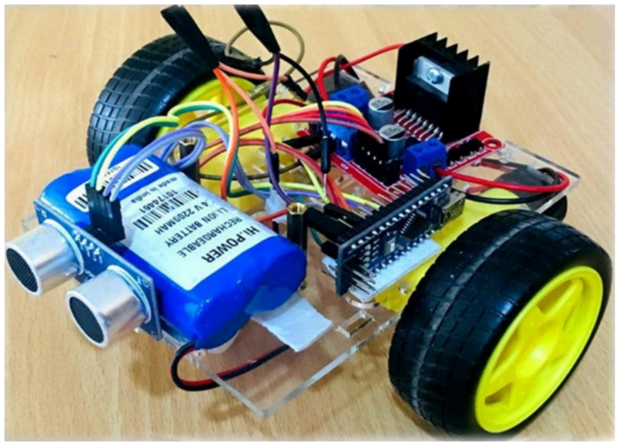
Collision Avoidance Report 📄:



By | Eng Belal Hani Abu Sabha .

1-Case Study :

Specifications :

1-Ultra-Sonic Sensor calculate distance to control in motor speed .

2-if distance below 50 then speed=0 else speed=30.

Assumptions:

1-The controller set up and shutdown procedures are not modeled .

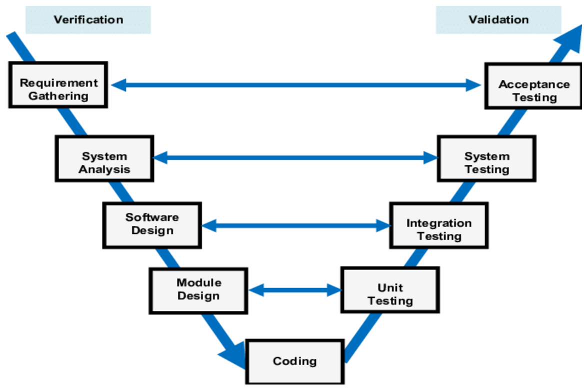
2-The controller maintenance is not modeled .

3-The Ultra-Sonic Sensor never fails .

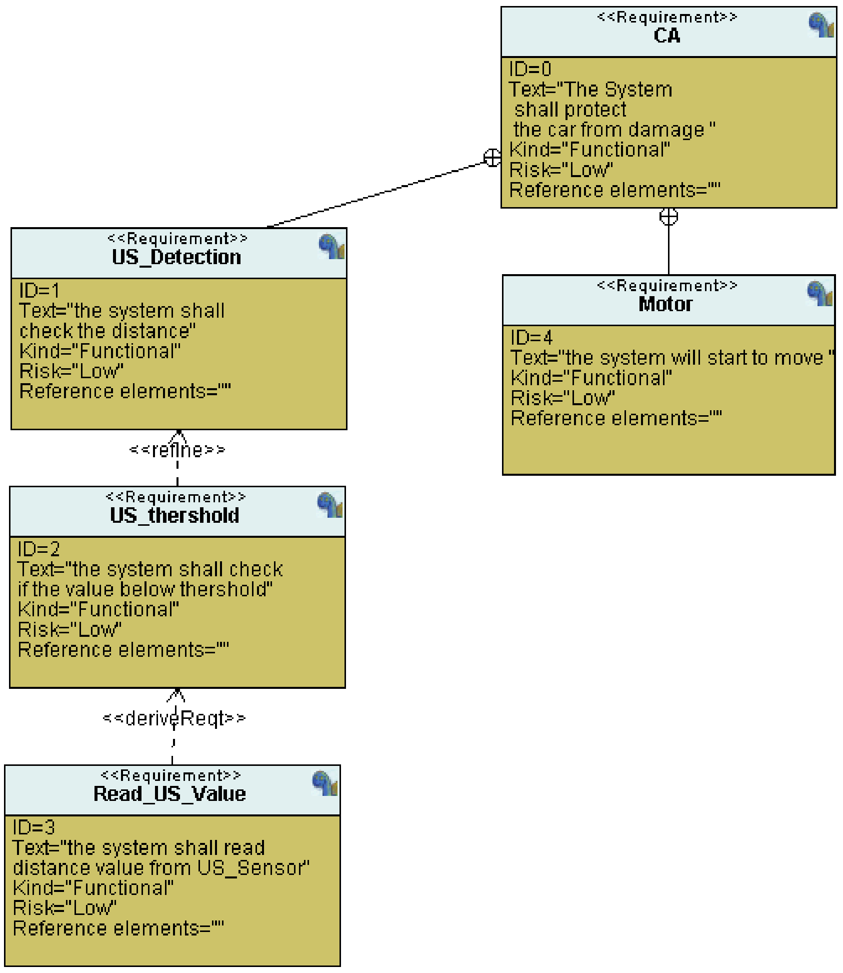
4-The DC Motor never fails .

