EEE3099S

MILESTONE 1

ONRAMP COURSES & PAPER DESIGN

Friso Vijverberg – VJVFRI001

Liam Breytenbach – BRYLIA002

Julian Dower – DWRJUL001

Table of Contributions

Contributor	Project Subsection	Details
Liam	**Requirement Analysis**	• Functional Requirements • Constraints • Possible bottlenecks • Project Manager Page
Friso	**Subsystem Design**	• Subsystem and Sub-subsystems Requirements • Subsystem and Sub-subsystems Specifications • Inter-subsystems Interactions • UML Diagrams • Development Timeline
Julian	**Acceptance Test Procedure**	• Figures of merits based on which you would validate your final design. • Experiment design to test these figures of merit. • Acceptable performance definition

Introduction

The advancement of autonomous systems has led to remarkable innovations that redefine the boundaries of what machines can achieve. This report goes into how an autonomous mobile robot is designed to navigate through an unknown maze while using sophisticated techniques such as line tracking, object detection, distance calculation and path optimization. By combining technologies such as line sensors, ultrasonic sensors and Arduino chips, this report demonstrates the connection between artificial intelligence, electronics, robotics, and sensors.

The primary objective of this project is to demonstrate how a robot can intelligently manoeuvre through a maze, accurately detecting objects along its path, calculating their distances, and ultimately identifying the nearest object once all the objects have been located. This project will show the remarkable potential of robotic systems in accomplishing complex challenges and will teach how to integrate the skills gathered over the previous semesters of the curriculum in this design problem relevant to Mechatronics.

Requirements Analysis

1. **Functional Requirements**
   - The mobile robot must be autonomous.
   - The mobile robot must be able to navigate through a maze by tracking a line.
   - Must be able to identify objects within the maze.
   - The robot must stop at a 'detect line' and measure the distance to the object.
   - The robot must acknowledge when all the objects in the maze have been found.
   - After finding all the objects within the maze, the robot must calculate the closest object to its location and navigate to that object.

2. **Constraints**
   a) Sensor constraints
      - Maximum of 5 line sensors
   b) Power system constraints
      - 2 x 18650s must be used to power the module
   c) Physical constraints
      - Differential drive robot must be an Arduino Turtle: 2WD Mobile Robot Platform
      - Axle length: 13.6 cm
      - Wheel diameter: 6.2 cm
   d) Microcontroller constraints
      - Arduino Nano 33 IoT to be used
      - Arduino IDE may not be used

3. **Bottlenecks**

Bottlenecks are a point of congestion that can slow or stop progress in a project. Possible bottlenecks could include the following:

- Waiting for a team member to complete their subsection necessary for the next step of the project.
- Miscommunication between group members, leading to a subsection being incomplete.
- Missing components that are essential for the next step in the testing procedure.
- Work done incorrectly, causing more time to be spent correcting the error.

