Computer Science and Programming
Johnson 2024-25

Project Title

Autonomous Maze-Solving Robot

Team Members:

- Saksham

Table of Contents

---

1. Introduction

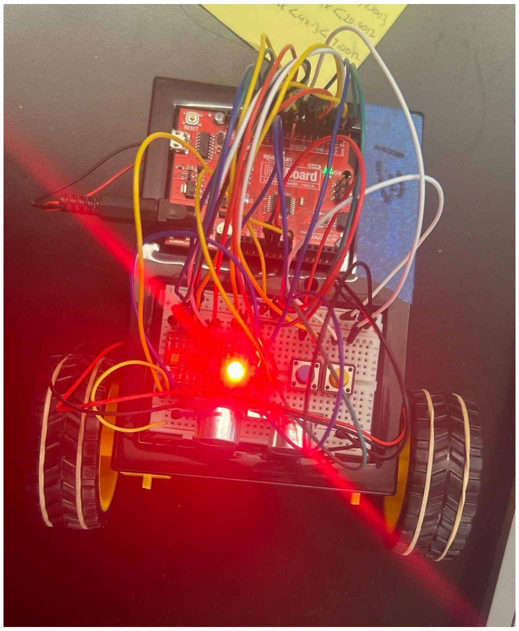
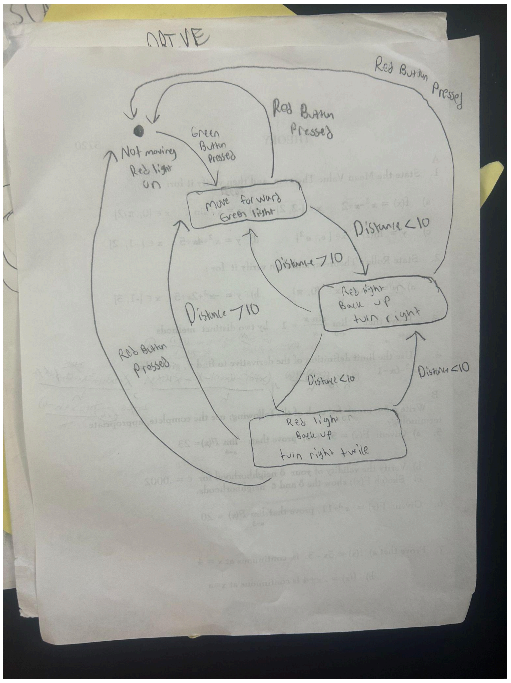
- **Purpose:** Describe the goal of the project.
  - To make a fully autonomous robot that can solve any maze without any human intervention.
- **Scope:** Define boundaries (e.g., robot size, maze size).
  - The robot will use one Arduino Qwiic and 4 AA batteries, the robot can also only use Arduino components (the exception of rubber bands). The maze will have both right and left turns and straight sections of varying length.

2. Objectives & Requirements

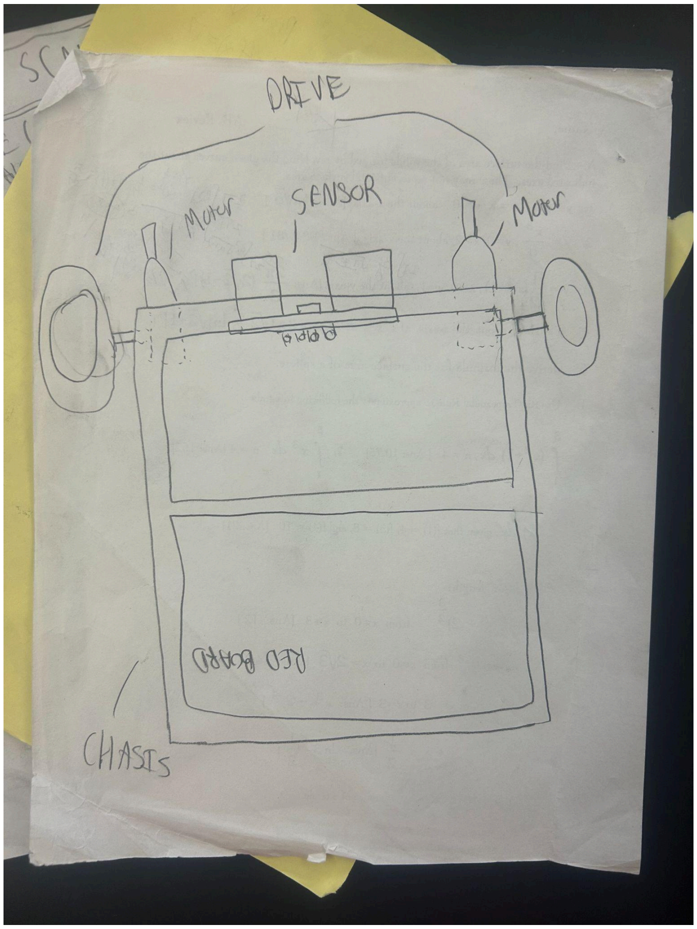
- **Functional Requirements:**
  - The robot needs a green button to start the robot, and a red button to stop the robot. The robot needs a light to turn green letting it be known that the robot is moving, the robot also needs a red light letting it be known that there is an object roughly 3 inches or less away from the robot.