2-Method:

We will use V-Model.



3-Requirments:

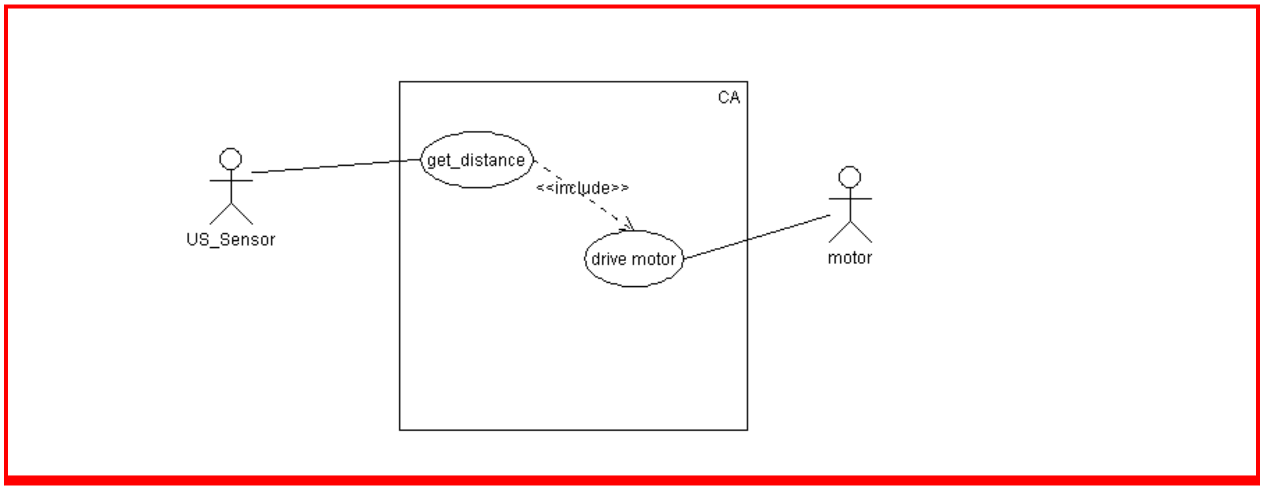


4- Space Exploration:

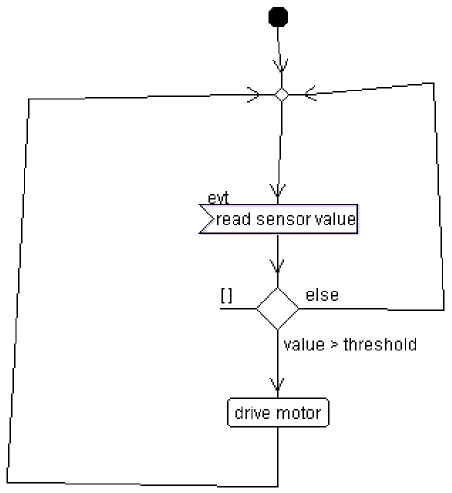
We will use STM32F103C6 microcontroller that uses arm cortex-M3 .

5- System Analysis:

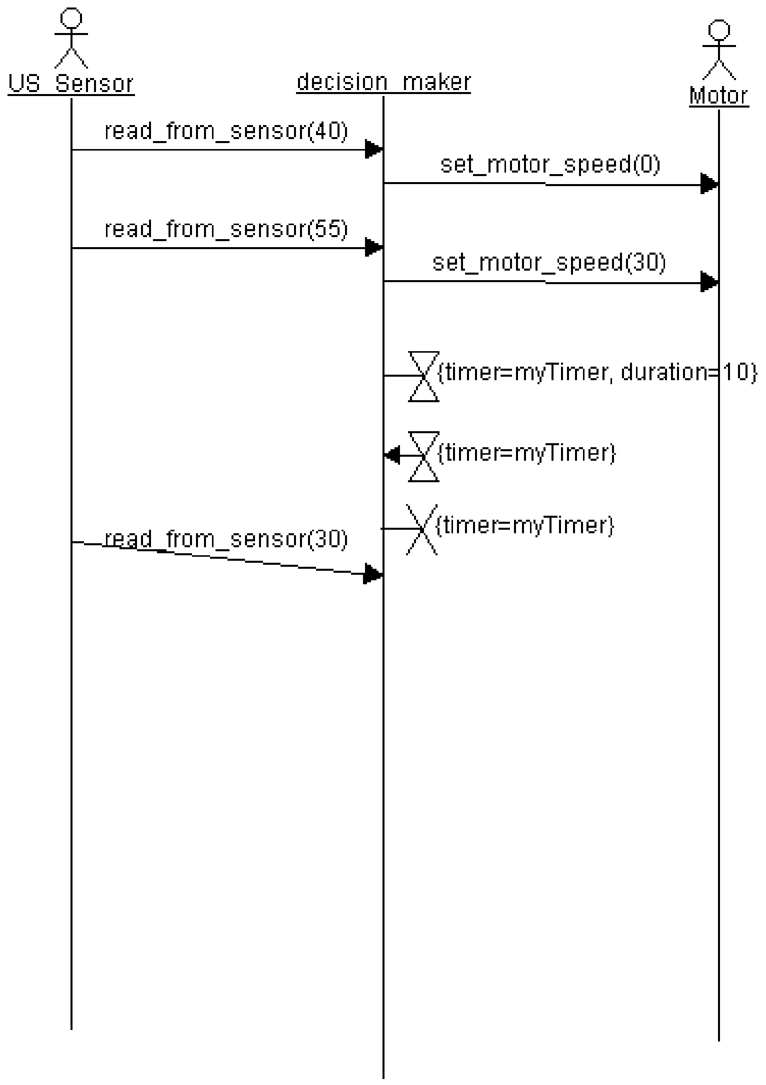
1- Use Case Diagram:



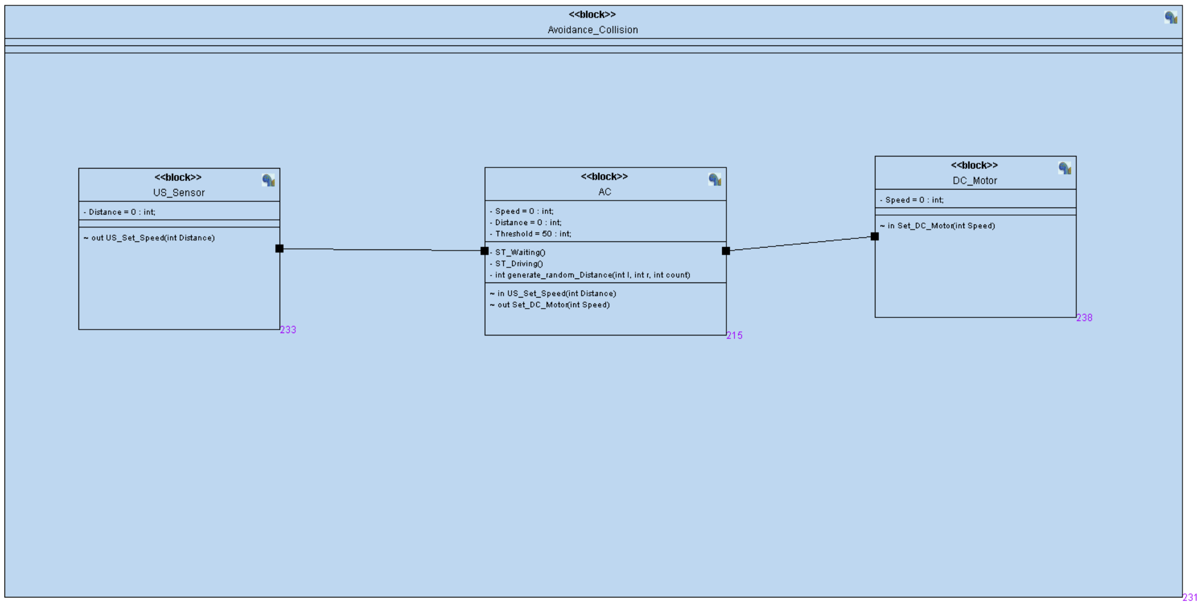
2-Activity Diagram:



3-Sequence Diagram:

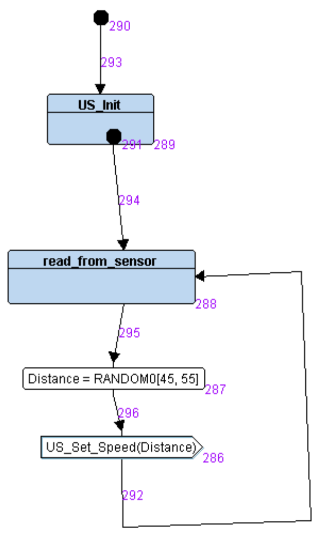


6-System Design:

