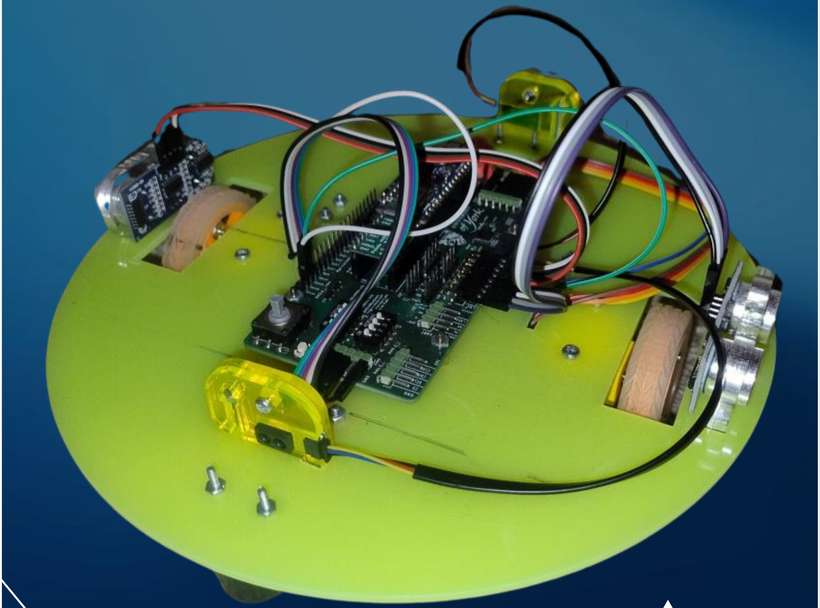


ARB Maze Solving Robot

William Betteridge

Table of contents

01 Design and Construction

02 Peripherals

03 Navigating the Maze Pt.1

04 Navigating the Maze Pt.2

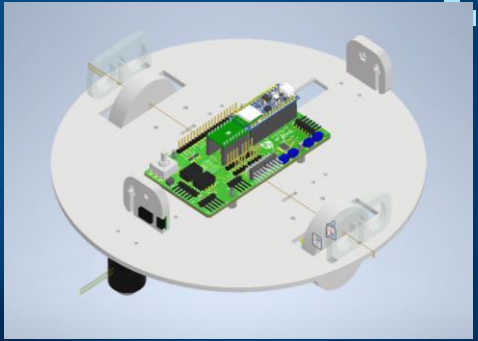
05 Improving the Accuracy

06 Mapping the Maze

01 Design & Construction

Overall Design

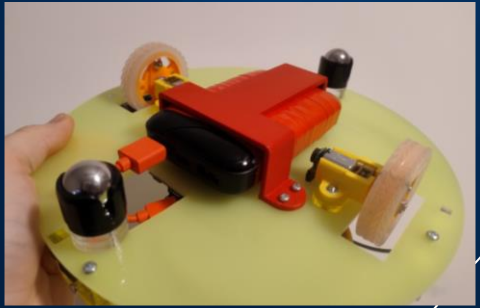
- Weight centrally distributed design.
- Sensors attached in cardinal directions
- 22cm diameter, larger size makes it tricky to navigate small gaps



Battery Holder

- Battery secure and weight centrally distributed
- Efficient use of materials to print holder

- Improve design to place battery in a more suitable location for easier removal



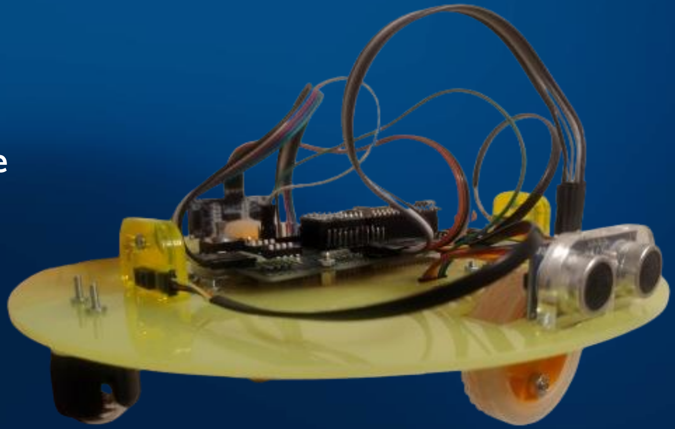
02 Peripherals

Sensors

- Four sensors to cover all cardinal directions
- IR sensors on front offer a narrower more accurate measurement, which can read data faster.
- Ultrasonic Sensors on right offer a wider FOV and longer-range measurement

Motors

- Mounted on the bottom equidistant from centre
- Circular robot design allows to rotate on the spot