Subsystem Design

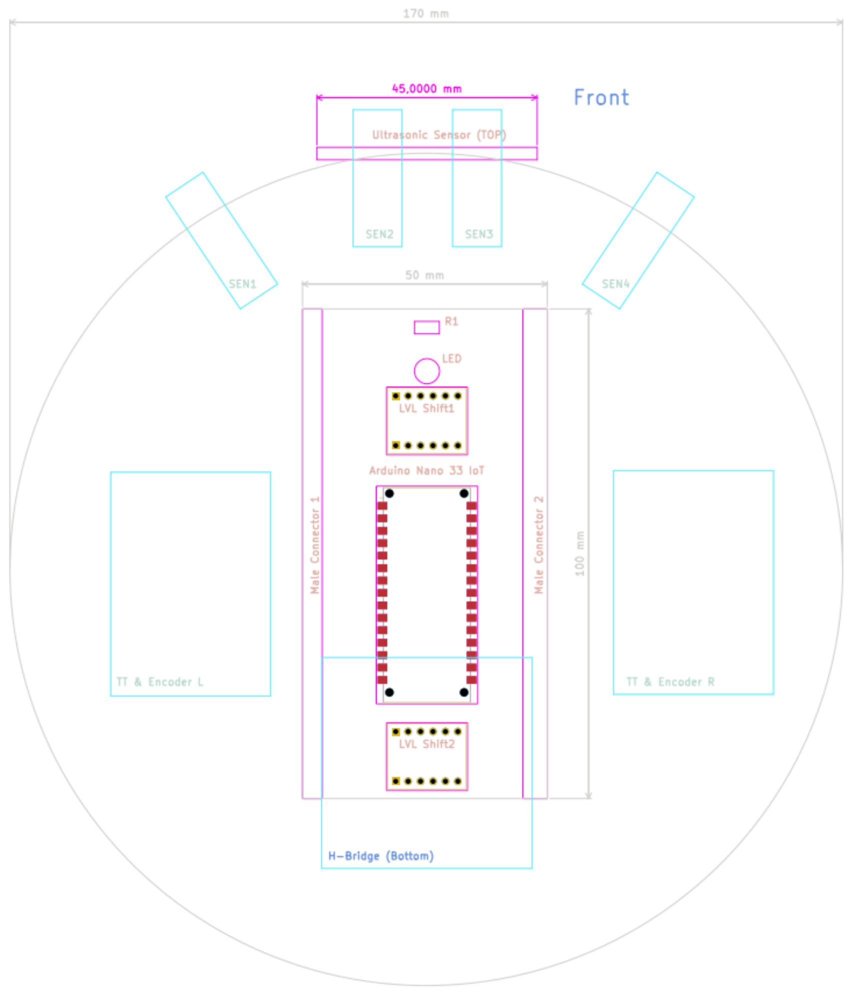

Fig. 1. Top view design layout. Blue outlines are on the bottom level, pink outlines on the top level.

