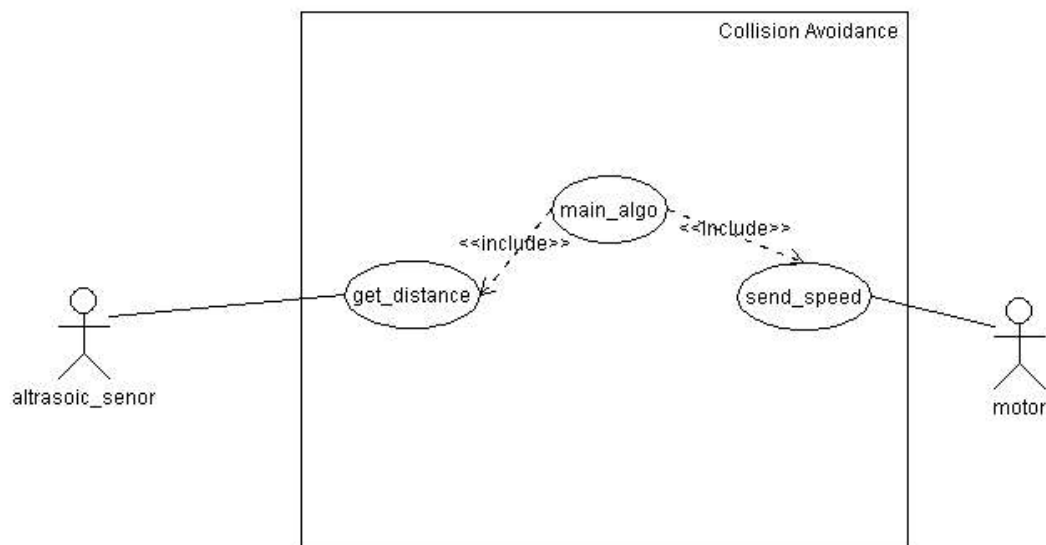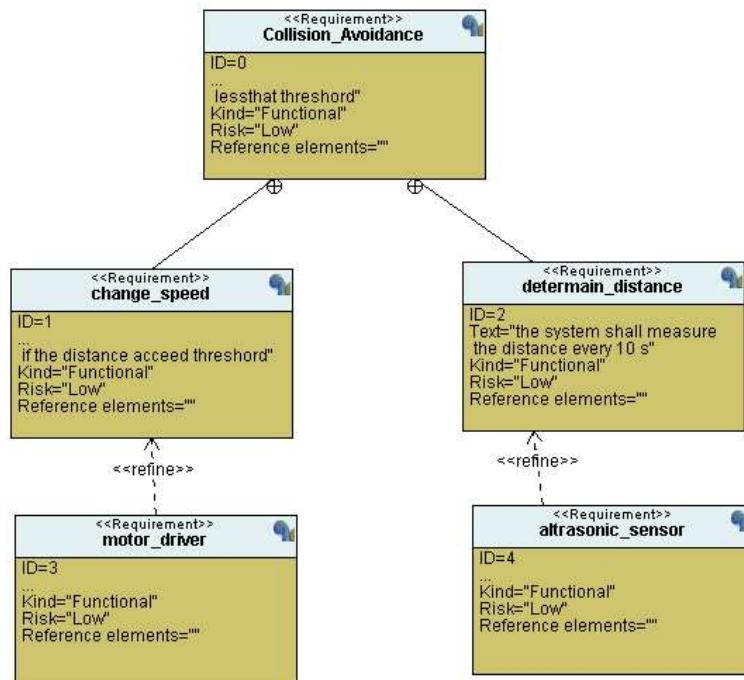Case :
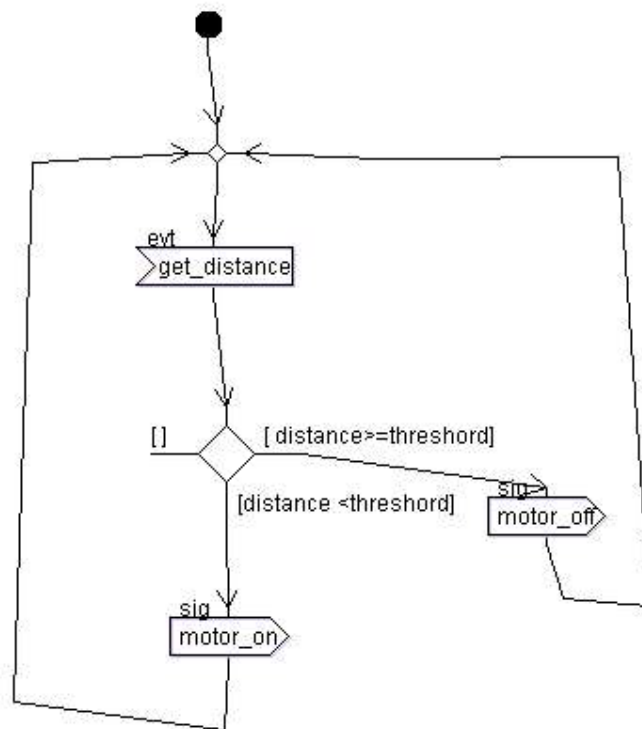
A car that avoid obstacles by detecting distance between the car and the obstacle if it is above a threshold =50