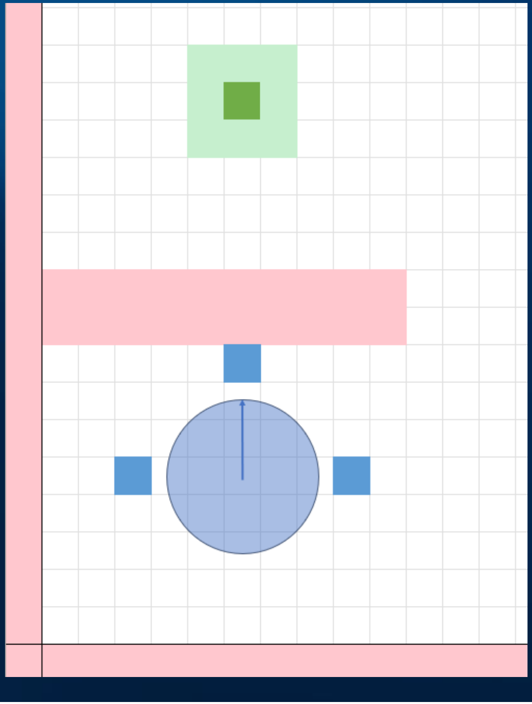
03 Navigating the Maze Part 1

Planning and Creating an Algorithm

- Initially designed a simple wall following algorithm
  - Slow and prone to getting stuck

- Custom algorithm
  - More direct
  - Accounts for various scenarios

- Custom Algorithm Method
  - Calculates the distance from the front, left and right. (blue squares)
  - The shortest distance to the finish (green) is the direction to move
  - Checks sensors to see if it can move
  - Moves, or if cannot move, moves to the next shortest position



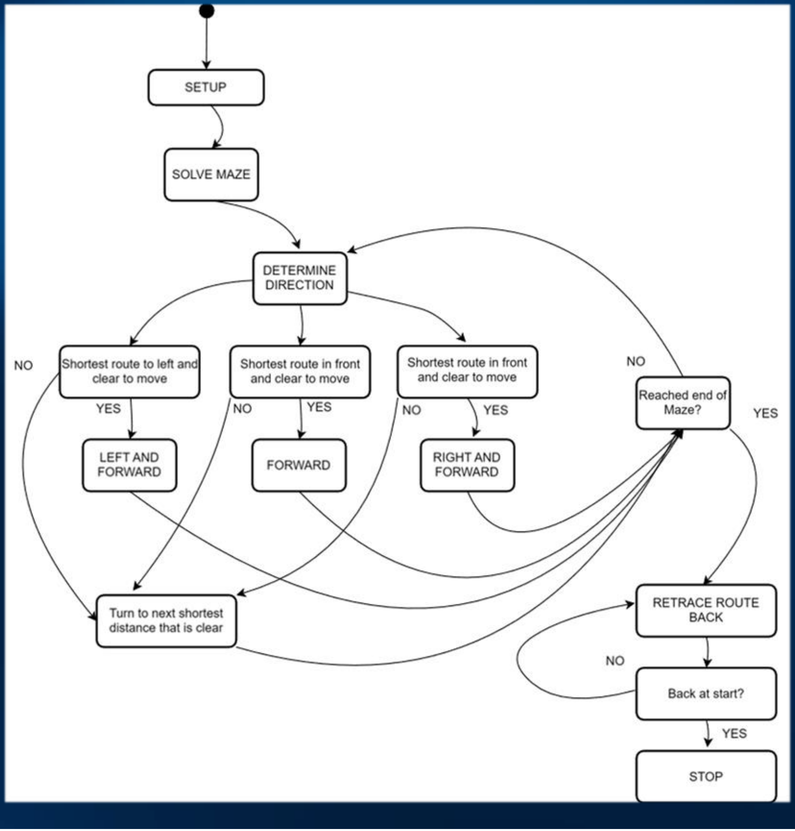
04 Navigating the Maze Part 2

Improving the Algorithm

- Robot moves in 5cm steps
- Calculates new shortest direction each step
- Calculates its current position in map based on encoders
- Records the history of each coordinate the robot stops at

Testing and Success

- Using the serial port to read data whilst running the algorithm
- Checking all values are as expected
- Moving the finish point to various locations to test different corners and routes