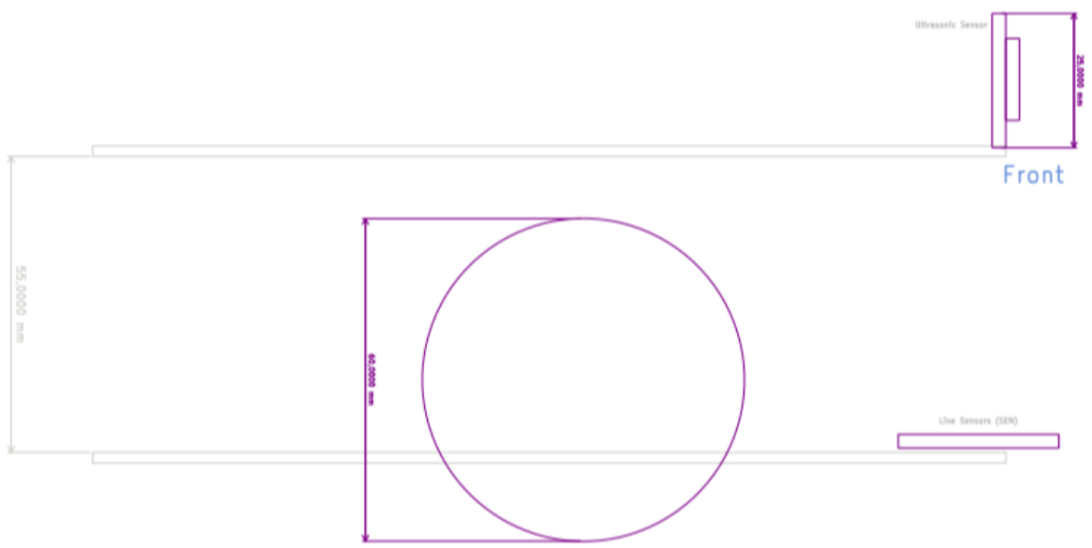

Fig. 2. Right-side view layout to indicate orientation of ultrasonic sensor

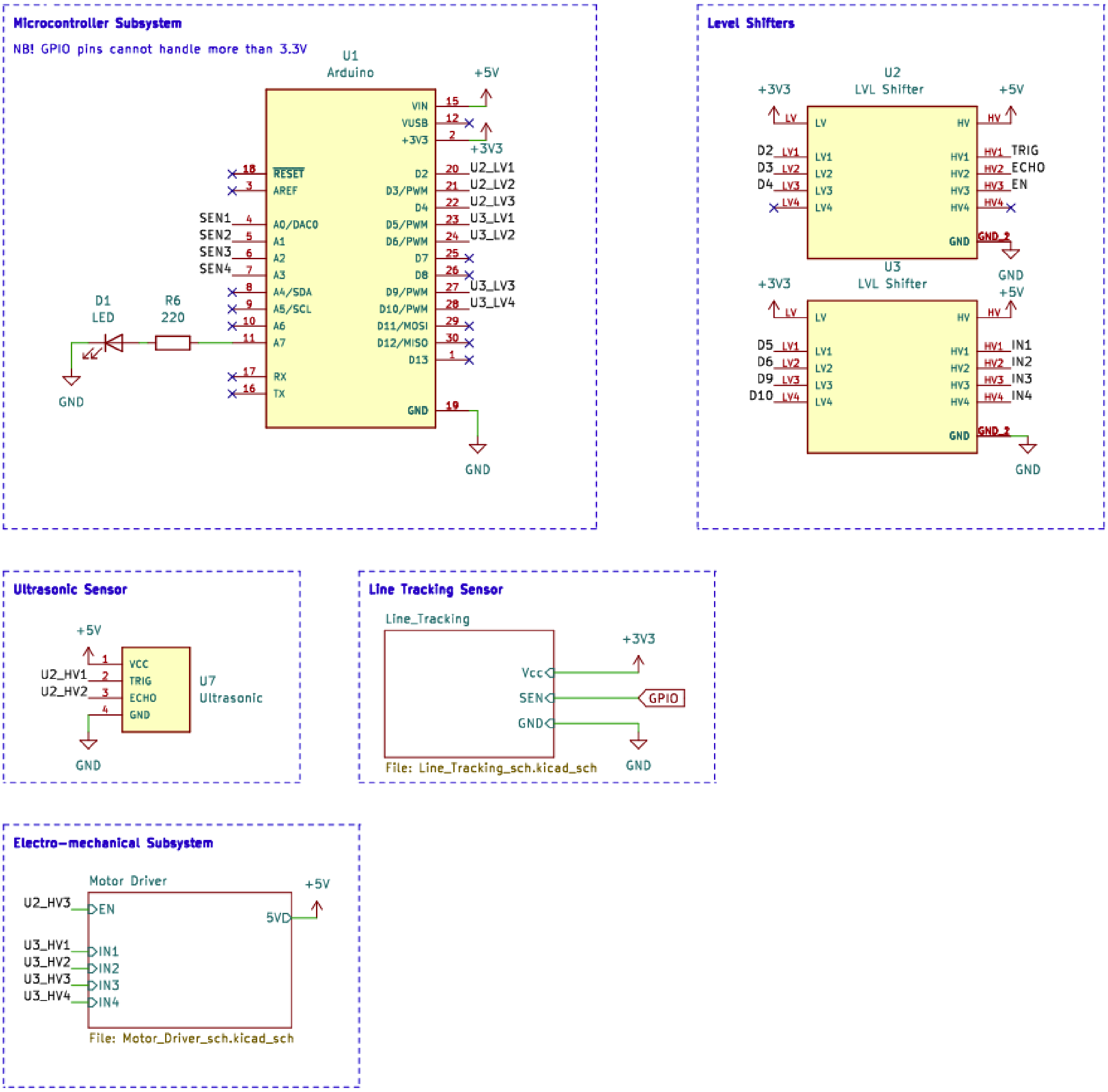

Fig. 3. Parent Schematic showing connections with Arduino, sensors, level shifters and electro-mechanical subsystem

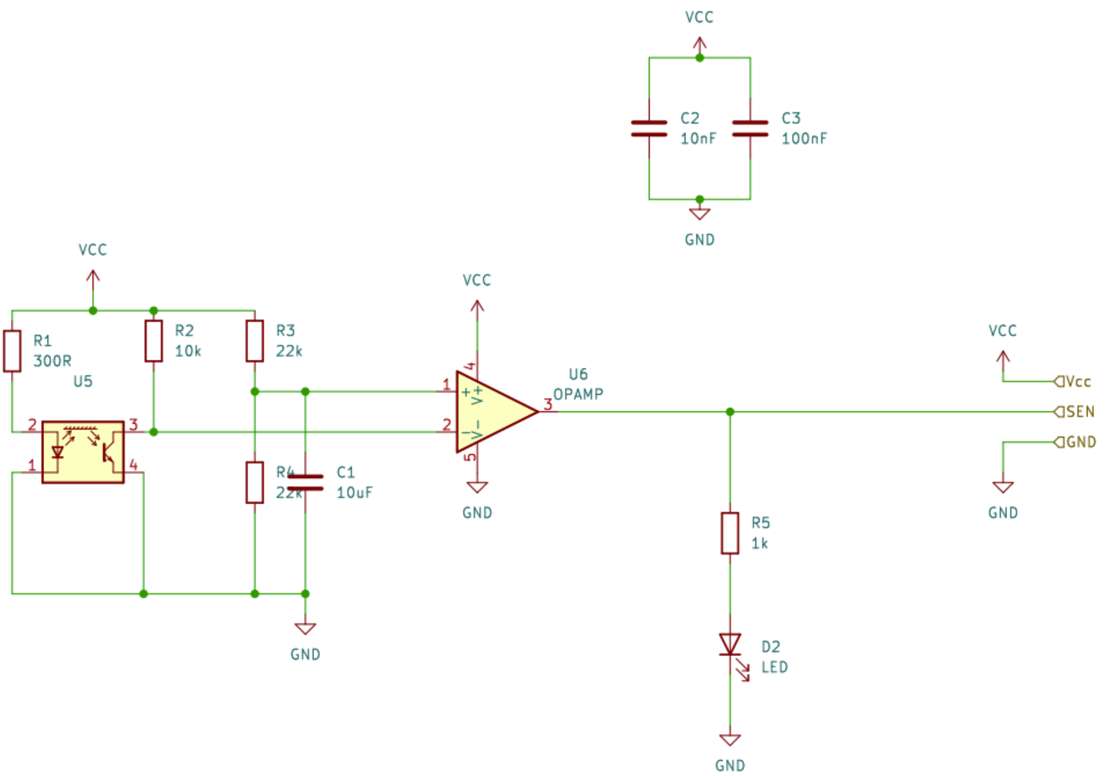

Fig. 4. Schematic of the line sensor demonstrating connections and operation [1].

