A white text on a black background

Description automatically generated



# Version History

| Version | Date | Status | Author | Description | Update |
| --- | --- | --- | --- | --- | --- |
| 0.1 | 11/19/2024 | Draft | Adil | Document creation | Document creation |
| 0.1 | 11/19/2024 | Draft | Dylan, Kenneth | Added class diagram and flowchart | Class diagram, flowchart |
| 0.2 | 11/26/2024 | Draft | Adil, Dylan | Updated the class diagram and flowchart. Added descriptions to every design. Formatted the document. | Class diagram, flowchart, document |

# Table of Contents

[Version History 2](#_heading=h.gjdgxs)

[Introduction 5](#_heading=h.30j0zll)

[Design Phase 1 5](#_heading=h.1fob9te)

[Design Phase 2 8](#_heading=h.3znysh7)

[Implementation Phase 1 8](#_heading=h.2et92p0)

[Implementation Phase 2 8](#_heading=h.tyjcwt)

[Testing Phase 1 8](#_heading=h.3dy6vkm)

[Testing Phase 2 8](#_heading=h.1t3h5sf)

[Reflection 8](#_heading=h.4d34og8)

[Bibliography 8](#_heading=h.2s8eyo1)

# Introduction

## Design Phase 1

## Flowcharts

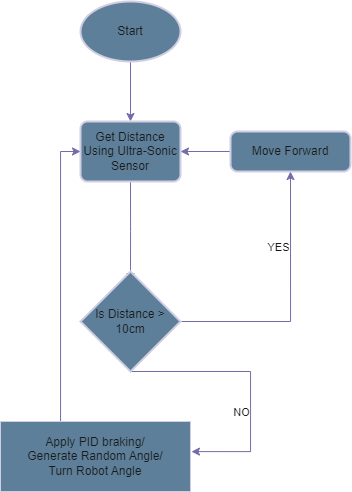
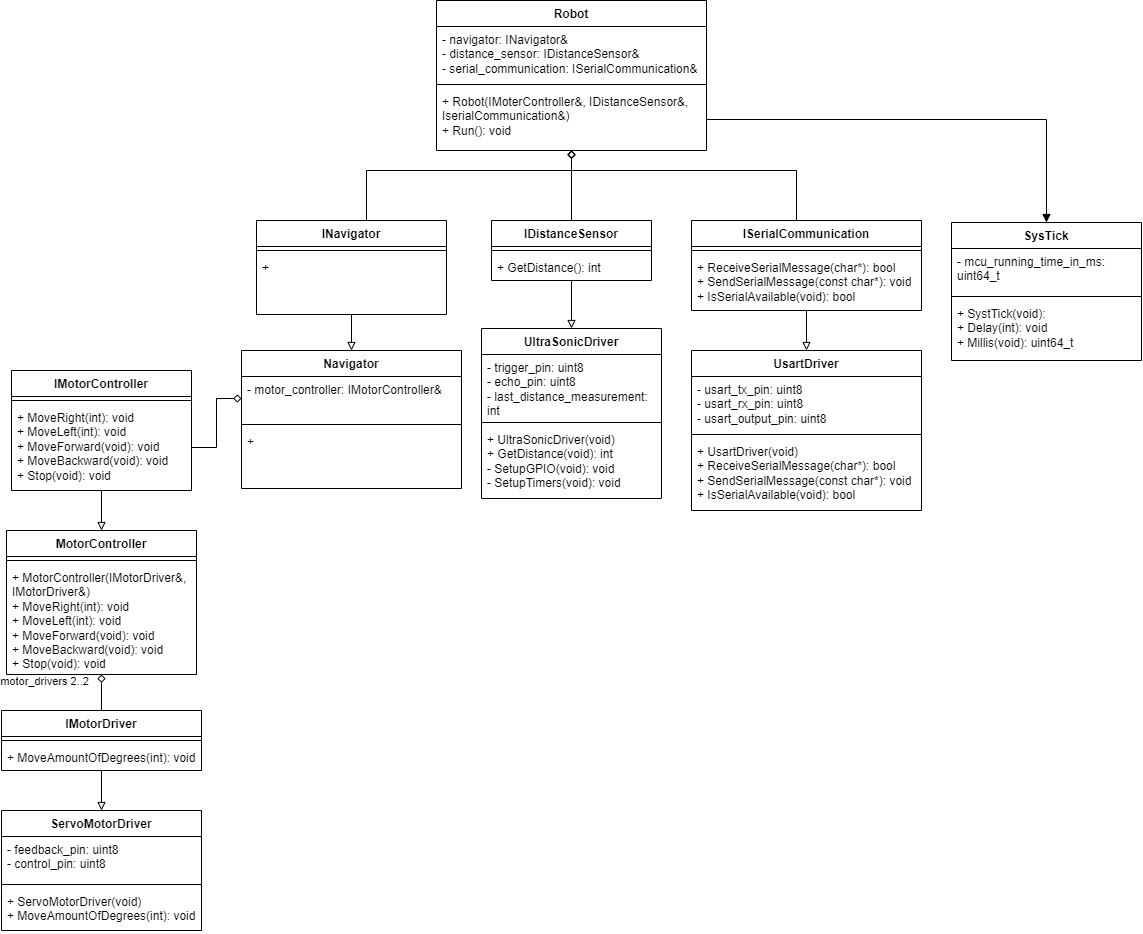
  
