Robotic Vehicle in A Maze
IOT - Winter 2024/25

Adan, Muhammad, Rand

The robot is designed to navigate a maze from start “S”, to goal “G” while the user builds the maze blocks in real time based on what he creates in the GUI.
The system will send the maze to the robot as a matrix filled with “S”, “G”, “F” for free way to pass and “W” for a wall. Then the robot will find the optimal way for the solution using Astar algorithm.
The maze can be changed dynamically: User can add a wall that is not in the initial maze that he built in GUI, and the robot will search for another way to pass.

Constraints:

- Block size should be about 30cmX30cm.
- Minimum maze size is 3X3 not included edge walls.
- The maze should has one start point and one end point.
- Each of start and end point should be surrounded by 3 walls.
- The maze shouldn’t contain a 2x2 free blocks.
- The maze must contains at least one path from start to end point.
- The dynamic wall shouldn’t be in the column and the line that there intersection is a way for the “G” (end point).
- The maze shouldn’t contain two consecutive turns.

Robot Connections:

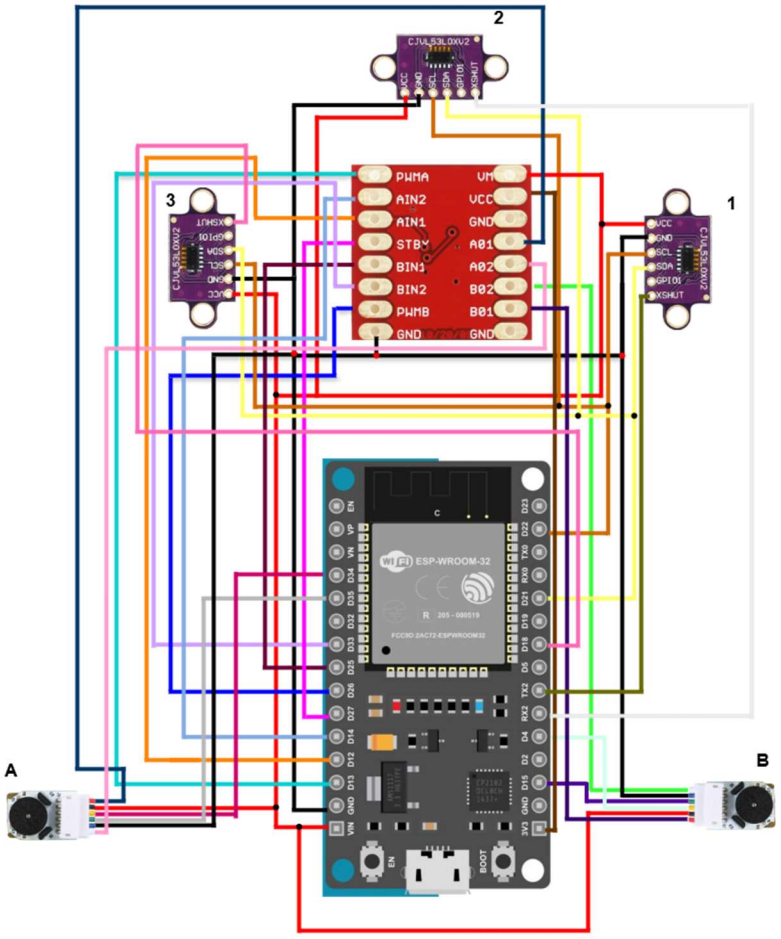
Connections of motor A - Left:
M1 -> A02 in TB6612FNG
GND -> GND
C1 -> D35
C2 -> D34
VCC -> VIN (5V)
M2 -> A01 in TB6612FNG

Connections of motor B - Right:
M1 -> B02 in TB6612FNG
GND -> GND
C1 -> D15
C2 -> D4
VCC -> VIN (5V)
M2 -> B01 in TB6612FNG

Connections of sensor 1
VCC -> VIN (5V)
GND -> GND
SCL -> D22
SDA -> D21
XSHUT -> TX2

Connections of sensor 2
VCC -> VIN (5V)
GND -> GND
SCL -> D22
SDA -> D21
XSHUT -> RX2

Connections of sensor 3:
VCC -> VIN
GND -> GND
SCL -> D22
SDA -> D21
XSHUT -> D18



MazeGUI

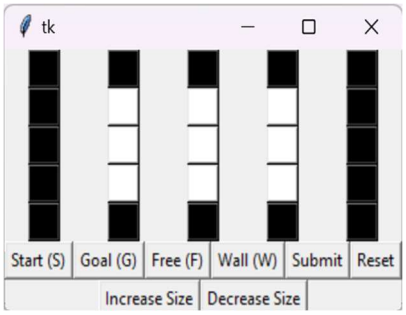
How to run?

Downloaded the MazeGUI directory, open terminal in that directory and run:
python MazeSolver.py

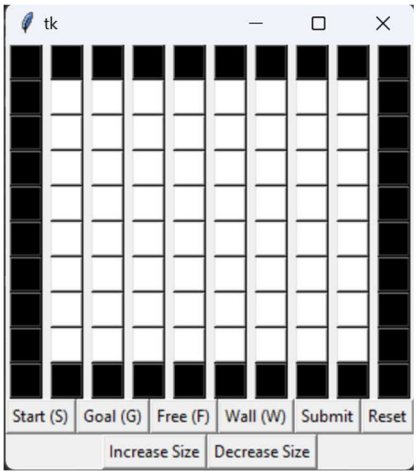
Note:

- You should have Python installed on your computer.
- If a package is missing, you can download it by running: pip install <package_name>
- You should also have a service account in Google Cloud Console:
    - It should have access to a Google sheet named Transmit in your drive.
    - You should also include in the MazeGUI directory a JSON formatted key file for your service account with the name “JSONKey.json.”
    - You can follow a guide for creating a Google service account. ChatGPT can also provide you a step-by-step guide.

Here is how the GUI will look:



3X3



8X8

Buttons:

- **Start, Goal, Free, and Wall:** will help you set the blocks accordingly.
- **Reset:** If you feel that the maze is a mess, or there is a bug with the GUI, you can hit the reset button.
- **Submit:** When you have finished building your maze, you can hit this button to send it to the Google sheet.
- **Increase/Decrease size:** These buttons will help you change the maze size to match the maze you built.

RoboticCarFinal - Directory**:**
After downloading this directory, you may open the MazeRunnerRobot.ino file using Arduino IDE and upload the code to the robot. If you face compilation issues, try using older versions of the libraries required. You may know what libraries are being used when running compilation and getting errors or by simply checking which libraries are imported and downloading them one by one.