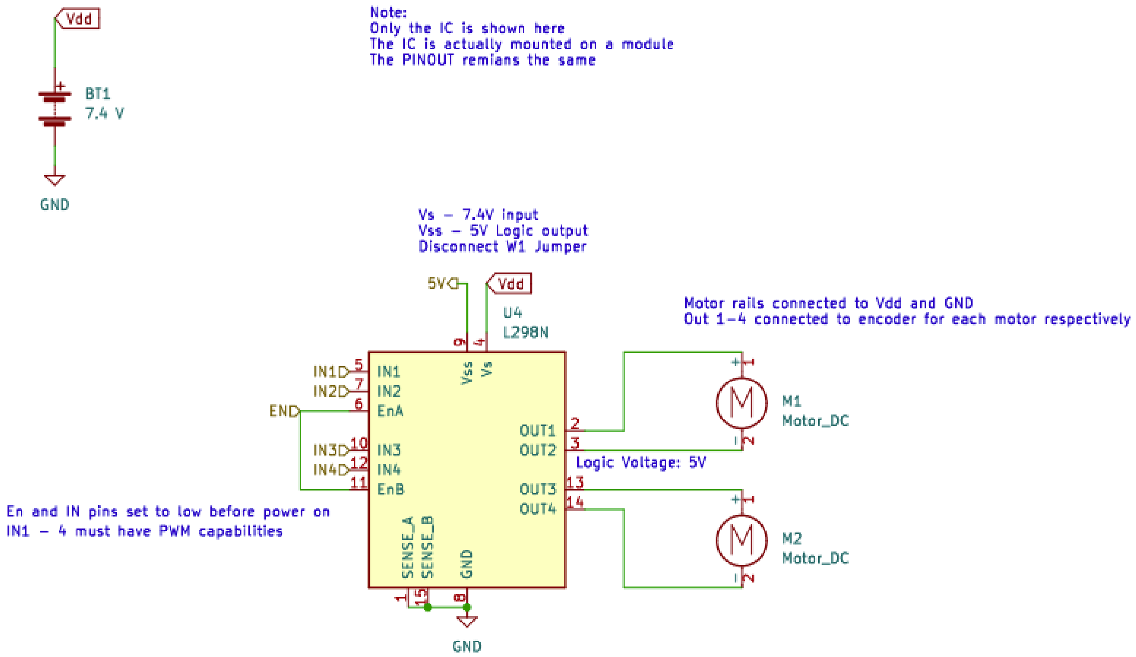

Fig. 5. Schematic of electro-mechanical subsystem. Note: the full H-bridge module and the encoders are not included. The pinout remains unaltered.

The requirements and specifications for each sub system and sub-subsystem are listed below. Please refer to Fig. 1 and 2 for any physical requirements/specifications in addition to the placement of the ultra-sonic sensor. See Fig. 3-5 for a graphical demonstration of connections and electrical requirements/specifications.

Note: no additional voltage regulator has been added to the system. There is an onboard regulator on the microcontroller and H-bridge module which can provide sufficient driving current (see requirements/specifications below).

1. Electronic Subsystem

Requirements

- The entire electronics must occupy a Veroboard of at most 10 cm x 5 cm.

Power

Requirements

- Electronics should be operated at most at a 5V logic level.
- Motor supply must be between 7V and 7V5.
- 3V3 and 5V voltage regulator with sufficient output current:
    - Ultrasonic Sensor: 15mA
    - H-bridge: 30µA per pin

Specifications

- Battery voltage: 7V4
- 5V voltage regulator (78M05) on H-bridge with a maximum of 500mA driving current [5]
- 3V3 internal voltage regulator on Arduino with a maximum of 150mA driving current [6]
- GPIO input/output voltage: 3V3
- Level shifter:

  HV: 5V LV: 3V3

Arduino

Requirements

- Red LED placed on the top of robot to indicate ‘detect’ state.
- Computational capacity to handle a maze solving algorithm while operating peripherals.
- 1 MB Flash memory capacity for driver algorithm.
- Approximately 100 kB RAM to store distance to object.
- PWM on 4 GPIO pins to allow for speed control.
- <10mA driving current on analogue GPIO pins

Specifications

- Microcontroller: SAMD21 Cortex®-M0+ 32bit low power ARM MCU
- Memory: 256kB SRAM, 1MB Flash
- 5 GPIO pins with PWM capabilities [6]
- Maximum 40mA output current on GPIO pins (can provide less).

