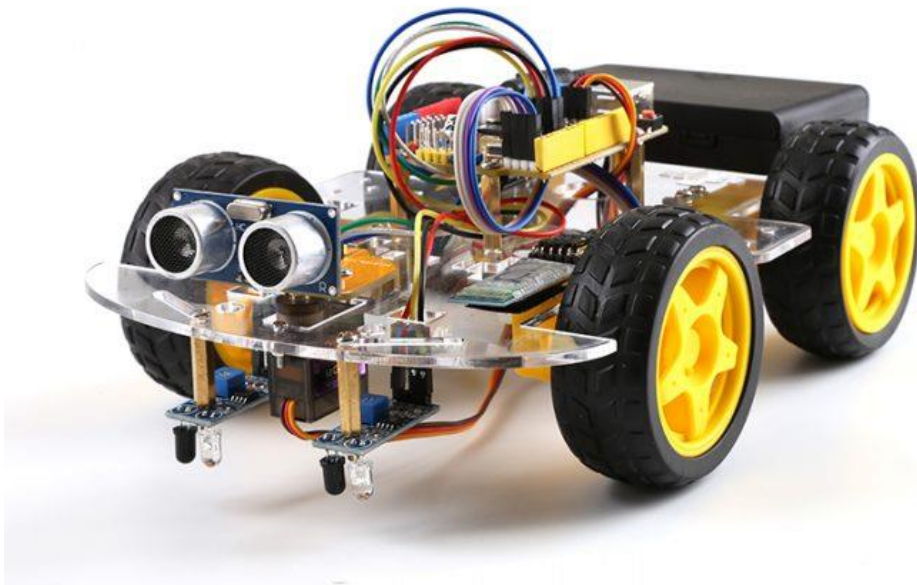


# Collision Avoidance Report

Name: Ahmed Emad Hassan



## Contents

Introduction.....	3
Project Description.....	3
System Architecting/Design Sequence .....	3
Case Study and Method .....	3
Software Life Cycle model used .....	3
Requirements .....	3
Customer Requirements.....	3
Requirement Diagram.....	4
Space Exploration/partitioning.....	4
Microcontroller.....	4
System Analysis .....	5
Use case Diagram .....	5
Activity Diagram .....	5
Sequence Diagram .....	6
System Design .....	6
Hardware Components .....	6
Block Diagram.....	7
State Machines for Every diagram.....	7
State Machine for US Sensor .....	7
State Machine for Collision Avoidance .....	8
State Machine for DC Motor.....	8
System Testing .....	9
Implementation .....	10
Debugging using GDB .....	10
Implementation Results .....	10

# Introduction

## Project Description

The Project is about Collision Avoidance, the car moves when Obstacles are above threshold distance and stops when Obstacles are near the car when Obstacles are below threshold

## System Architecting/Design Sequence

System Architecting and design follow this sequence:

1. Case Study and Method
2. Requirements
3. Space Exploration/partitioning
4. System Analysis
5. System Design

## Case Study and Method

### Software Life Cycle model used

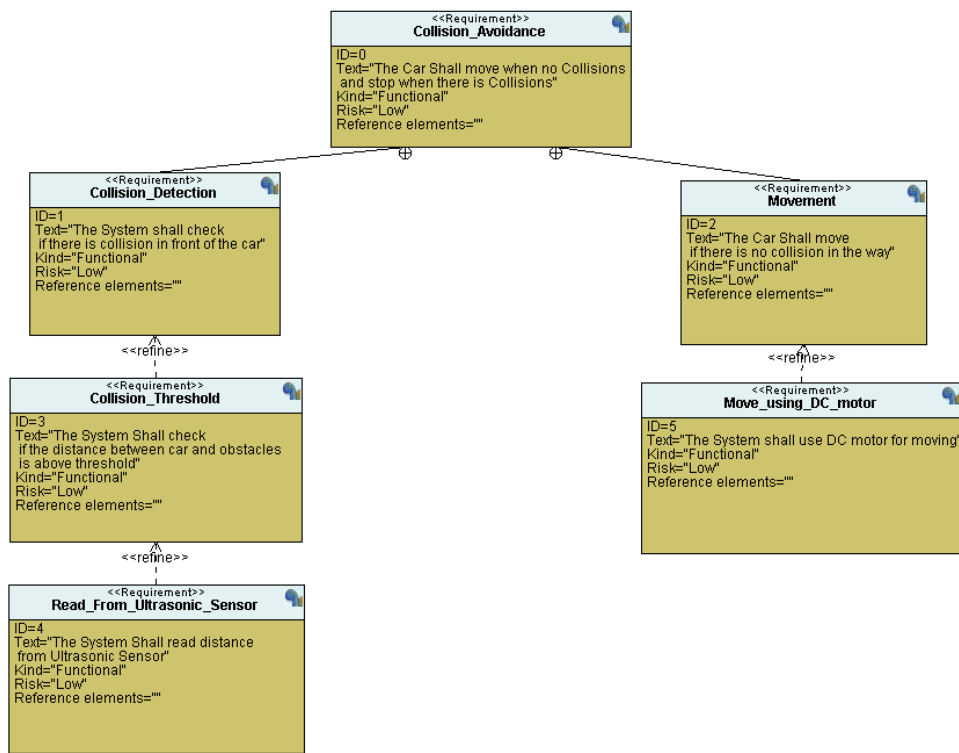
The software life cycle model will be used is V model

