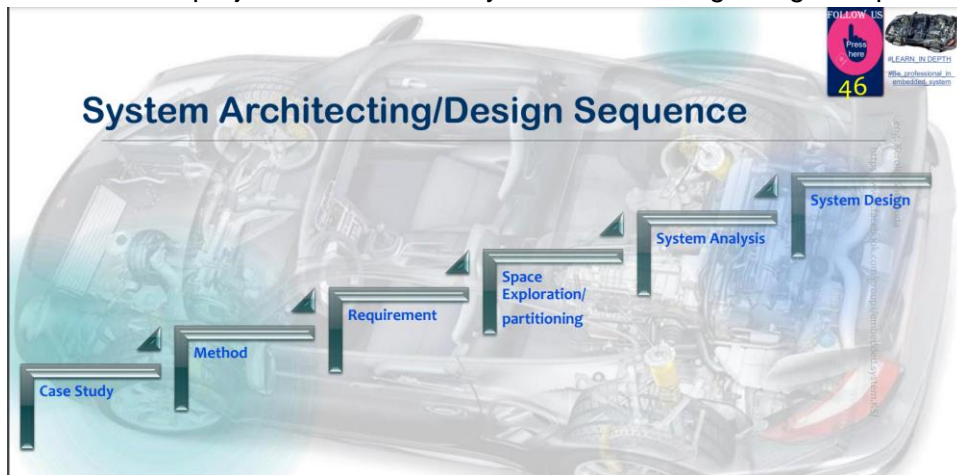# Collision Avoidance Project

To create this project we will follow System Architecting\design Sequence as shown below :

lets begin with Case study of the Project

Case study

Specification (from the client):
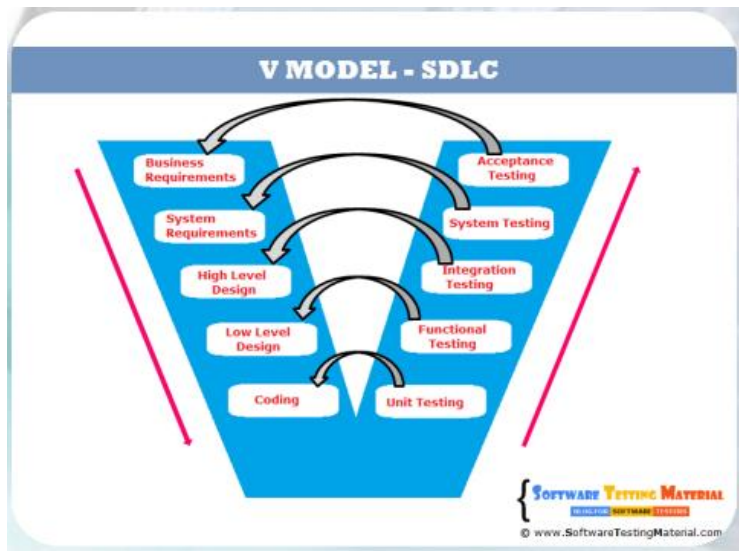- A system to stop the car from hitting an object while driving it

