

Support and assistance for this assignment will be available by posting questions on the discussion forum on Canvas. 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 timeslots through zoom or Teams. Please email to make an appointment.

Rules for Submission

Due Date:

31st May 2021 at 6pm for the Mondays 6pm- 7:30pm class in the CB11.10.403 lab

31st May 2021 at 7:30pm for the Mondays 7:30pm- 9:00pm class in the CB11.10.403 lab

2nd June 2021 at 12pm for the Wednesday 12pm- 1:30pm class in the CB11.10.403 lab

Code

The code must be submitted as a **PDF Document** and in a **single .ino file.** Both files should have the exactly same content. Students must submit **all their code** into Canvas (via two submission links for pdf and .ino file types) by the due time (as given in the above 'Due Date' section) and before your demonstration (we will start marking after the due time based on your submitted code).

Demonstration

Students doing the subject on campus: We will mark based on in class, face to face demonstration for all the students. You will need to set up your system and get it ready for marking at the beginning of the class and your Canvas submission ready for downloading. We will be randomly approaching the students and mark the work. During marking, you will be asked to download your code from Canvas and upload it to your set up. You should demonstrate during your allocated class time.

Students doing the subject online: This option is only open for the students who are overseas, in another state or medically recommended not to be on campus. Your code will be downloaded from Canvas and tested in the markers hardware. Any special instructions need to be communicated with the marker.

Students found using other student's code will receive no marks for this assignment and have disciplinary actions taken against them by the university.

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".