*Figure 1 - Flow diagram of the moving algorithm of the robot for phase 1*

Figure 1 shows the flow diagram for the maze robot for phase 1. These are based on the FR1-6. When the Car starts it first reads the distance is >10 cm it moves forward if not it applies the break using the PID, generating a random Angle either -90 or 90 degrees. Then Turn the Robot Angle.

## Class Diagrams

  
*Figure 2 - Class diagram of the robot for phase 1*

As shown in figure 2, The robot class is the central part of the code where everything is connected. The robot class will contain references to INavigator, IDistanceSensor and ISerialCommunication. The INavigator interface gets realized by the navigator class. The purpose of the navigator class is to decide what actions need to be performed based on made measurements. For phase 1 this will be a simple algorithm that simply chooses a direction if there is an obstacle. For phase 2 the navigator will contain a purpose fit maze solving algorithm. To control the movements of the robot the navigator contains a reference to IMotorController. IMotorController is realized by the MotorController class. This class contains two references to IMotorDriver. Each reference is to a different wheel. This allows the navigator class to control both servo motors from one place. The IMotorDriver is realized by the ServoMotorDriver class. This class is a device driver for the Parallax feedback 360 servo motor. As mentioned above, the navigator will make decisions based on made measurements. These measurements are made by an ultrasonic sensor. The UltraSonicDriver class is a device driver to make the ultrasonic sensor work. This class also realizes the IDistanceSensor interface. To allow a user to make changes to the preset thresholds serial communication is utilized. To be specific the USART protocol is utilized. This is implemented by the UsartDriver class that realizes the ISerialCommunication interface. Lastly, the SysTick class gives the robot class access to both Millis and Delay functions.

## State Diagrams

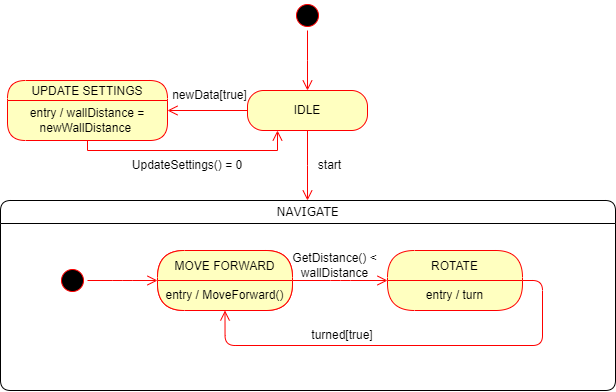
  
*Figure 3 - State machine of the robot for phase 1*

Figure 3 shows the state diagram for the robot for phase 1. The robot will first start at the IDLE state. When the robot receives new settings it goes to the UPDATE SETTINGS state. After setting everything up which is optional, the robot starts and goes to ‘NAVIGATE’. Inside, it starts going forward until the distance retrieved from the ultra sonic is smaller than the set threshold. If this is the case, it rotates and keeps going until it encounters a new obstacle. This cycle continues until there is either no power or the robot is manually shut down.

# Design Phase 2

# Implementation Phase 1

# Implementation Phase 2

# Testing Phase 1

# Testing Phase 2

# Reflection

# Bibliography