System Assumption:
- The car has an ulta sonic sensor
- The car is driven by 1 motor
- The sensor never fails
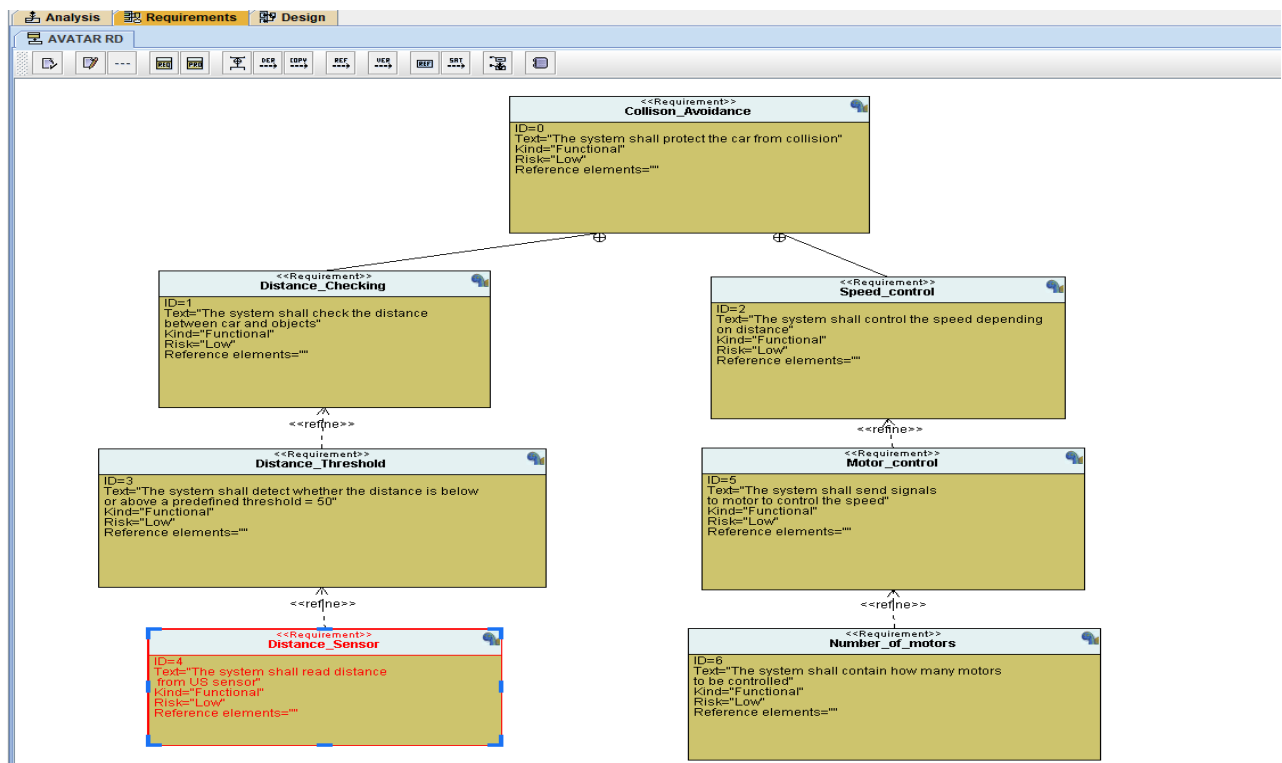- The motor never faces power cut

Now we will choose method

Method

In this project we will use V MODEL as shown below:

===
 Now lets use requirements nodes to define our requirements for this project

Requirements

Space Exploration\Partitioning

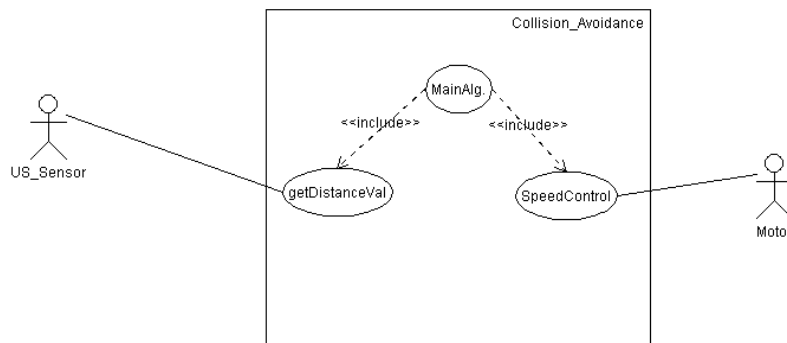In this project the most optimal way is to use Stm32

