CG2271: Real-Time Operating Systems

Project

LumiNUS Team Number: _____ HP Number: _____

The aim of the project is to add some excitement to this module and allow you to develop an RTOS-based system beyond the structured labs. The project is expected to be completed at your own time, beyond the lab slots. You are free to make use of the Makers Lab to develop your project.

The aim of the project is to design a RTOS-based robotic car that will be controlled through an Android App. The robotic car must be able to fulfil the following features:

1. Establish a WiFi connection with the Android App
2. Receive commands from the Android App and execute the correct response
3. Move the car in multiple directions.
4. Control the various LED's according to the car's status
5. Play different sounds/tunes according to the cars status.

Grouping: The project can be completed in groups of 3 / 4 (preferably 4).

Your project group MUST be formed with other students in the SAME Lab Group.

Assembly Instructions:

1. The assembly of the chassis is quite intuitive so should be manageable by everyone.
2. The H-bridge motor drivers are similar to what you have used before in EPP2. Their interface details can be found here: https://www.pololu.com/product/2130
3. The interface of the WiFI module and the Android App Interface will be elaborated through a video tutorial.

Requirements Checklist:

A. WiFi Connectivity

Requirement	Level of Achievement
1. Develop a User Interface Button to establish WiFi connectivity with the Robot	
2. Robot must respond with TWO LED Flashes at the Front (Green LED's) to indicate that the connection has been established.	
3. Robot must play any unique tone sequence to indicate that connection has been established.	

B. Motor Control

Requirement	Level of Achievement
1. The robot must be able to move in all FOUR directions, Forward, Left, Right and Back.	
2. The robot must be able to perform curved turns while moving.	
3. The robot must stop all movement if no command is being sent.	

C. LED Control

Requirement	Level of Achievement
1. The front 8-10 Green LED's must be in a Running Mode (1 LED at a time) whenever the robot is moving (in any direction).	
2. The front 8-10 Green LED's must all be lighted up continuously whenever the robot is stationery.	
3. The rear 8-10 Red LED's must be flashing continuously at a rate of 500ms ON, 500ms OFF, while the robot is moving (in any direction).	
4. The rear 8-10 Red LED's must be flashing continuously at a rate of 250ms ON, 250ms OFF, while the robot is stationery.	

D. Audio Control

Requirement	Level of Achievement
1. The robot must continuously play a Song tune from the start of the challenge run till the end.* There should not be any break in the song even if the robot is not moving.	
2. When the robot completes the challenge run, the robot must play a unique tone to end the timing.	

*You are free to select any Song Tune. For this test, you must play the actual audio clip of the song and demonstrate that you are able to replicate a similar tune using the buzzer.

E. Self-Driving Ability

Requirement	Level of Achievement
1. A "Start" button on the App must activate the robot to perform self-driving	
2. The robot must be able to go straight for at least 60cm and perform a U-turn back to the starting point.	
3. The robot must stop by itself without any remote control.	
4. The LED's and Audio are required to fulfil the requirements specified in Part C and D.	

RTOS Architecture Minimum Requirements:

The architecture should have a minimum of 4 tasks and 1 Interrupt

tBrain: Decode the data from the Serial Port and perform the necessary action

tMotorControl: Control the Action of the Motors

tLED: Control the LED's

tAudio: Provide Audio Output

Serial_ISR: The Serial Data coming from the BT06 device. The serial data **MUST** be captured through the use of Interrupts.

This is a general guideline. You can have more tasks if you wish but not less.

You are free to decide the way in which the tasks will communicate and synchronize with each other. You must ensure that shared data is protected using appropriate RTOS constructs.

Global Variables:

You can declare OS constructs as Global. However, task synchronization and passing of information between tasks should be performed with OS tools and techniques. Usage of Global variables should be minimized.

RTOS Code Submission:

Please zip up only the files with your implementation. You do not need to submit the RTOS library files.

Please upload your code to LumiNUS <Project Code> Folder with the format 'TEAM-XX.zip'

LumiNUS Video Submission:

Each team is to submit a 2-3 min video about their entire journey doing this project. Videos longer than 3 minutes will not be graded and will be given **ZERO** marks. The marks distribution for the video is as shown below.

Creativity – 8 marks

Technical Overview – 8 marks

Total = 16 marks

Please upload to LumiNUS <Project Video> Folder with the format 'TEAM-XX.mp4'. Other video formats like .mov are also acceptable.

*** If you are unable to upload your file to LumiNUS, due to its size, please upload it to a shared drive or as a listed/unlisted YouTube video. You can then upload a document with the link. ***

Grading Criteria:

Item	Marks
Fulfilment of Requirements (16 x 1)	16
Challenge Run 1 Leaderboard Rank	30
Challenge Run 2 Leaderboard Rank	30
RTOS Code Submission	8
Video Submission	16

