

NAME : Adem Alnajjar

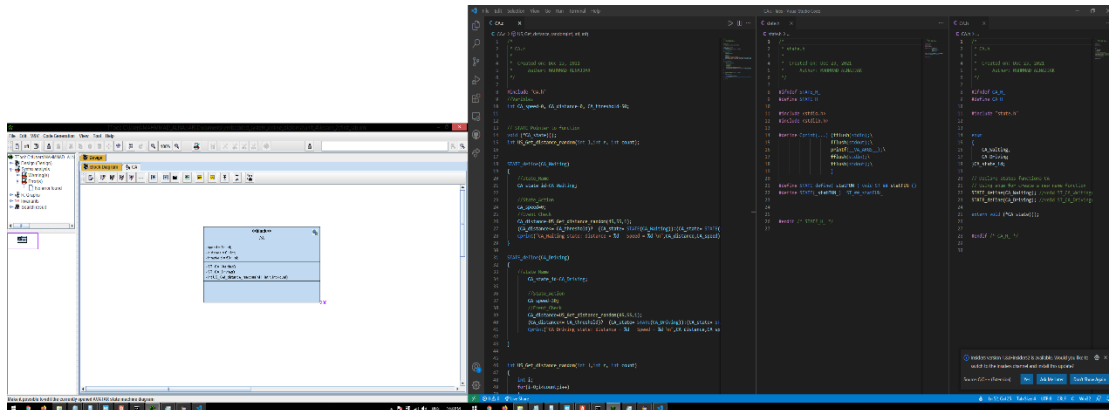
Mastering Embedded System Online Diploma

Requirements:

The car robot takes the distance from it to an obstacle from an ultrasonic sensor and compares this distance to a threshold equals 50cm. the car robot moves until the distance to obstacle becomes equal or less than 50 cm then the robot stop moving

1- Using Simple Module :

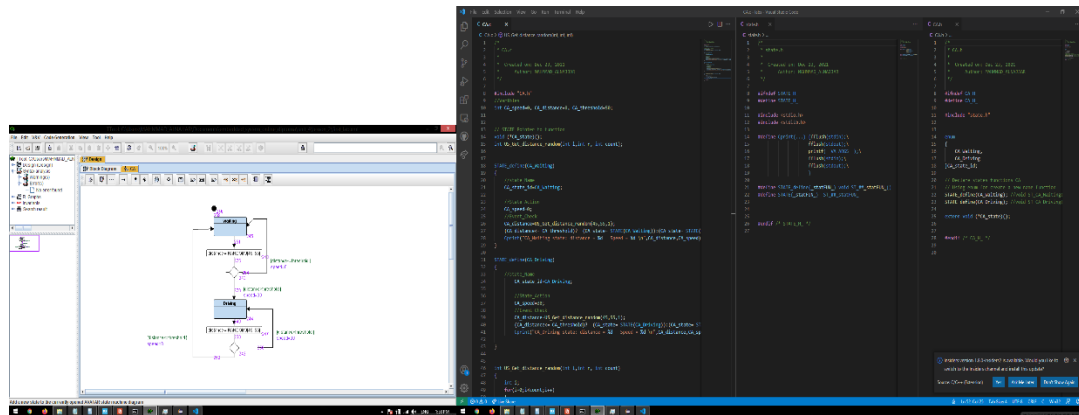
- In this method, we write all the libraries and functions of the elements in the same file and draw it in the same block



b. State Diagram

This diagram explains the two states of our system. First state is waiting state in which the car robot stop moving until distance became greater than 50 cm, in this case the robot switches from waiting state to driving state.

Second state is driving state in which the car robot moves with specific speed. When the distance became less than or equal 50 cm, the robot switches from driving state to waiting state and stop moving



c. Simulation



2- Using Multiple Modules:

Here we used three modules to describe the system one module for ultrasonic sensor, DC motor, module for controlling and one for connecting these modules