- **Performance Requirements:**
    - The robot must complete the maze in under 1 minute and 30 seconds. The robot must also navigate the maze all on its own, any interventions such as nudging or pushing the robot will cause -5 points. If the robot is unable to complete the maze it is -15 points, if the robot cannot move at all it is -30 points.
- **Constraints:**
    - The robot can only use Arduino components(exception of rubber bands), making sure to only have 4 AA batteries and 1 RedBoard Qwiic. The robot has to be autonomous to earn an A grade. The robot must have a start switch that activates motion once the robot is placed on the track. Once the robot is on the maze any interventions will cause point deductions.

3. System Architecture and Design

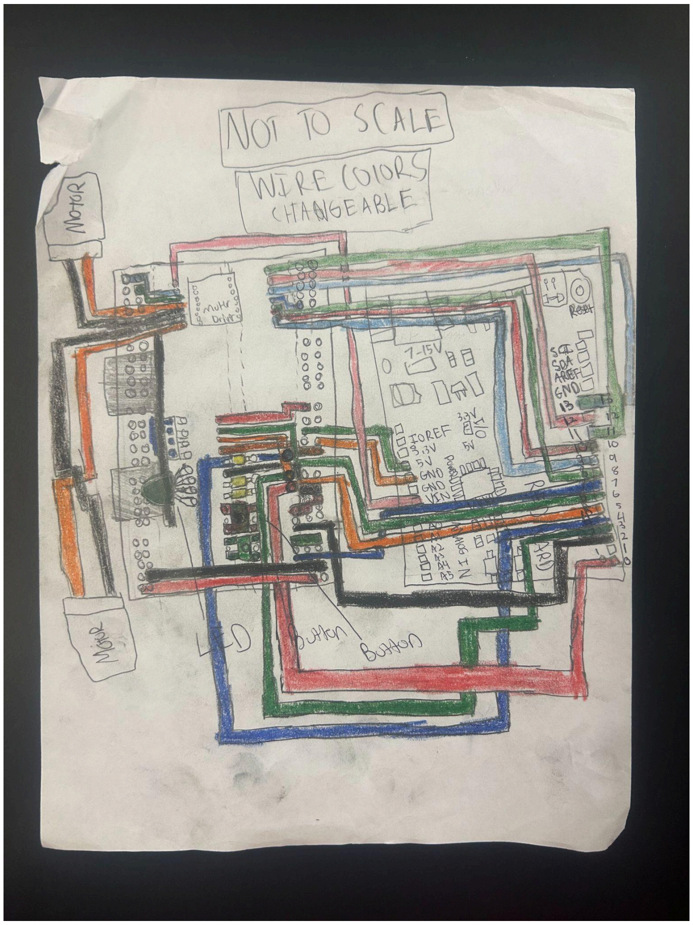
- **Block Diagram:** (Diagram or sketch of system and connections
    -  

- **Component List:**
    - 26 Jumper wires
    - 3 330Ω Resistors
    - 1 4 pin LED
    - 1 Distance Sensor
    - 2 Motors
    - 2 buttons
    - 1 Motor Driver
-

- **Chassis & Drive:** Description and drawings.
    - 

- **Sensor Placement:** distance sensors.
    - The Distance Sensor is placed with this orientation on these specific pins
        - A14 - VCC
        - A15 - Trig
        - A16 - Echo
        - A17 - GND
-

- **Electrical Wiring:** Diagram and pin assignments.



