TI-RSLK A\* Pathfinder Rover

# Short Project Description

This project turns the **TI-RSLK** kit into a fully autonomous path-finding robot. Using an **HC-SR04 ultrasonic sensor** on a **Servo**, the bot first *explores* an unknown grid, labelling every cell as **walkable, blocked, or unknown** while updating a live grid on the **SSD1306 OLED** display. In *route* mode, the robot will end the best path to reach the end of the grid (4, 4); the robot runs an **A\*** **search** in real-time, renders the optimal path on the OLED, then drives the course, using the **Motors**, until it reaches the target and halts.

# Original Conceptualization

The original concept was inspired by lab work on robot operation and a desire to merge them with embedded systems skills developed during the semester and the A\* Algorithm I learned and developed for RIT’s Intro to AI course (CSCI-331). Early sketches called for a simple wall-follow routine, but a design review pivoted the idea toward *dynamic* mapping plus classic heuristic search so the robot could adapt to changing environments instead of memorizing a fixed maze.

# Obstacles Faced

* **A\* Algorithm Implementation** – While I previously developed the algorithm it needed to be adapted from Python to C and be designed for use with the other components of the project.
* **Explore and Route Functions** – The Route function couldn’t work as intended until the Explore function was developed to handle different scenarios.
* **Servo Angles** – During development, the MSP432 board short circuited and burned out. After receiving a new board, the servo needed to be recalibrated and the PWM for rotating the servo to the left, right, or center needed to be readjusted.
* **Dead-ends and Blocked Paths** – The Explore function needed to be able to handle paths that resulted in a dead-end or blocked paths. This involved different grids where a path would lead to a dead end and have a no other alternative path. The robot would need to back track to a previous cell and find a new route.

# Testing & Verification

1. **Servo and Ultrasonic Sensor** – To test the servo and ultrasonic sensor, I used different cardboard boxes at different distances to determine what would be the best distance for detection and decision-making.
2. **Clock Delays and Motor PWMs** – To test and verify the Clock Delays and Motor PWM, a mock grid was made to determine travel distance and turning speed.
3. **Dead-ends and Blocked Paths** – To test and verify the dead-end and blocked paths logic, a mock grid with different types of dead-ends and blocked paths was made.
4. **Explore Function** – To test and verify the decision logic and the grid display in the Explore Function, a full 5 by 5 grid was made using cardboard boxes with different paths.
5. **Route Function and A\* Algorithm** – To test and verify the Route Function and the A\* Algorithm, a simple and complex grid were made so that the robot could differentiate between different possible paths and calculate the optimal route for the robot to take.

# Pictures or Videos

# Possible Improvements with More Time

* **Implement User Defined Points** – Instead of having the robot find a path to reach the endpoint of the grid (4, 4), the user could provide coordinates for the Robot to travel to for more dynamic usage.
* **Combine the Explore and Route Functions** – This would also allow the robot to have dynamic replanning, allowing the robot to update the path of the robot should an obstacle appear.