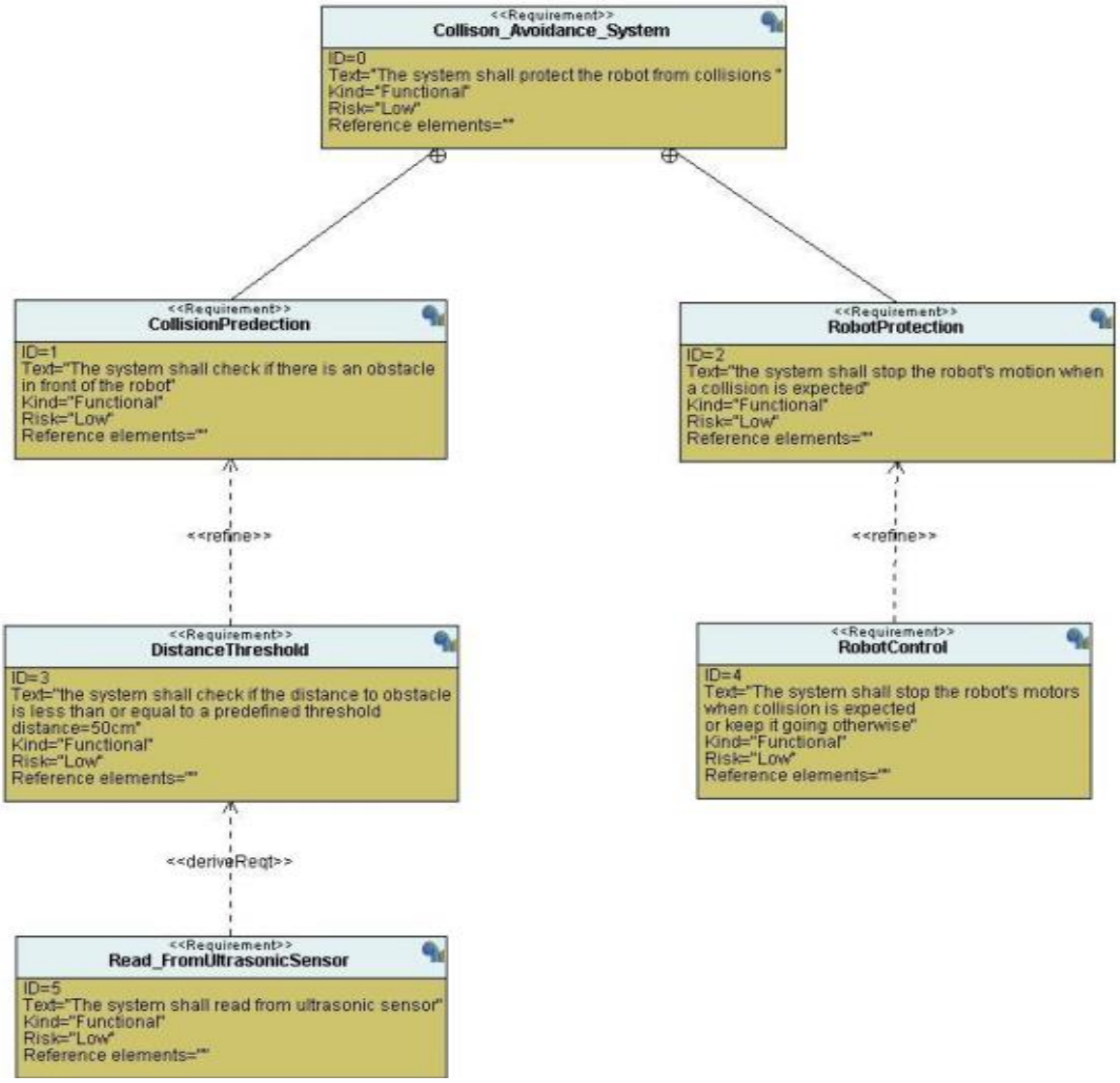
Collision Avoidance project

ENG: Mostafa Hassan Aboshaker

Introduction:

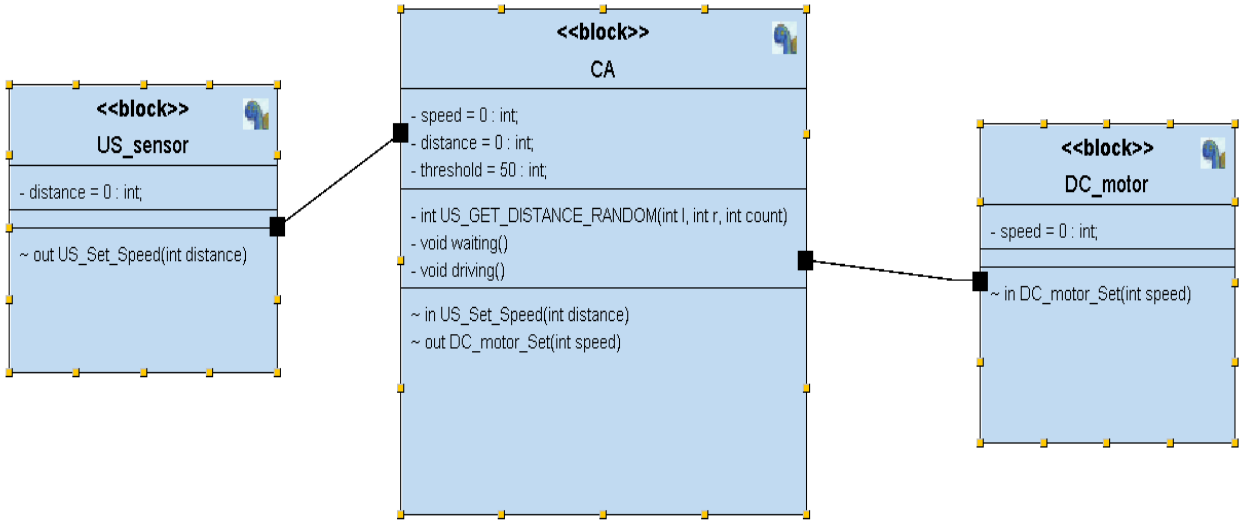
The Collision Avoidance Project is an initiative aimed at developing advanced systems and technologies to prevent collisions and enhance safety in various domains, such as transportation, robotics, and industrial settings. The primary goal of the Collision Avoidance Project is to reduce the risk of accidents and improve overall safety by implementing proactive measures that can prevent collisions from occurring in the first place. This involves using a combination of sensors, algorithms, and real-time data analysis to enable autonomous decision-making and prompt action.

Requirement diagram:

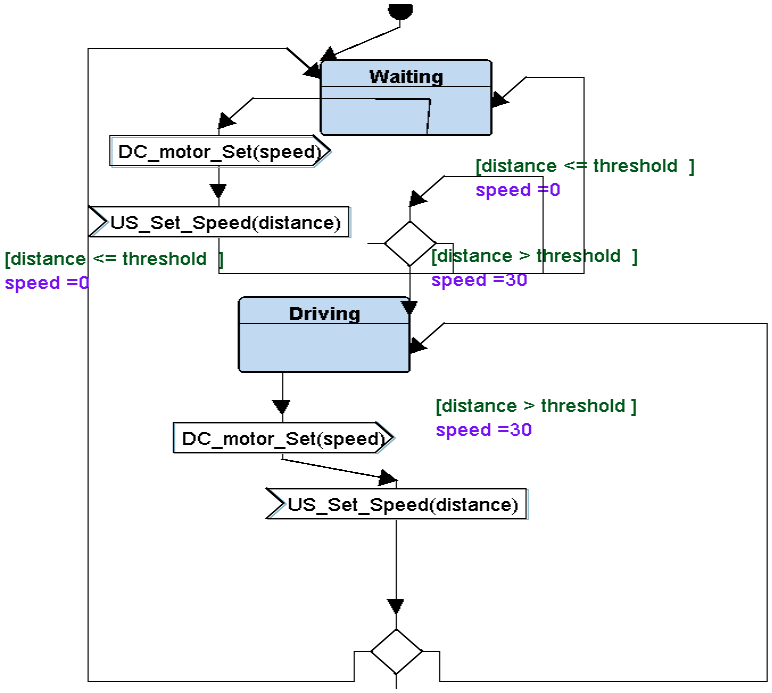


Block diagram:

in the block diagram we will try to model a simple collision avoidance system which consists of an ultrasonic sensor an algorithm to decide if the actuators (motors) should run or stop and DC motor which is considered the output, to do so we used the TTool. First, we created a block diagram for the whole system as shown in the figure below, we defined the variables speed distance and threshold then we tried to make a flowchart based on the states in which the system can be.