Sensor

Requirements

- Array of line sensors must map the position of the line relative to the robot
- Line sensors to be placed facing downwards
- Ultrasonic sensor must measure distance to objects placed in front of the robot
- Sensors must be able to operate off a power supply ≤ 5 V

Specifications

- Line sensor detection range: 1 – 2cm
- Line sensor power supply: 3V3
- Line sensor operating current: < 10mA [2]

- Ultrasonic sensor detection range: 2 – 400cm
- Ultrasonic sensor measuring angle: 15°
- Ultrasonic sensor power supply: 5V
- Ultrasonic sensor working current: 15mA [3]

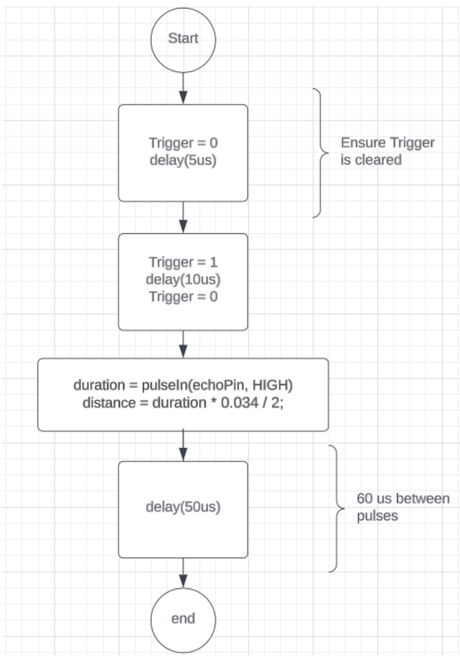

Fig. 5 Flowchart for Ultrasonic Sensor [4]

2. **Electro-mechanical Subsystem**

Requirements

- 7V < Motor Supply < 7V5
- Encoder Logic voltage = 5V

Specifications

- Motor supply = 7.4V
- Driver Current = 2A per motor
- Level shifter:

  HV: 5V [Encoder Input] LV: 3V3[Arduino I/O]

- Gear ratio: 1:120 [7]

Inter – Subsystem Interactions

Please refer to Fig. 7 under the UML Diagram section for a graphical demonstration of the interfacing.

Arduino Module

- Electro-mechanical subsystem
  - Arduino input voltage: 5V obtained from Vss pin on H-Bridge
  - H-bridge input pins: 3V3 PWM GPIO pins level shifted to 5V
- Sensor sub-subsystem
  - Ultrasonic input pin [Trigger]: 3V3 digital GPIO pin shifted to 5V
  - Ultrasonic output pin [Echo]: 5V level shifted to 3V3 and connected to digital GPIO pin
  - Line Sensor output pin: 4 GPIO pins at 3V3 logic level

Electro-mechanical subsystem

- Arduino sub-subsystem
  - Arduino input voltage: 5V obtained from Vss pin on H-Bridge
  - H-bridge input pins: 3V3 PWM GPIO pins level shifted to 5V
- Electro-mechanical subsystem
  - Ultrasonic sensor source voltage: 5V obtained from Vss pin on H-bridge
- Power sub-subsystem
  - 7V4 obtained from battery connected to Vs on H-bridge [Note: Jumper W1 must be disconnected]
  - Motor Rails: 0 – 7V4

Sensor sub-subsystem

- Arduino sub-system
    - Ultrasonic input pin [Trigger]: 3V3 digital GPIO pin level shifted to 5V
    - Ultrasonic output pin [Echo]: 5V level shifted to 3V3 and connected to digital GPIO pin
    - Line sensor output pin: 4 GPIO pins at 3V3 logic level
    - Line sensor source voltage: 3V3 obtained from Arduino
- Electro-mechanical subsystem
    - Ultrasonic sensor source voltage: 5V obtained from Vss pin on H-bridge

Power sub-subsystem

- Electro-mechanical subsystem
    - 7V4 obtained from battery connected to Vs on H-bridge [Note: Jumper W1 must be disconnected]
    - Motor Rails: 0 – 7V4