    - Jumper Wire: A1 - VIN
    - Jumper Wire: (A2 - 5V Rail +)
    - Jumper Wire: (A3 - GND Rail -)
    - Jumper Wire: (J1 - Digital Pin 11)
    - Jumper Wire (J2 - Digital Pin 12)
    - Jumper Wire (J3 - Digital Pin 13)
    - Jumper Wire (J4 - 5V Rail +)
    - Jumper Wire (J5 - Digital Pin 8)
    - Jumper Wire (J6 - Digital Pin 9)
    - Jumper Wire (J7 - Digital Pin 10)
    - Jumper Wire (E14 - 5V Rail +)

- Jumper Wire (E15 - Digital Pin 6)
- Jumper Wire (E16 - Digital Pin 5)
- Jumper Wire (E17 - GND Rail -)
- Jumper Wire (J18 - Digital Pin 4)
- Jumper Wire (E19 - GND Rail -)
- Jumper Wire (J20 - Digital Pin 3)
- Jumper Wire (J21 - Digital Pin 0)
- Jumper Wire (J22 - GND Rail -)
- Jumper Wire (J24 - Digital Pin 2)
- Jumper Wire (J25 - GND Rail -)
- Jumper Wire (J27 - Digital Pin 7)
- Jumper Wire (5V Rail + - 5V (Red Board))
- Jumper Wire (GND Rail - -GND (Red Board))
- Jumper Wire (5V Rail + - 5V Rail +)
- Jumper Wire (5V Rail + - 5V Rail +)
- Motor Driver (C1 - C8) & (G1 - G8)
- Right Motor (Red - A4) & (Black - A5)
- Left Motor (Black - A6) & (Red - A7)
- Distance Sensor (A14(VCC), A15(Trig), A16(Echo), A17(GND))
- RGB LED (A18(Blue), A19(GND), A20(Green), A21(Red))
- 330Ω Resistor (E18 - F18)
- 330Ω Resistor (E20 - F20)
- 330Ω Resistor (E21 - F21)
- Red Button (D22 - D24) & (G22 - G24)
- Green Button (D25 - D27) & (G25 - G27)

4. Maze-Solving Algorithm

- **Algorithm Description:** (General process used to solve maze)

go forward (Green light), if you see a wall go right, else keep going forward

go right (Red light), if you see a wall go right twice, else keep going forward

go right twice(Red light), if you see a wall go right once, else keep going forward

- **Flowchart:** (Step-by-step logic)
    - ○
- **Pseudocode:** (Document key routines or functions)

```
if (start_button_pressed)
{       if (distance < 3)
        {
                Red_light();
                turn_right();
                if (distance < 3)
                {
                        Red_light();
                        turn_right_twice()
                }
        }
        else
        {
                Green_light();
                move_forward();
        }
}
else
{
        Stop moving();
        Red_light();
}
```

5. Software Design

- **Module Breakdown:** Describe segments or functions that control the robot (Sensor reading, motor control, decision logic)
    - Sensor Reading - Controls the robot by detecting how far away the object is. This is distance is used in the code in order to determine what the robot should do
    - Motors - used to make the car move and turn
    - Buttons - Used to start or stop the robot by detecting whether or not the corresponding button is pressed and telling the code that. Then the code tells the robot to start or stop.
    - LED - Used to change the color of the light based on inputs given by the sensor and button. If there is an object within 3 inches the light turns red, if there is not an object within 3 inches and the green button is pressed then the light turns green, if the red button is pressed then the light turns red.
    - Motor Control - Controls the Motors and gives commands to the motors telling them how long to spin, what direction to spin, what speed to spin.