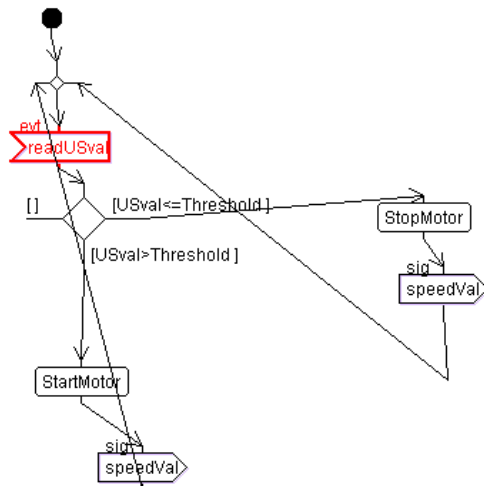===

System Analysis

System analysis has 3 Analysis methods:
   1- System boundary and main function --> Use case diagram
   2- Relation between main functions --> Activity Diagram
   3- Communications between main system entities and actors --> Sequence diagram
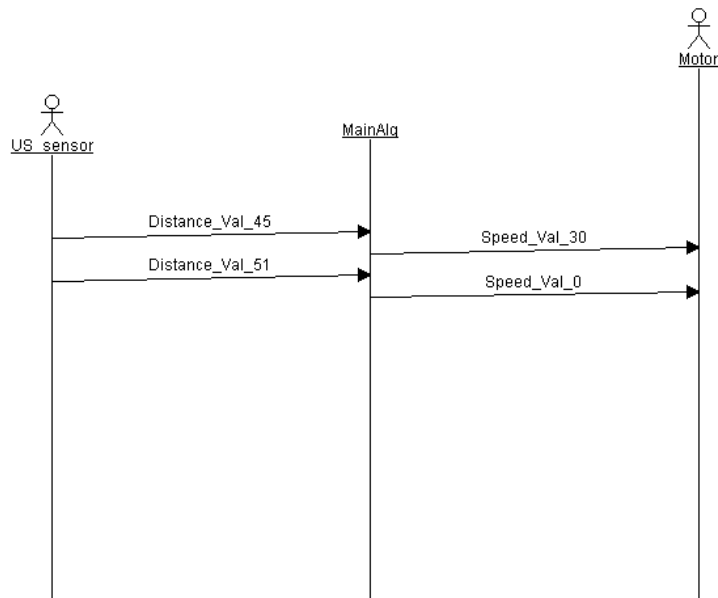
Let's begin with Use case diagram:

Activity diagram :

Sequence Diagram:

The last thing before implementation is to create System design

==

System Design

Block diagram for our system :
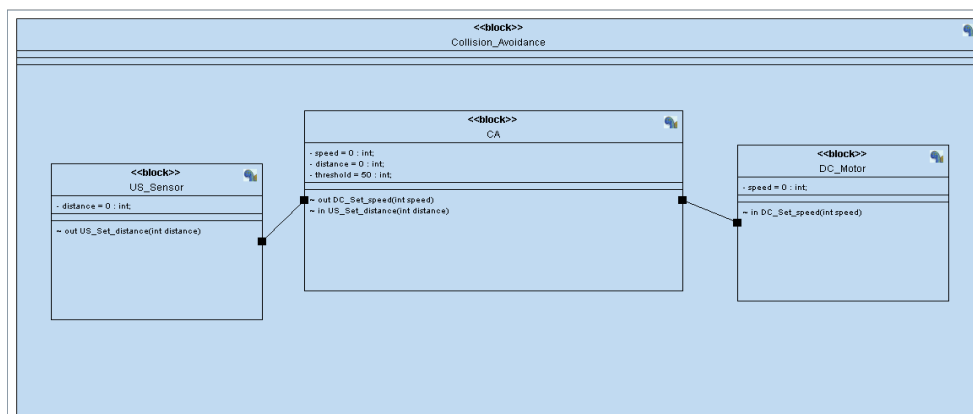
State machine for US_Sensor :
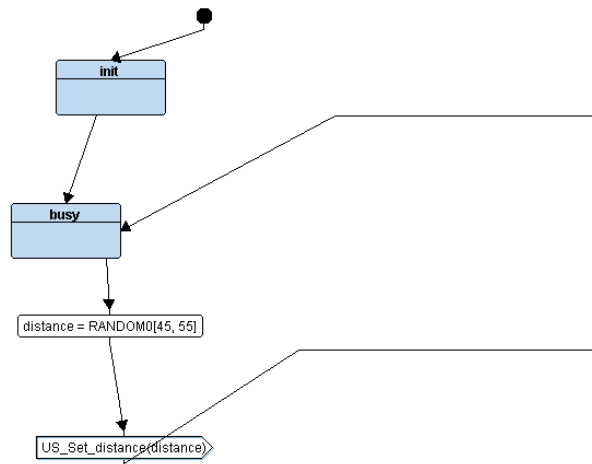
State machine for DC_Motor:
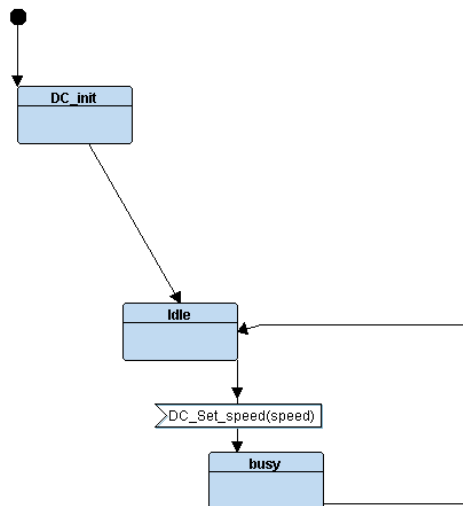
State machine for CA :
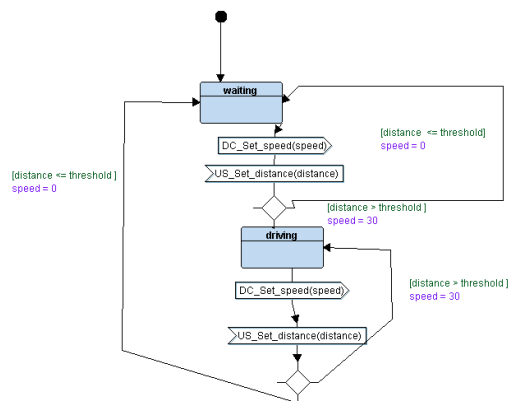
System Design Logic :
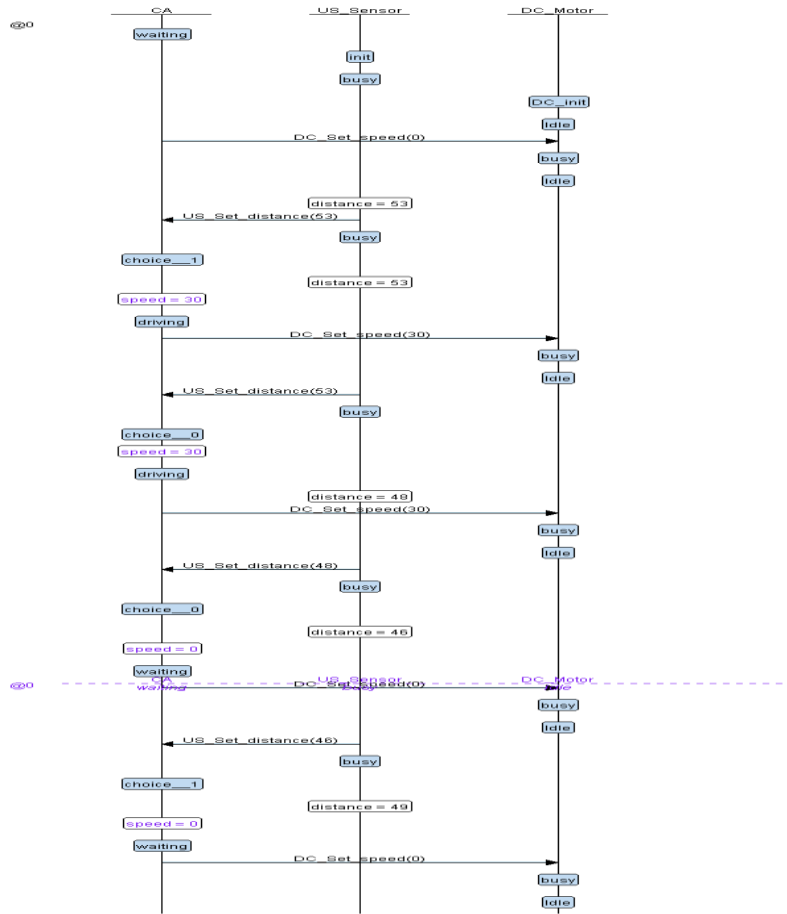
Finally, now we implement our code but is this project we can't simulate so we just run the project on Eclipse