Use case :

requirements digram
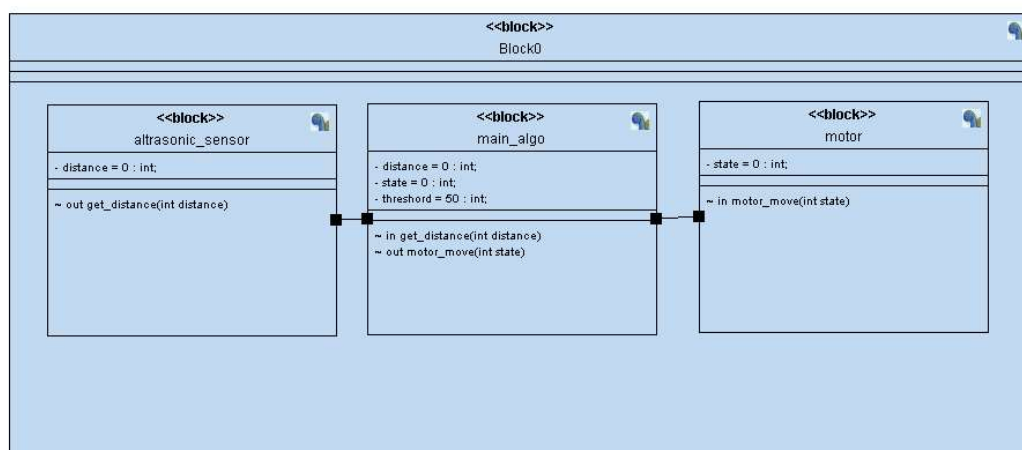

<<Requirement>>
Collision_Avoidance

ID=0
...
"lessthat threshord"
Kind="Functional"
Risk="Low"
Reference elements=""

<<Requirement>>
change_speed

ID=1
...
"if the distance acceed threshord"
Kind="Functional"
Risk="Low"
Reference elements=""

<<Requirement>>
determain_distance

ID=2
Text="the system shall measure
the distance every 10 s"
Kind="Functional"
Risk="Low"
Reference elements=""

<<refine>>

<<refine>>

<<Requirement>>
motor_driver

ID=3
...
Kind="Functional"
Risk="Low"
Reference elements=""

<<Requirement>>
altrasonic_sensor

ID=4
...
Kind="Functional"
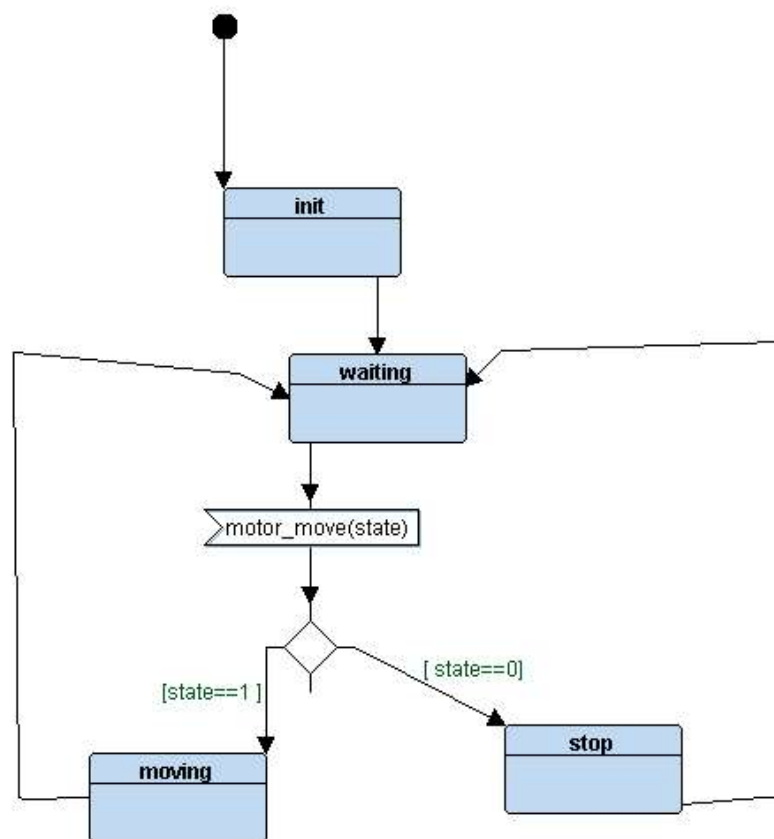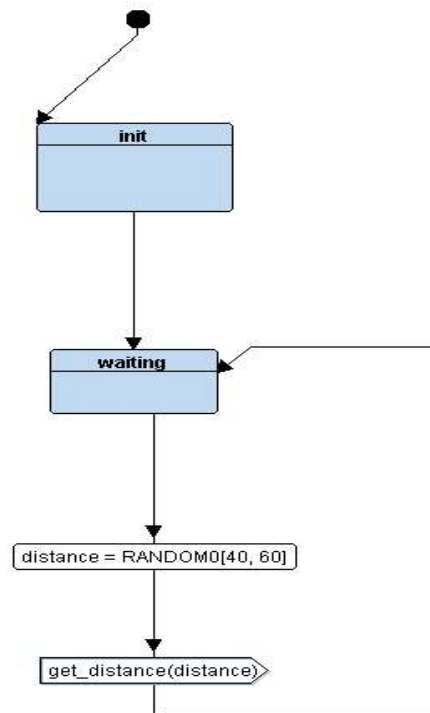Risk="Low"
Reference elements=""

acrivity diagram

Block diagram

Motor driver diagram

Sensor driver

main_algorithm

waiting

motor_move(state)

get_distance(distance)

[distance >=threshord]
state =0

[distance<threshord]
state =1

driving

motor_move(state)

get_distance(distance)

[distance>=threshord]
state =0

[distance<threshord]

Block0   motor   main_algo   altrasonic_sensor

@0

runing

init

waiting

waiting

init

waiting

motor_move(0)

choice__0

stop

waiting

distance = 40

get_distance(40)

waiting

@0   Block0   motor   main_algo   altrasonic_sensor   @0
      runing   waiting   choice__0   waiting

distance = 59

state = 1

driving

motor_move(1)

choice__0

moving

waiting