UML Diagrams

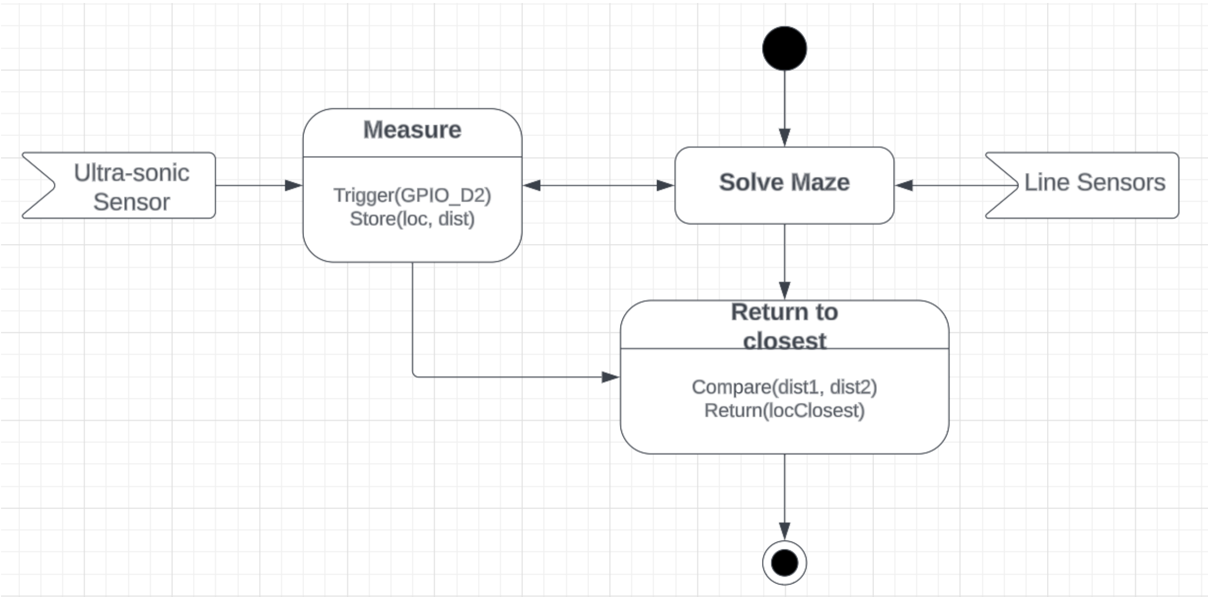

Fig. 6. UML State Diagram for autonomous algorithm

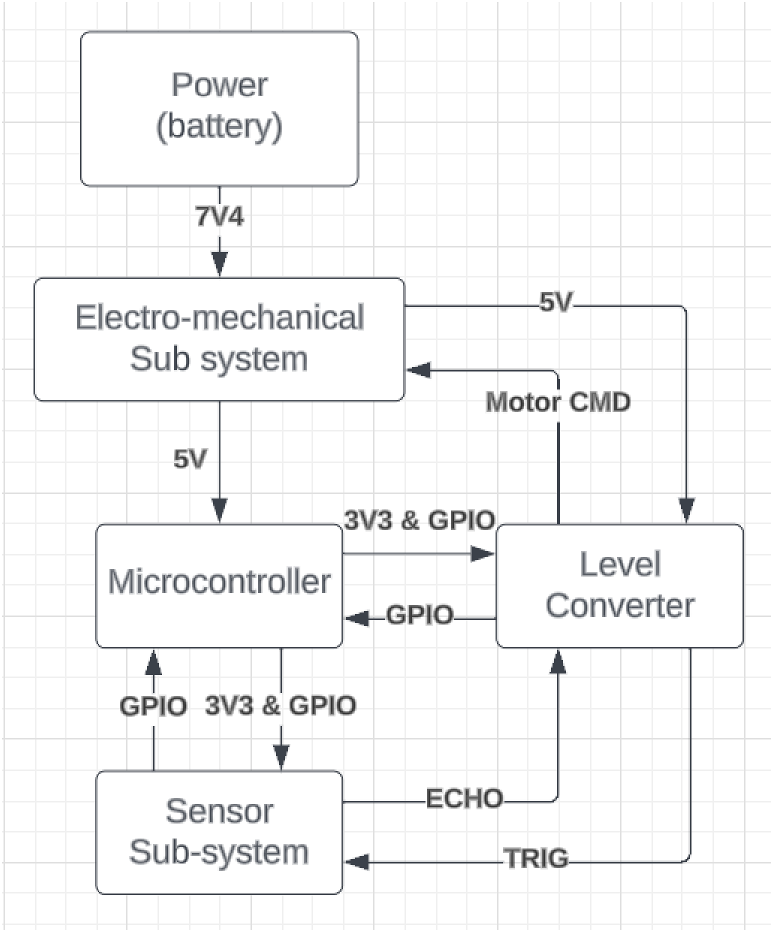

Fig. 7. High order UML system diagram demonstrating interfacing.

Acceptance Test Procedures

Figures of merit are defined as the criteria used to evaluate the robot's performance and quality. The figures of merit for the line-following robot are:

- Speed: how fast it can follow the line without losing track or deviating from it
- Accuracy: how closely it can follow the line without overshooting or undershooting the turns or curves.
- Robustness: how well it can cope with different lighting conditions, surface textures, line widths, and colours.
- Efficiency: how much power it consumes and how long it can operate on a single charge.
- Sensory Output: the accuracy of the output values of the ultrasonic and line sensors.

Experiment design is the method that we use to test the figures of merit and collect data. Some experiment designs for a line following robot are:

- Speed: by measuring the time that it takes to complete a fixed distance or a fixed number of laps on a given track. It also compares its speed with other robots or a reference speed.
- Accuracy: by measuring its deviation from the line using sensors, cameras, or markers. Can also calculate the error rate or percentage of deviation from the line over a given distance or time.
- Robustness: By testing on different tracks with varying lighting conditions, surface textures, line widths, and colours. One can also introduces obstacles, gaps, or noise in the line and sees how it reacts and recovers.
- Efficiency: by measuring its power consumption using a multi-meter or a power meter. It also measures its battery life or run time on a given track or task.
- Sensory Output: by executing test programs once connected the physical circuit has successfully been built. The code snippets for the line and ultrasonic sensor test program are given below [2], [3]:

```
// Ultrasonic Sensor
GPIO.output(TRIG, True)
time.sleep(0.00001)
GPIO.output(TRIG, False)

while GPIO.input(ECHO)==0:
      pulse_start = time.time()
while GPIO.input(ECHO)==1:
      pulse_end = time.time()

// Line Sensor
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  Serial.println(digitalRead(2)); // print the data from the sensor
  delay(500);
```

Acceptable performance is the standard that we use to judge whether the robot meets its expectations and requirements. Some possible acceptable performance definitions for a line following robot are:

- Speed: it should be able to follow the line at a minimum speed of *X* m/s without losing track or deviating from it.