6. Testing & Validation

- **Test Plan:** (Specific action steps to ensure robots success)
    - Make sure the buttons work by pressing the buttons, press the on button first and check that the motors start moving and the light turns green, then press the off button and see if the car turns off and the light turns right
    - Check the LEDS, run the code and see if the light starts off red, press the button and see if the light turns green, put your hand in front of the sensor to see if the light turns red
    - Check the motors by placing the car in the maze and make sure it makes all the necessary turns needed.
- **Success Criteria:** Criteria for passing tests.
    - The on button turns on the car, the off button turns off the car. When the car is off the LED is red and the motors are not running. When the car is on the LED is

green and the motors are moving forward. If there is an object in front of the car it begins turning away from the object and the LED is red.

- **Results Log:** Record test runs and outcomes.
  - Test 1: LED does not work, Button Does not work, the car does work however and it can move in the maze, does hit walls and not turn effectively.
  - Test 2: LED works, buttons dont work, the car does the same as Test 1
  - Test 3: LED does not work, Buttons dont work, car does not move.
  - Test 4: LED works, Buttons work, car does not move
  - Test 4: LED works, Buttons work, car is unable to turn.
  - Test 5: LED works, Buttons work, car can move and turn. When placed in maze it starts going through the maze but gets stuck 25% of the way through
  - Test 6: LED Works, Buttons work, car can move and turn. When placed in the maze it starts going through the maze and gets stuck 10% of the way through
  - Test 7: LED Works, Buttons work, car can move and turn. When placed in the maze it starts going through the maze and is unable to make the 180 degree turn.

7. Timeline & Milestones

Date	**Milestone**
5/9/25	Research & design document
5/16/25	Basic navigation prototype
5/23/25	Full maze tests & tuning

8. Daily Log & Task Tracker

Daily Log (one entry per class period):

Date	**Tasks Planned (ID)**	**Tasks Completed (ID)**	**Issues/Blockers**	**Learnings & Notes**	**Next Steps**
5/5/25	1	1	the wires were bad	Need to make sure the wires don't get too messy	Task 2

Date	Tasks Planned (ID)	Tasks Completed (ID)	Issues/Blockers	Learnings & Notes	Next Steps
5/6/25	2	2	The wires got in the way of the motor	Make sure that the worse don't get in the way of the sensor or motor	Task 3
5/7/25	3	none	The button code is not working	Sometimes the button works but then the light doesn't	Task 3
5/8/25	3	None	the code isn't compiling	bugs and syntax need to be fixed effectively	task 3
5/9/25	3 & 4	Task 4	The button doesnt work	Sometimes the button works but then the light doesn't	task 3
5/12/25	3	None	The button doesnt work	Sometimes the button works but then the robot doesn't	task 3
5/13/25	3	None	The stop button doesn't work	Everything works but the stop button	task 3
5/14/25	3	None	The sensor doesn't work	Always showing 0 when serial.println	task 3
5/15/25	3	None	The sensor doesn't work	Always showing 0 when serial.println	task 3
5/16/25	3	None	The other button and	Now sensor is working but buttons aren't,	task 3

Date	Tasks Planned (ID)	Tasks Completed (ID)	Issues/Blockers	Learnings & Notes	Next Steps
			other light doesn't work	need to modify code	
5/19/25	3	None	Second Button, both lights dont work	Very glitchy, does not work 100%	task 3
5/20/25	3 and 4	3	wiring is bad now	Car moves very slowly and needs to be fixed up.	task 4
5/21/25	Task 4	4	None	Now need to make sure robot can operate in maze	task 5
5/22/25	Task 5	None	It keeps getting stuck at certain points	Make sure to modify turntime in code, also switch from state machine to lots of if then statements. (actually more efficient code somehow)	Task 5
5/23/25	Task 5	5	Presentation day	Presentation day	Nothing

Task Tracker:

Task ID	Description	Assigned To	Priority	Status	Due Date
1	make sure the circuit works	Saksham	High	complete	5/6/2025
2	apply code and make sure the circuit works	Saksham	high	Not Started	5/7/2025

Task ID	Description	Assigned To	Priority	Status	Due Date
3	make buttons and light bulbs on code	Saksham	high	Not started	5/9/2025
4	place buttons and light bulbs on circuit	Saksham	High	Not started	5/9/2025
5	Test maze	Saksham	High	Not started	5/23/2025
6					
7					
…					