## Requirements

### Customer Requirements

Create a Collision Avoidance System and it should contain all necessary diagrams

## Requirement Diagram



## Space Exploration/partitioning

### Microcontroller

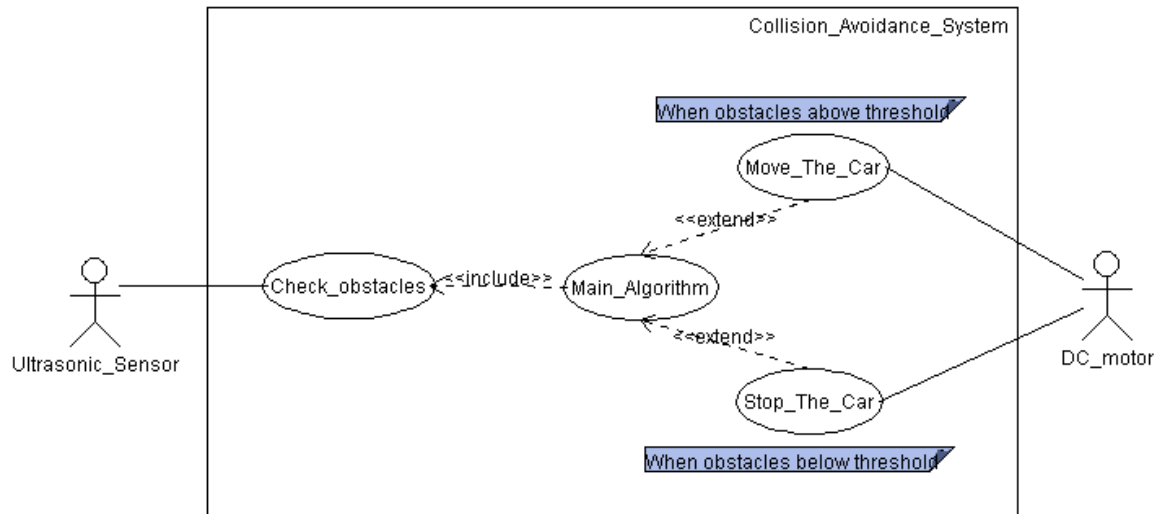
The microcontroller that will be used at this project is stm32

Reason:

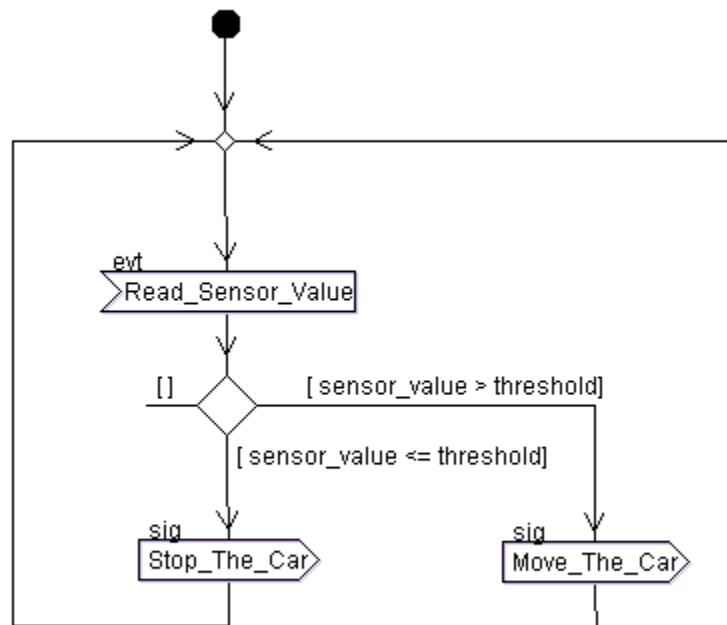
In a collision avoidance project using STM32, the exploration and partitioning of tasks is crucial for ensuring that the system operates efficiently and effectively. This involves breaking down the project into manageable components or modules, each responsible for specific functions. Below is a structured approach to project exploration and partitioning, focusing on the key tasks and subsystems involved.