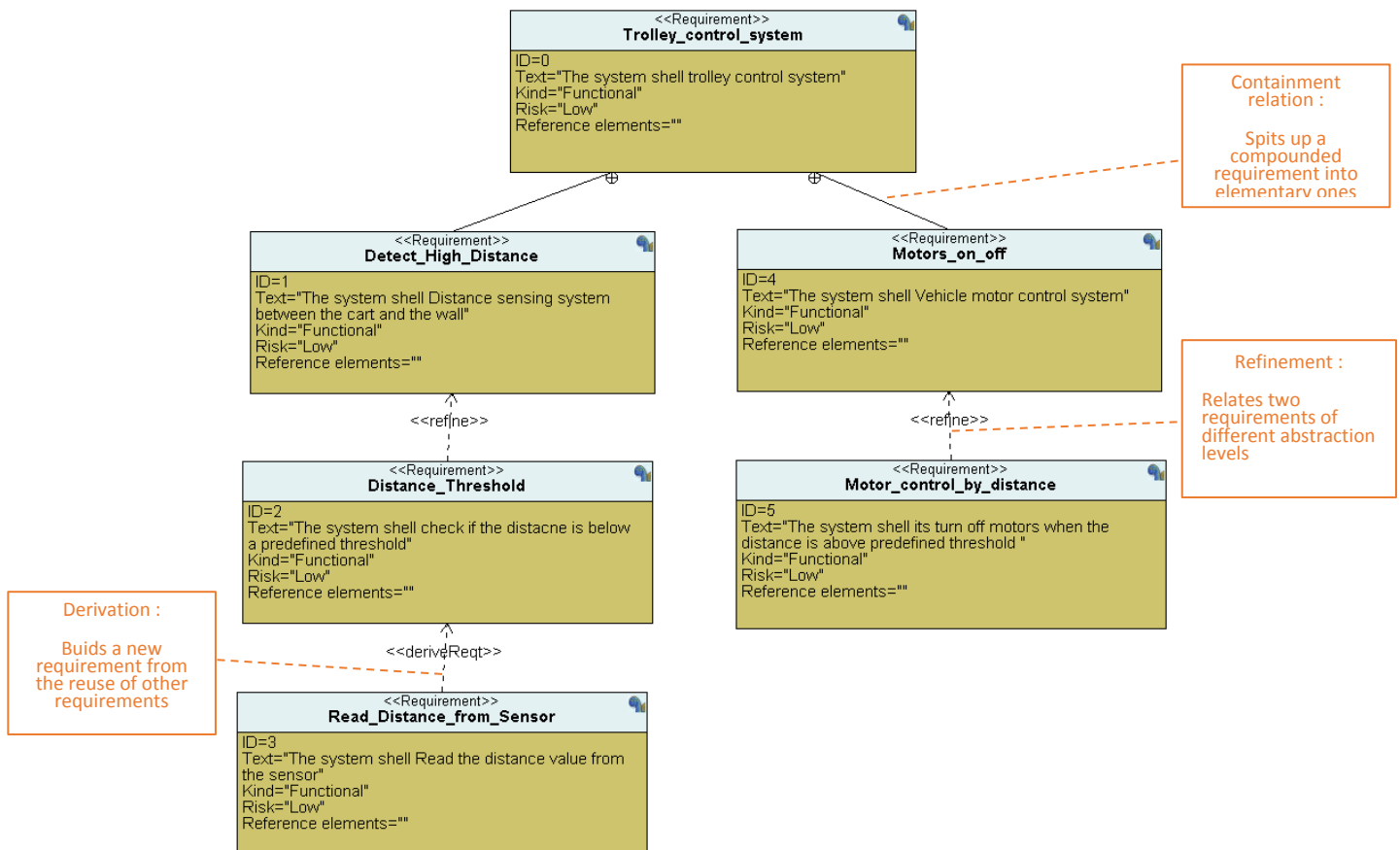
❖ Customer requirements:

The customer wanted the system that when the cart approaches a wall or object with a distance of about 30 centimeters, the cart stops and when the distance is greater than 30 centimeters, the cart continues to move

❖ Method:

In this project, we used this method : waterfall model

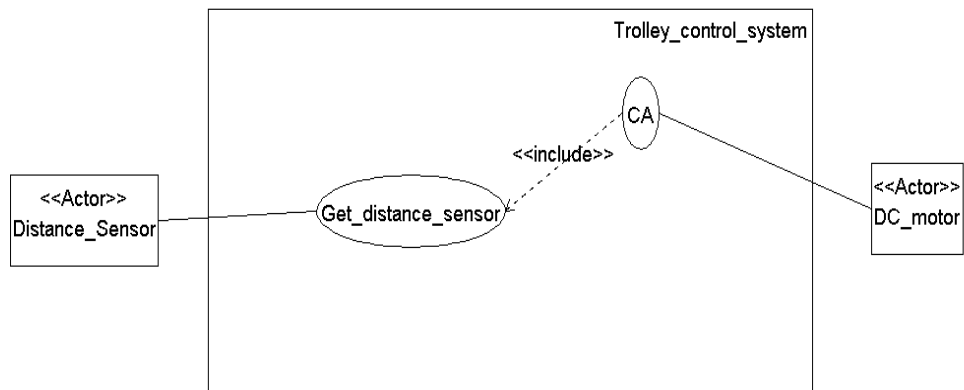
❖ Requirement :