- Accuracy: it should be able to follow the line with a maximum deviation of *Y* cm or mm or a maximum error rate of *Z*% over a given distance or time.

- Robustness: it should be able to follow any line with any width, colour, texture, and lighting condition without losing track or deviating from it. It should also be able to handle obstacles, gaps, or noise in the line without crashing or stopping.

- Efficiency: it should be able to follow the line with a minimum power consumption of *W* watts or joules per meter or second. It should also be able to operate on a single charge or battery for at least *T* hours or minutes.

- Sensory Output: line sensors should all be giving feedback, ultrasonic sensor accurate within 2%

Variable	Description
X	The minimum speed of the robot in m/s
Y	The maximum deviation of the robot from the line in cm
Z	The maximum error rate or percentage of deviation of the robot from the line
W	The minimum power consumption of the robot in watts
T	The minimum battery life or run time of the robot in minutes

Project Manager

Monday.com is used for the Project manager Page:

https://liam210120.monday.com/boards/1251372072

Project

Task	Owner	Due Date	Status	Priority	Notes	Timeline
Weekly Report 1	L F J	~~Aug 4~~	Done	Low	This is the weekly r...	Aug 4
Weekly Report 2	L F J	~~Aug 11~~	Done	Low	This is the weekly r...	Aug 11
Milestone 1 (4)	L F J	~~Aug 18~~	Done	High	Report	Aug 18

Subitem	Owner	Status	Due Date	Priority
Complete MATLAB courses	L F J	Done	~~Aug 18~~	Critical ⚠
Requirement Analysis	L	Done	~~Aug 18~~	Medium
Subsystem Design	F	Done	~~Aug 18~~	Medium
Acceptance Test Procedure	J	Done	~~Aug 18~~	Medium
+ Add subitem				

Task	Owner	Due Date	Status	Priority	Notes	Timeline
Weekly Report 3	L F J	Aug 18	Working on it	Low	This is the weekly r...	Aug 11 - 18
Weekly Report 4	L F J	Aug 25	Not Started	Low	This is the weekly r...	Aug 18 - 25
Weekly Report 5	L F J	Sep 1	Not Started	Low	This is the weekly r...	Aug 25 - Sep 1
Weekly Report 6	L F J	Sep 8	Not Started	Low	This is the weekly r...	Sep 1 - 8
Milestone 2 (5)	L F J	Sep 14	Not Started	High	Report	Aug 18 - Sep 14

Subitem	Owner	Status	Due Date	Priority
Distance and Angle Control Al...	L		Sep 14	Medium
Line following algorithm	F		Sep 14	Medium
Object Detection Algorithm	J		Sep 14	Medium
Localisation Algorithm	L		Sep 14	Medium
Simulation	L F J		Sep 14	High
+ Add subitem				

Task	Owner	Due Date	Status	Priority	Notes	Timeline
Weekly Report 7	L F J	Sep 15	Not Started	Low	This is the weekly r...	Sep 8 - 15
Milestone 3	L F J	Sep 14	Not Started	High	Report	Aug 18 - Sep 14
Weekly Report 8	L F J	Sep 22	Not Started	Low	This is the weekly r...	Sep 15 - 22
Weekly Report 9	L F J	Sep 29	Not Started	Low	This is the weekly r...	Sep 22 - 29
Weekly Report 10	L F J	Oct 6	Not Started	Low	This is the weekly r...	Sep 29 - Oct 6
Weekly Report 11	L F J	Oct 13	Not Started	Low	This is the weekly r...	Oct 6 - 13
Weekly Report 12	L F J	Oct 20	Not Started	Low	This is the weekly r...	Oct 13 - 20
Final Report		Oct 20	Not Started	Critical ⚠	Report	Oct 20
+ Add task						
		Aug 4 - Oct 20				Aug 4 - Oct 20

Git Repo

Please follow the link to open the git repository for the project:
https://github.com/FVijverberg/EEE3099S

Onramp Course Certificates

Friso Vijverberg

1. MATLAB Onramp - https://matlabacademy.mathworks.com/progress/share/certificate.html?id=320b2a5f-05b2-4f84-b3e9-b32a317f5357
2. Simulink Onramp - https://matlabacademy.mathworks.com/progress/share/certificate.html?id=48b4253d-6e03-4723-910f-23285f4a7e24
3. Stateflow Onramp - https://matlabacademy.mathworks.com/progress/share/certificate.html?id=48b4253d-6e03-4723-910f-23285f4a7e24
4. Control Design Onramp - https://matlabacademy.mathworks.com/progress/share/certificate.html?id=74b006ea-35b4-4c92-a2ef-0234ffc20ae4