# System Analysis

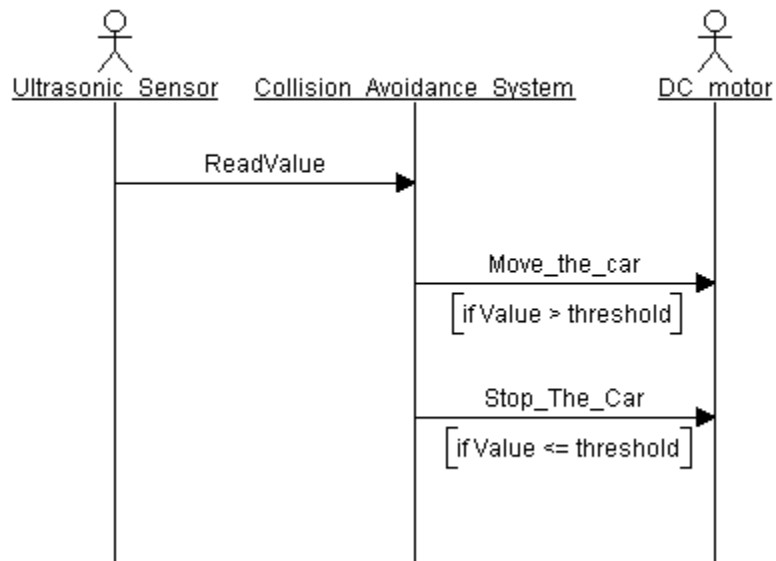
## Use case Diagram



## Activity Diagram

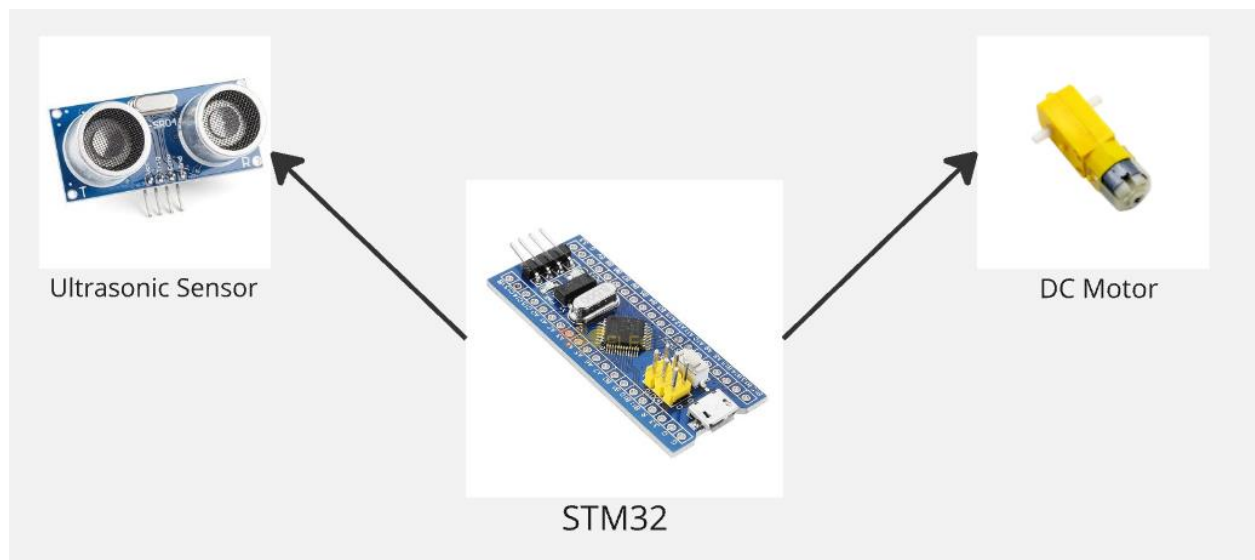


## Sequence Diagram

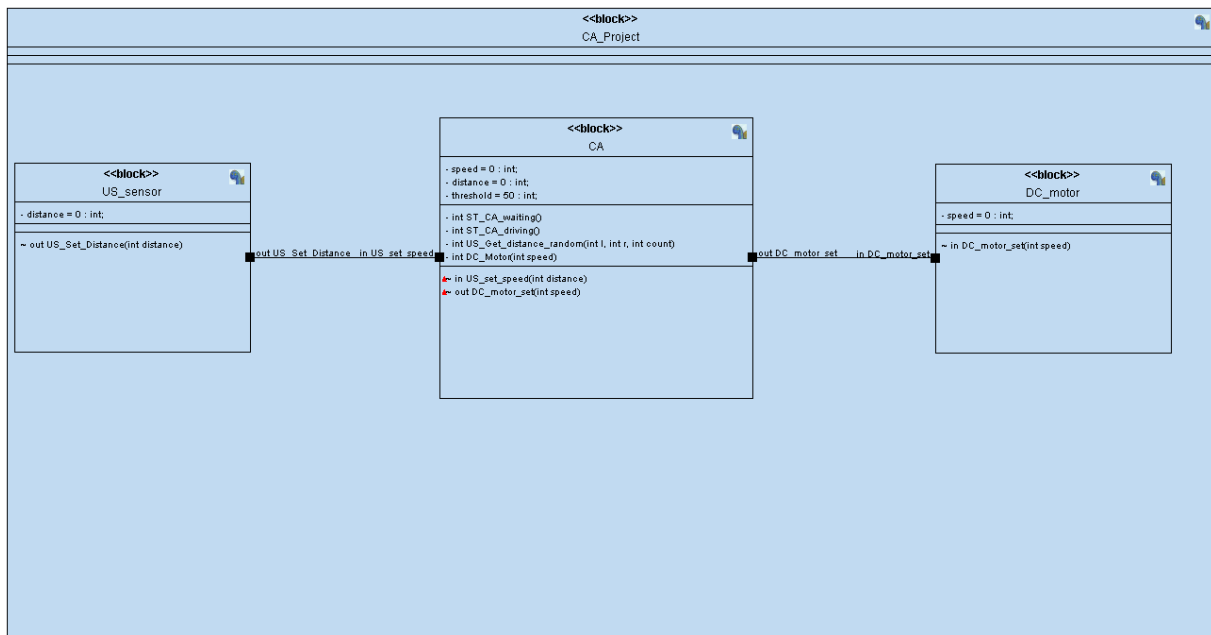


## System Design

### Hardware Components

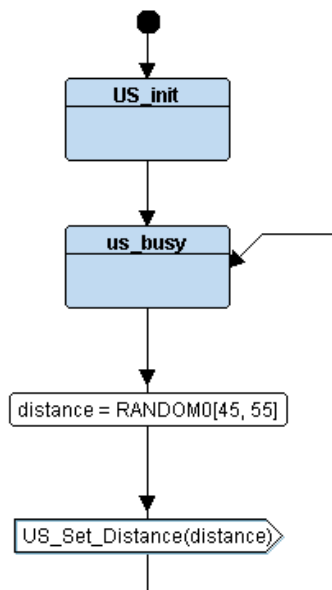


## Block Diagram

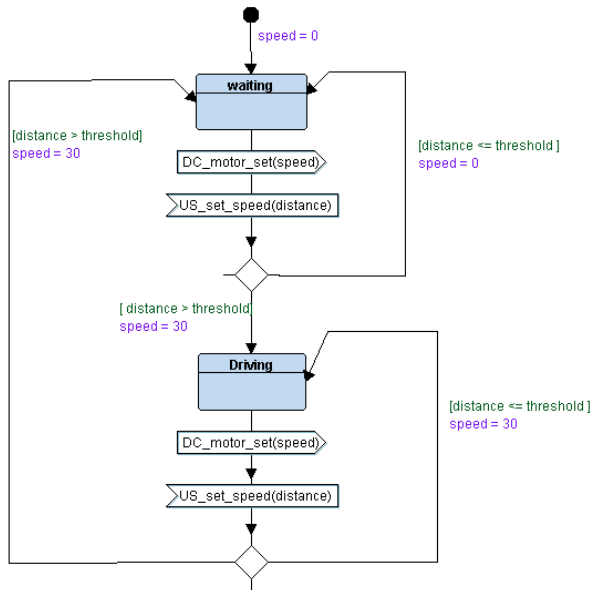


## State Machines for Every diagram

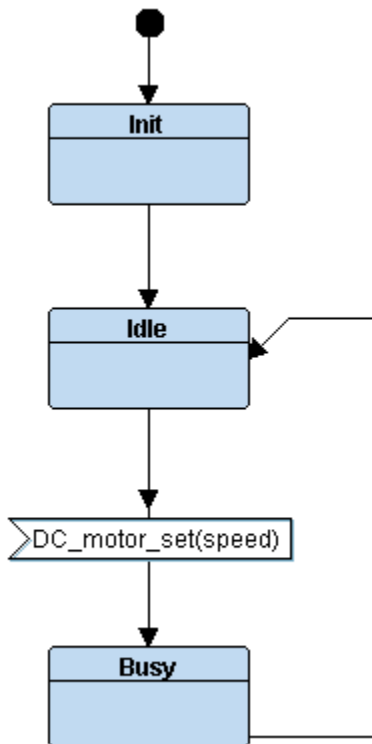
### State Machine for US Sensor



## State Machine for Collision Avoidance

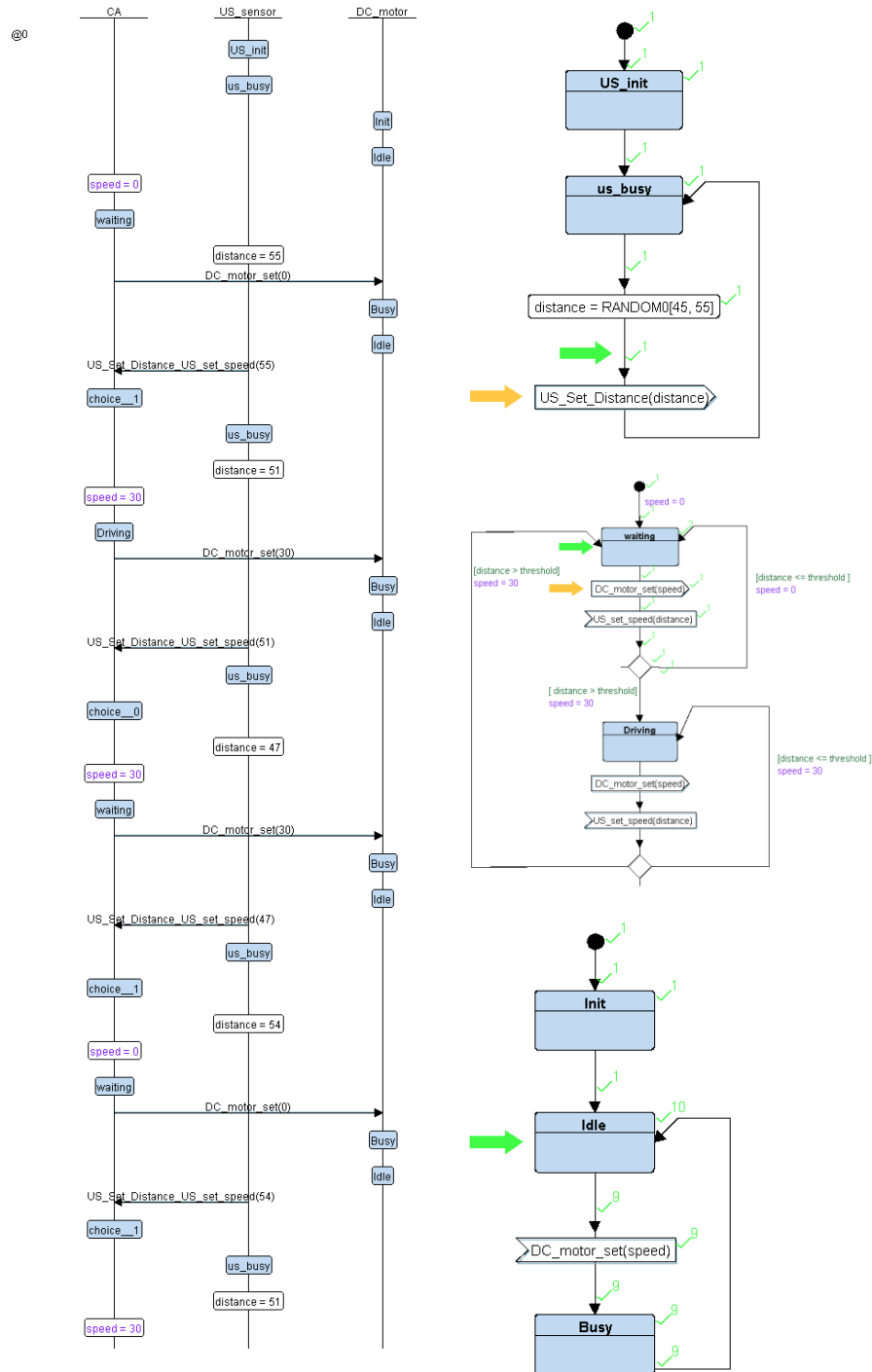


## State Machine for DC Motor





# System Testing



# Implementation

## Debugging using GDB

```
C:\Windows\System32\cmd.exe - gdb CA_test.exe

CA.c
16
17 //STATE Pointer to function
18 void (*CA_state)() = STATE(CA_waiting);
19
20 STATE_define(CA_waiting)
21 {
22     //state_name
23     CA_state_id = CA_waiting;