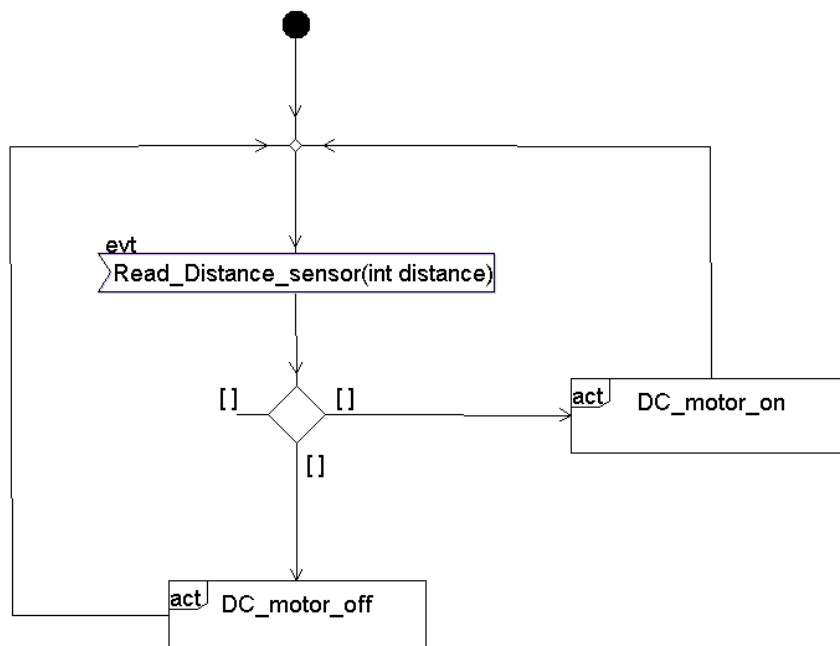
- Specification (from the client)
 - Control the vehicle system when you see a barrier or a wall

❖ System Analysis :

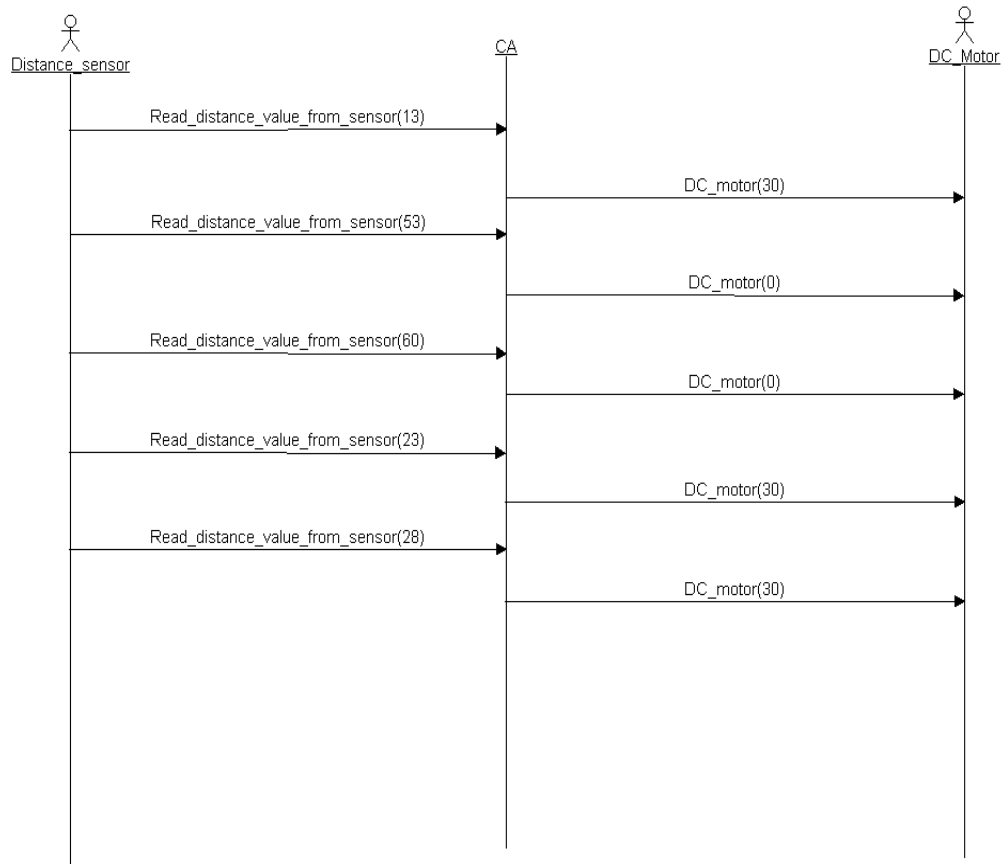
- System boundary and main functions :