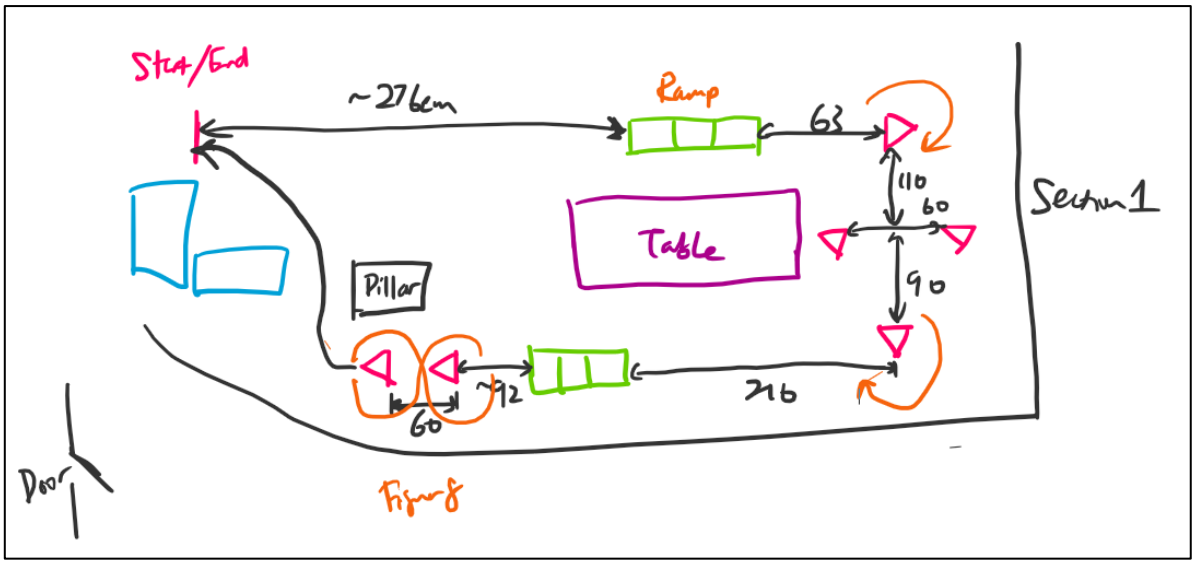
Total Marks for the Project = 100

Contribution to the Final Grade = 40%

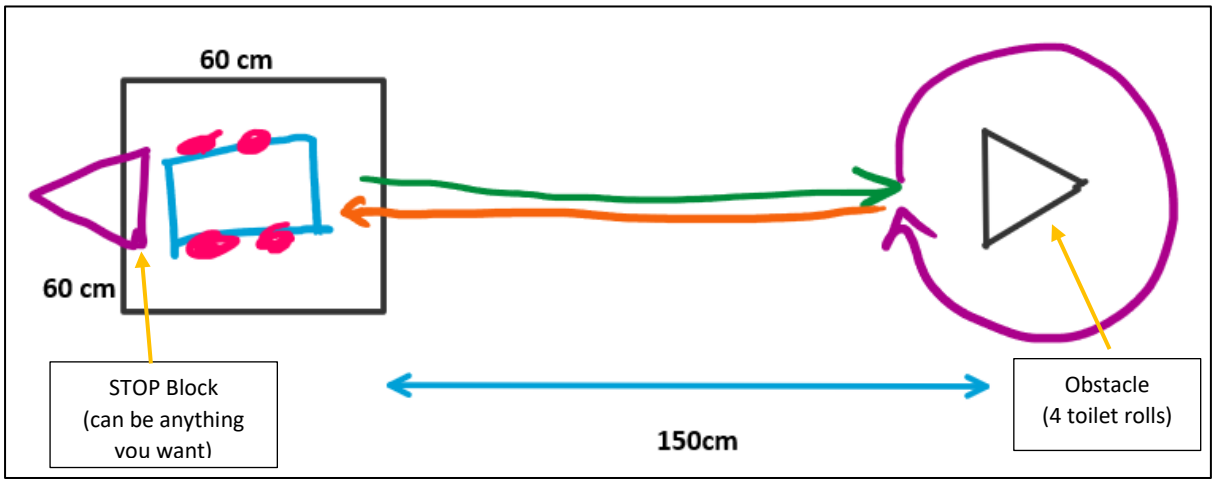
CHALLENGE RUN

The challenge run will require you to control your robot and navigate it through a simple maze while fulfilling all the requirements in the checklist above. The actual challenge run maze is still being finalized, so stay tuned for an announcement on that.

Challenge Run 1: Remote-Control Mode



Challenge Run 2: Self-Driving Mode



- In the Self-Driving mode of operation, there is no tape or indication on the floor to indicate the position of the obstacle.
- Students are NOT allowed to place any markers on the floor to indicate the position of the obstacle.
- You can develop your solution by integrating appropriate sensors to your design.
- Hardcoding the distance to stop is strictly NOT allowed.
  - This will be verified with the obstacle placed a random distance from the robot.
  - The distance can be in the range of 90cm – 150cm.
  - There are no marks for this verification. However, if the verification fails, you will automatically be disqualified from Challenge Run 2 and you will get 0 marks for it.

***** Each Team is given a budget of $40 to purchase additional components to fulfil the requirements of Challenge Run 2*****

The integration of the sensor for obstacle detection MUST only be done using the KL25Z.

Obstacle

The obstacle is to be built using FOUR toilet rolls taped together as shown here:

To ensure consistency, the diagonal of the obstacle should be within 10 – 11cm. This will be verified by the tester before your Challenge attempt.

***** You are free to cover the obstacle with any type/colour of paper. *****

IMPORTANT POINTS TO NOTE:

- The Leaderboard Ranking will be based on how fast your robot is able to traverse both the challenges.
- There will be Penalty timings imposed if certain events happen, e.g. hitting the wall/block, not stopping within the box.
- Your Final Leaderboard rank will only be known after all the groups have completed the challenges.
- Each group is given **ONLY TWO ATTEMPTS** for each challenge run. The second attempt must be taken **immediately** after the first attempt. You will not be given any additional time in-between attempts.
- The **BEST** timing out of the 2 attempts will be taken.

Challenge Run 1:

Attempt	Timing	Hits	Ramp1	Ramp2	Final Timing
1					
2					

Challenge Run 2:

Attempt	Timing	Hits	Stop within Box	Final Timing
1				
2				

***** SUBMISSION DEADLINE FOR ALL PROJECT RELATED MATERIAL: 19th November 2021 (11:59PM) *****

THE END