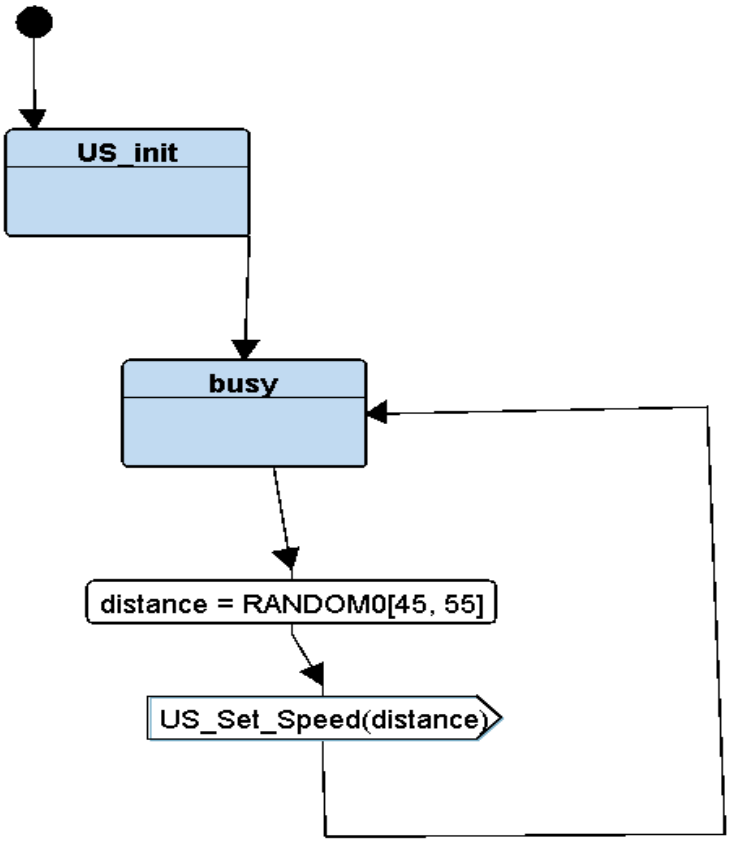
Collision avoidance design:



```
#include "CA.h"
int CA_speed=0,CA_distance=0,CA_threshold=50;
void (*CA_state)();
void US_set_distance(int d)
{
	CA_distance=d;
	if(CA_distance<=CA_threshold)
		CA_state=waiting;
	else CA_state=driving;
	printf("\n US----distance=%d----->CA",CA_distance);}
void waiting()
{   //state name
	CA_STATUS=CA_waiting;
	printf("\n CA_waiting state: distance=%d  speed=%d",CA_distance,CA_speed);
	//state action
	CA_speed=0;
	DC_motor(CA_speed);}
void driving()
{//state name
	CA_STATUS=CA_driving;
	printf("\n CA_driving state: distance=%d  speed=%d",CA_distance,CA_speed);
	//state action
	CA_speed=30;
	DC_motor(CA_speed);}
```

Ultrasonic Sensor design:

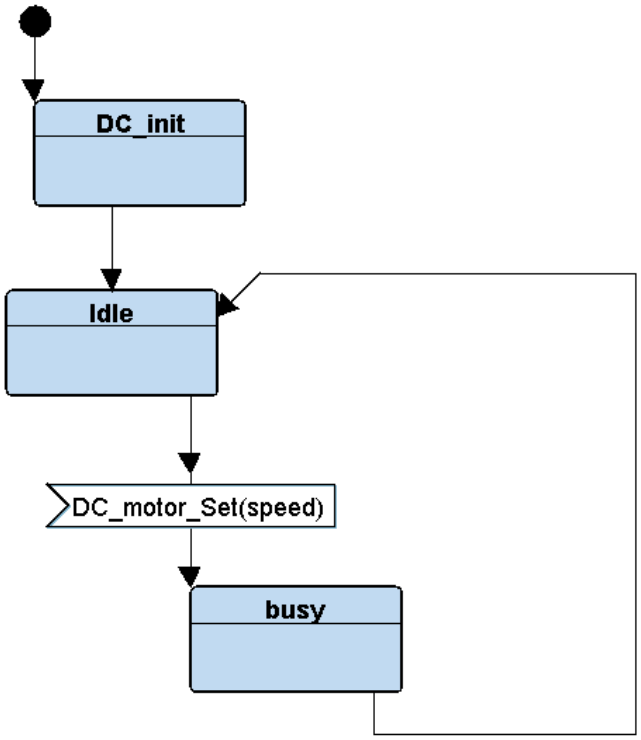
Ultrasonic sensors are commonly used in collision avoidance systems due to their effectiveness in detecting objects and measuring distances. These sensors emit high-frequency sound waves (ultrasonic waves). In a collision avoidance system, ultrasonic sensors are typically placed strategically on a vehicle or a robot to provide a 360-degree view of the surroundings. These sensors continuously emit ultrasonic waves and receive the reflections from nearby objects.



In this application we will git the sensor readings from random function that provides us with different values. These random values will be from 45 to 55.We chose these limits because we defined the Threshold value=50.

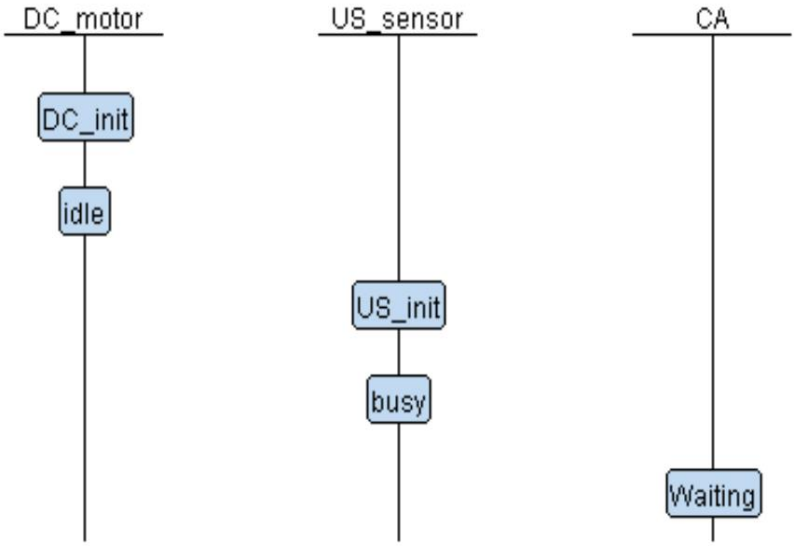
DC-motor design:

DC motors can play a crucial role in collision avoidance systems, particularly in applications that involve physical movement or actuation. It can be used to control the movement of various components within a collision avoidance system. In automotive collision avoidance systems, DC motors are commonly used in power steering mechanisms. These motors assist in steering the vehicle and can be controlled by the collision avoidance system to make quick adjustments or corrections to avoid potential collisions.



Results:

after analyzing the syntax of the system, we can simulate interactively the system, the tool shows you the variables, the states and other parameters to help you on understanding and rectifying the system logic.



US_waiting state: distance=46

US----distance=46----->CA

CA_waiting state: distance=46 speed=0

CA----speed=0----->DC

DC_busy state: speed=30

US_waiting state: distance=52

US----distance=52----->CA

CA_driving state: distance=52 speed=30

CA----speed=30----->DC

DC_busy state: speed=30

US_waiting state: distance=50

US----distance=50----->CA

CA_waiting state: distance=50 speed=0

CA----speed=0----->DC

DC_busy state: speed=30

US_waiting state: distance=50

US----distance=50----->CA

CA_waiting state: distance=50 speed=0

CA----speed=0----->DC

DC_busy state: speed=30

US_waiting state: distance=55

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

CA_driving state: distance=55 speed=30

CA----speed=30----->DC

DC_busy state: speed=30