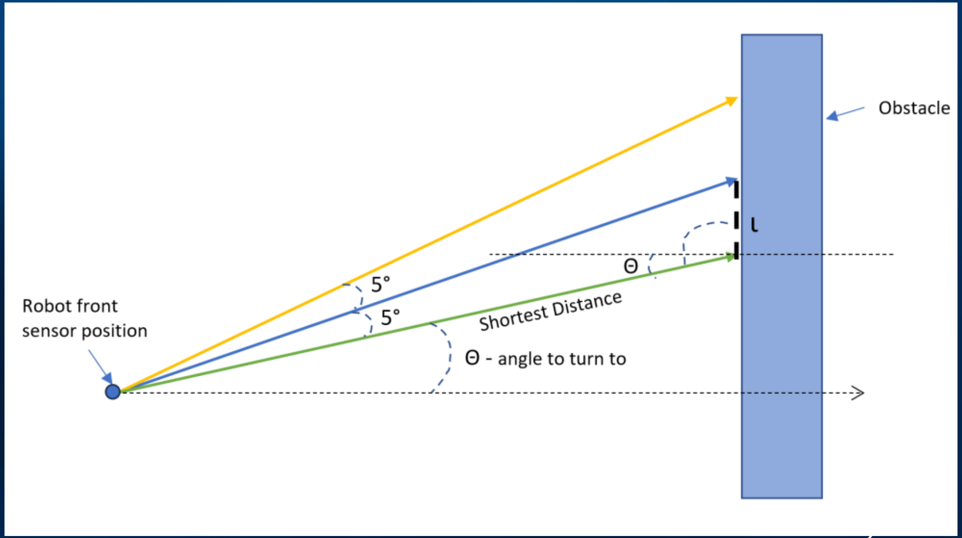


SETUP
SOLVE MAZE
DETERMINE DIRECTION
NO
Shortest route to left and clear to move
Shortest route in front and clear to move
Shortest route in front and clear to move
NO
Reached end of Maze?
YES
YES
NO
YES
NO
YES
LEFT AND FORWARD
FORWARD
RIGHT AND FORWARD
Turn to next shortest distance that is clear
RETRACE ROUTE BACK
NO
Back at start?
YES
STOP

05 Improving the Accuracy

Correcting the Robot's Orientation

- Motors configurations different, and so it was difficult to get the robot to move straight, alignment method created.
- Use of the front sensor to measure the distance at different angles
- Calculate the angle to turn towards the shorter distance
- Orientate the robot perpendicular with the wall in front
- Did not work successfully enough due to spurious sensor readings and hence rotating the robot by an unexpected amount. Therefore, was not implemented in the final design



06 Mapping the Maze

Mapping Methods

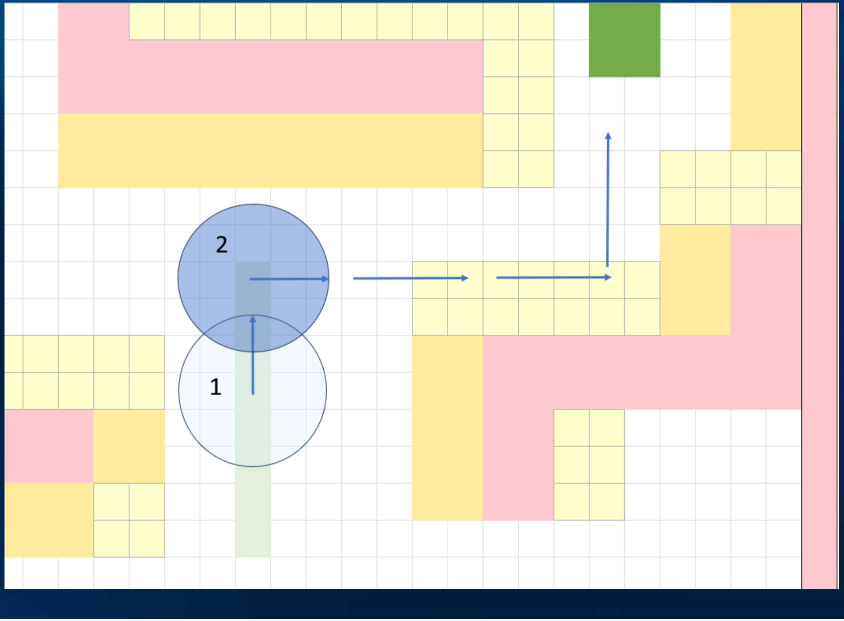
- Each step forwards the robot takes all the sensor distances are read and if the objects are in range they are adding into the map
- Obstacles have a “Border” placed around them to prevent the robot crashing
- “Partial Borders” are placed where the robot hasn’t directly been in front or along side the edge of the obstacle

Handling Obstacles

- If the robot is moving forwards it can travel through a “Partial Border”, but if it’s turning right or left it cannot.
- This prevents narrow gaps from being fully closed off and inaccessible

▲ ▲ ▲

- Free Space = White
- Robot History = Light Green
- Partial Border = Light Yellow
- Border = Yellow
- Obstacle = Red
- Finish = Green



Any Questions?

Thank You!