The project files is uploaded here on github rep :

https://github.com/mgtera200/Master-Embedded-System/tree/main/System.Architecting_Unit_4/Assignments/Collision%20Avoidance

This is what the final result looks like :

main.c — editor

```c
#include "CA.h"
#include "DC.h"
#include "US.h"
void setup()
{
    US_init();
    DC_init();
    US_state = US_busy;
    DC_state = DC_busy;
    CA_state = CA_waiting;
}
int main() {

    setup();
    while (1)
    {
        US_state();
        CA_state();
        DC_state();

    }
}
```

Console output:

```
<terminated> (exit value: -1) Collision Avoidance.exe [C/C++ Application] C:\Users\LTC\workspace\C_Pro...
 CA >------SPEED=30---> DC
DC busy state : speed=30


 US >----distance=45--> CA
CA was driving state: current distance = 45 current speed = 30
 CA >-----SPEED=0----> DC
DC busy state : speed=0


 US >----distance=47--> CA
CA was in waiting state: current distance = 47 current speed = 0
 CA >-----SPEED=0----> DC
DC busy state : speed=0


 US >----distance=52--> CA
CA was in waiting state: current distance = 52 current speed = 0
 CA >-----SPEED=30---> DC
DC busy state : speed=30


 US >----distance=51--> CA
CA was driving state: current distance = 51 current speed = 30
 CA >-----SPEED=30---> DC
DC busy state : speed=30


 US >----distance=52--> CA
CA was driving state: current distance = 52 current speed = 30
 CA >-----SPEED=30---> DC
DC busy state : speed=30


 US >----distance=46--> CA
CA was driving state: current distance = 46 current speed = 30
 CA >-----SPEED=0----> DC
DC busy state : speed=0


 US >----distance=55--> CA
```