Liam Breytenbach

1. MATLAB Onramp - https://matlabacademy.mathworks.com/progress/share/certificate.html?id=63d50307-9d44-4468-aa50-6e3a4a81deb7
2. Simulink Onramp - https://matlabacademy.mathworks.com/progress/share/certificate.html?id=ff196d29-af8d-421e-9e97-f70064a002aa
3. Stateflow Onramp - https://matlabacademy.mathworks.com/progress/share/certificate.html?id=fb90a9dd-b84d-4ad2-b38b-ae6e3c1ce82c
4. Control Design Onramp - https://matlabacademy.mathworks.com/progress/share/certificate.html?id=75ce9b19-4294-45c5-aa0c-29e27c0657a1

Julian Dower

1. MATLAB Onramp - https://matlabacademy.mathworks.com/progress/share/certificate.html?id=3514dfb0-360b-46f0-9bcf-64c14885f4d6
2. Simulink Onramp - https://matlabacademy.mathworks.com/progress/share/certificate.html?id=826095f9-05f9-47d1-a244-95fa763ef8e5
3. Stateflow Onramp - https://matlabacademy.mathworks.com/progress/share/certificate.html?id=33020bd9-a434-43f3-8fa8-e80fd72c076d
4. Control Design Onramp - https://matlabacademy.mathworks.com/progress/share/certificate.html?id=0ac49379-1441-4364-a7d6-c3e57678f9c5

References

[1] DF Robots, "SEN0017 | Digi-Key Electronics," *Digi-Key Electronics*, 2023. https://www.digikey.co.za/en/products/detail/dfrobot/SEN0017/6588488 (accessed Aug. 11, 2023).

[2] DF Robots, "Line_Tracking_Sensor_for_Arduino_V4_SKU_SEN0017-DFRobot," *wiki.dfrobot.com*. https://wiki.dfrobot.com/Line_Tracking_Sensor_for_Arduino_V4_SKU_SEN0017 (accessed Aug. 08, 2023).

[3] "HC-SR04 Ultrasonic Range Sensor on the Raspberry Pi," *The Pi Hut*, 2019. https://thepihut.com/blogs/raspberry-pi-tutorials/hc-sr04-ultrasonic-range-sensor-on-the-raspberry-pi (accessed Aug. 09, 2023).

[4] B. de Bakker, "HC-SR04 Ultrasonic Sensor with Arduino Tutorial (5 Examples)," *Makerguides.com*, Jan. 12, 2019. https://www.makerguides.com/hc-sr04-arduino-tutorial/ (accessed Aug. 09, 2023).

[5] "Motor Driver Dual H-Bridge Module L298N," *DIYElectronics*. https://www.diyelectronics.co.za/store/brushed-motor-drivers/34-motor-driver-dual-h-bridge-module-l298n.html (accessed Aug. 11, 2023).

[6] Arduino, "Nano 33 IoT | Arduino Documentation," *docs.arduino.cc*. https://docs.arduino.cc/hardware/nano-33-iot (accessed Aug. 11, 2023).

[7] DF Robots, "Wheel_Encoders_for_DFRobot_3PA_and_4WD_Rovers__SKU_SEN0038_-DFRobot," *wiki.dfrobot.com*. https://wiki.dfrobot.com/Wheel_Encoders_for_DFRobot_3PA_and_4WD_Rovers__SKU_SEN0038_ (accessed Aug. 11, 2023).

Appendix

Matlab Certificates

Friso

Liam

MathWorks | Training Services

Course Completion Certificate

Liam Breytenbach

has successfully completed **100%** of the self-paced training course

Stateflow Onramp

DIRECTOR, TRAINING SERVICES

15 August 2023

MathWorks | Training Services

Course Completion Certificate

Liam Breytenbach

has successfully completed **100%** of the self-paced training course

Control Design Onramp with Simulink

DIRECTOR, TRAINING SERVICES

3 August 2023

Julian

MathWorks | Training Services

Course Completion Certificate

Julian Dower

has successfully completed **100%** of the self-paced training course

MATLAB Onramp

DIRECTOR, TRAINING SERVICES

4 August 2023

MathWorks | Training Services

Course Completion Certificate

Julian Dower

has successfully completed **100%** of the self-paced training course

Simulink Onramp

DIRECTOR, TRAINING SERVICES

11 August 2023

MathWorks | Training Services

Course Completion Certificate

Julian Dower

has successfully completed **9%** of the self-paced training course

Stateflow Onramp

DIRECTOR, TRAINING SERVICES

17 August 2023

MathWorks | Training Services

Course Completion Certificate

Julian Dower

has successfully completed **100%** of the self-paced training course

Control Design Onramp with Simulink

DIRECTOR, TRAINING SERVICES

17 August 2023