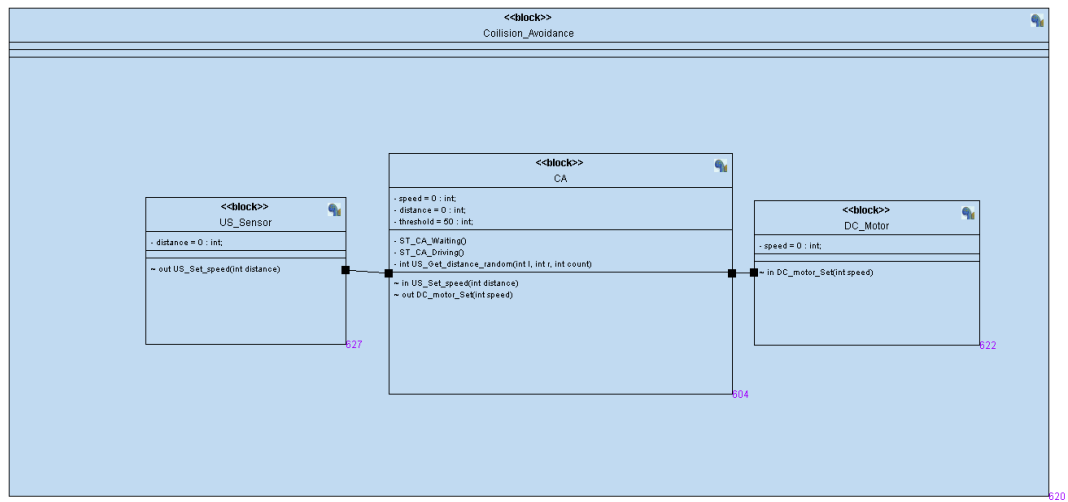
- Relations between main functions :



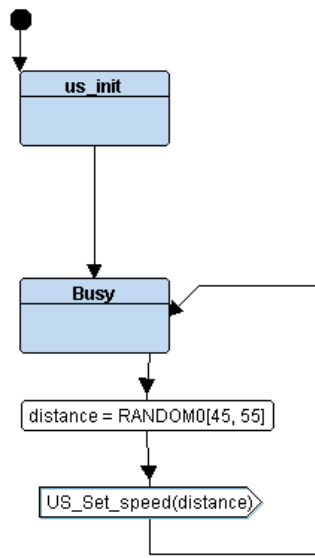
- Communications between main system entities and actors :



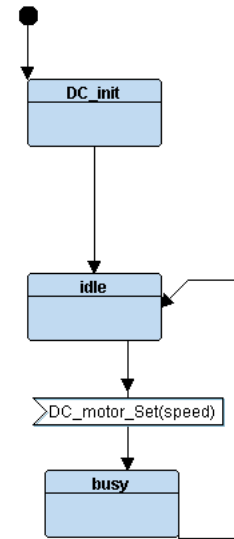
❖ System Design : (state machine)



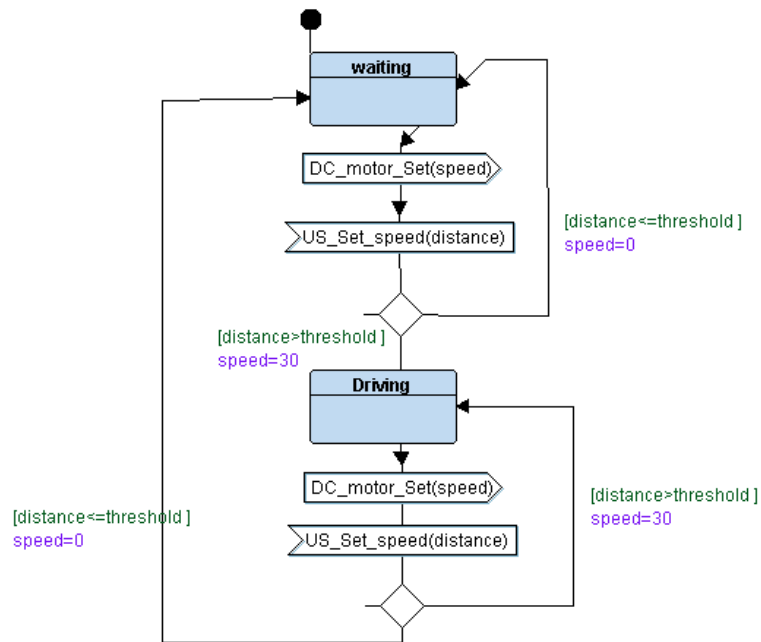
(Instance) Block Diagram



Read distance from sensor



DC Motor control



Main Algorithm

- **State Diagram (Ultrasonic sensor)**

We first initialize the sensor then the sensor working in busy state. In busy state the sensor reads distance and send it and return again to busy state

- **State Diagram (DC motor)**

After initializing DC motor it goes to idle state in which motor doesn't move. After comparing distance we send speed to DC motor and it goes to busy state and motor moves

- **State Diagram (CA Main Algoritma)**

Here it is the same like one module diagram and the same two states driving and waiting, but here it takes distance from ultrasonic sensor and send speed to dc motor after comparing distance with threshold.

❖ Simulation :