24     //state_action
25     CA_speed = 0;
26     //DC_motor(CA_speed)
27     CA_distance = US_Get_distance_random(45,55);
28     //event_check
29     (CA_distance <= CA_threshold)? (CA_state = STATE(CA_waiting)):(CA_state = STATE(CA_driving));
30     printf("CA_waiting state: distance %d Speed %d\n",CA_distance,CA_speed);
31     fflush(stdout);
32 }

native Thread 20804.0x1110 (src) In: ST_CA_waiting L23 PC: 0x7ff65e6416d8
[New Thread 20804.0x6680]

Thread 1 hit Temporary breakpoint 1, main () at main.c:19
(gdb) s
setup () at main.c:12
(gdb) s
main () at main.c:21
(gdb) s
ST_CA_waiting () at CA.c:23
(gdb) _
```

## Implementation Results

```
CA_waiting state: distance 52 Speed 0
CA_driving state: distance 47 Speed 30
CA_waiting state: distance 45 Speed 0
CA_waiting state: distance 51 Speed 0
CA_driving state: distance 53 Speed 30
CA_driving state: distance 52 Speed 30
CA_driving state: distance 55 Speed 30
CA_driving state: distance 46 Speed 30
CA_waiting state: distance 49 Speed 0
CA_waiting state: distance 54 Speed 0
CA_driving state: distance 50 Speed 30
CA_waiting state: distance 53 Speed 0
CA_driving state: distance 49 Speed 30
CA_waiting state: distance 49 Speed 0
CA_waiting state: distance 51 Speed 0
CA_driving state: distance 45 Speed 30
CA_waiting state: distance 55 Speed 0
CA_driving state: distance 55 Speed 30
CA_driving state: distance 51 Speed 30
CA_driving state: distance 51 Speed 30
CA_driving state: distance 48 Speed 30
CA_waiting state: distance 49 Speed 0
CA_waiting state: distance 55 Speed 0
CA_driving state: distance 54 Speed 30
CA_driving state: distance 52 Speed 30
CA_driving state: distance 45 Speed 30
CA_waiting state: distance 48 Speed 0
CA_waiting state: distance 52 Speed 0
CA_driving state: distance 51 Speed 30
```