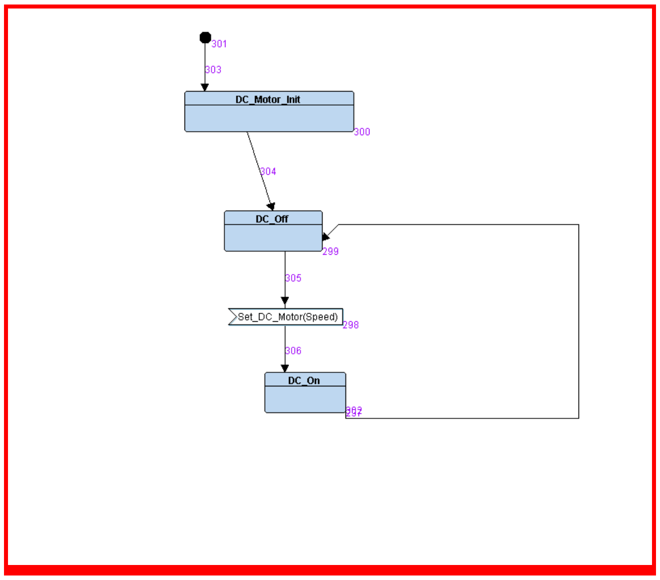


State machine for every block:

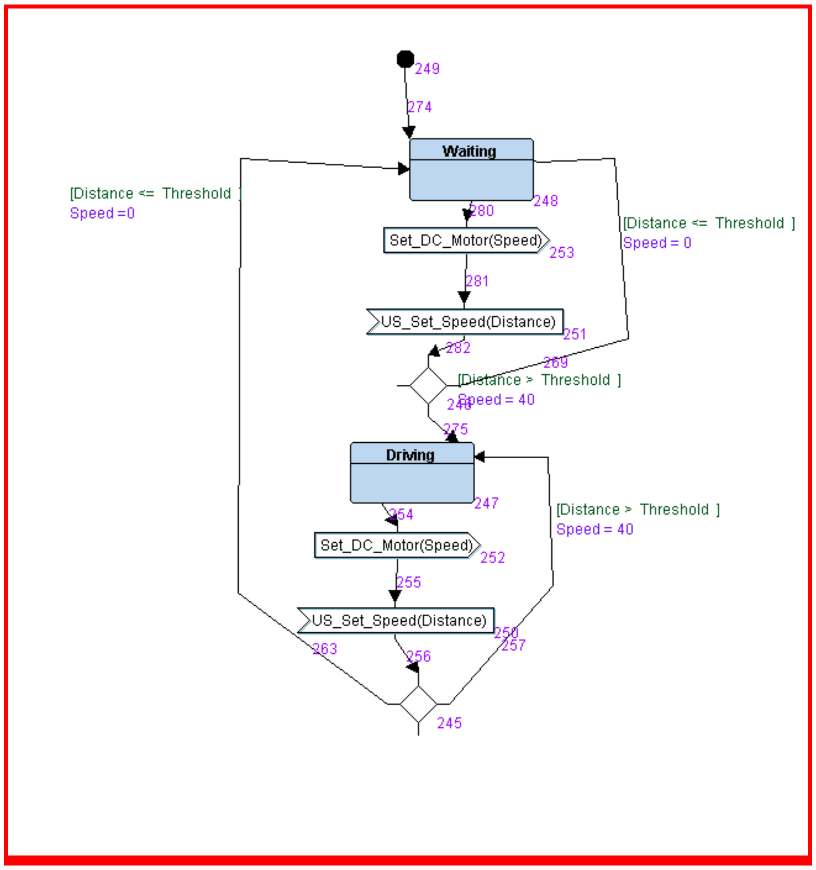
1-US_Sensor:



2-DC_Motor:

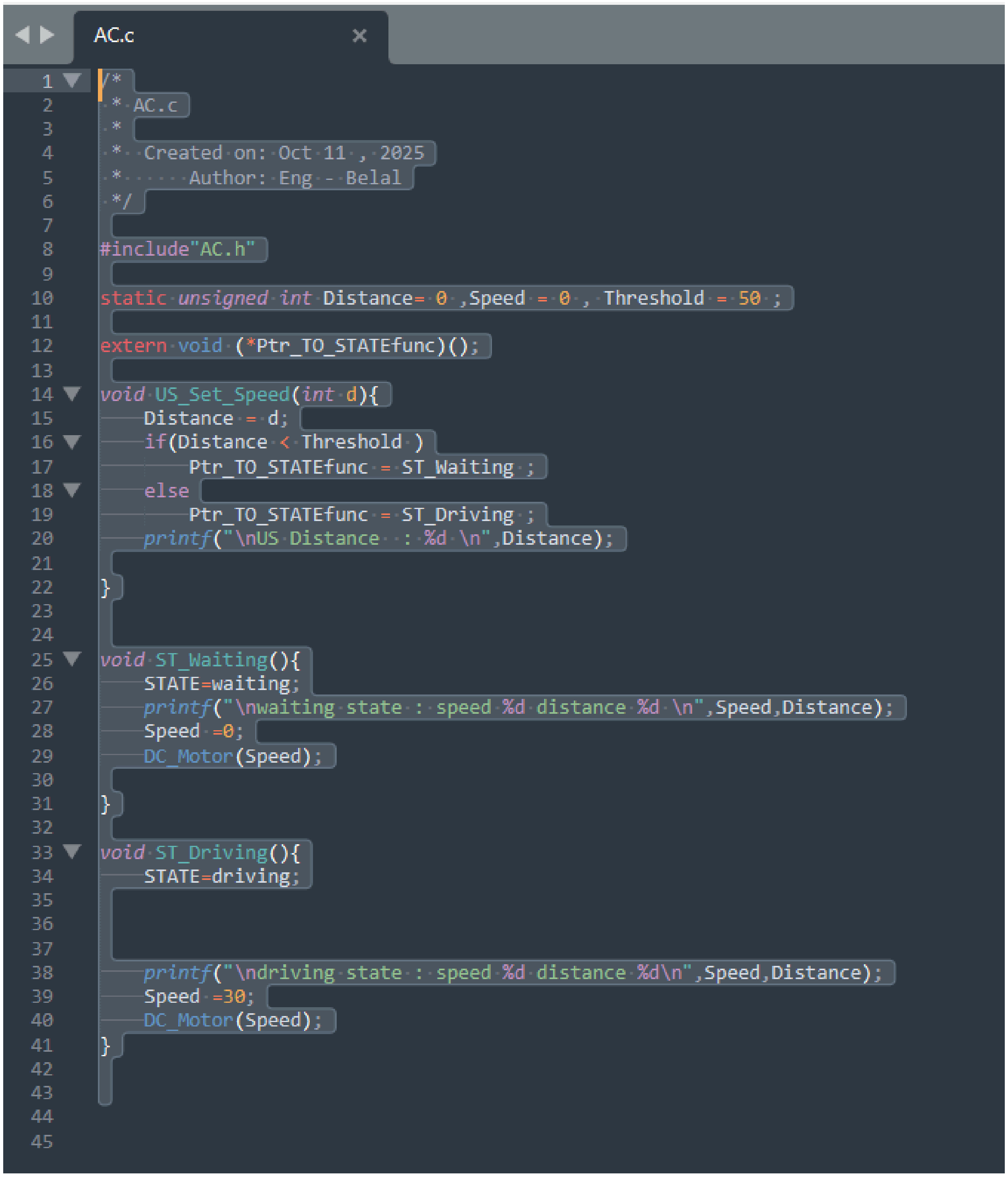

301
303
DC_Motor_Init
300
304
DC_Off
299
305
Set_DC_Motor(Speed)
298
306
DC_On
302
297

3-AC:


249
274
Waiting
248
280
Set_DC_Motor(Speed)
253
281
US_Set_Speed(Distance)
251
282
269
[Distance <= Threshold]
Speed =0
[Distance <= Threshold]
Speed = 0
[Distance > Threshold]
Speed = 40
246
275
Driving
247
254
Set_DC_Motor(Speed)
252
255
US_Set_Speed(Distance)
250
257
263
256
245
[Distance > Threshold]
Speed = 40

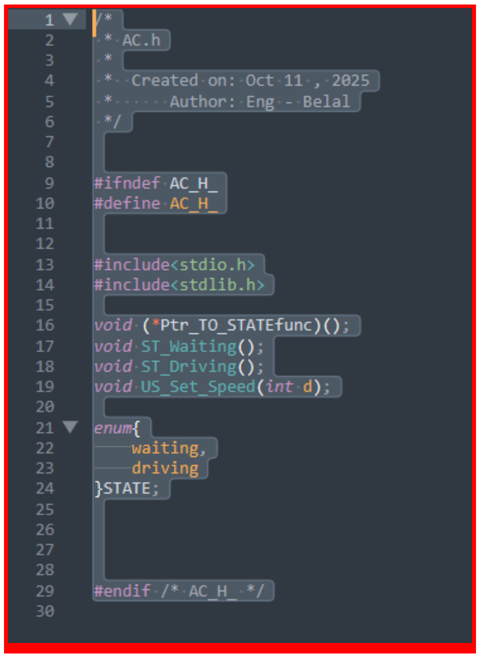
7-Coding:

AC.c:



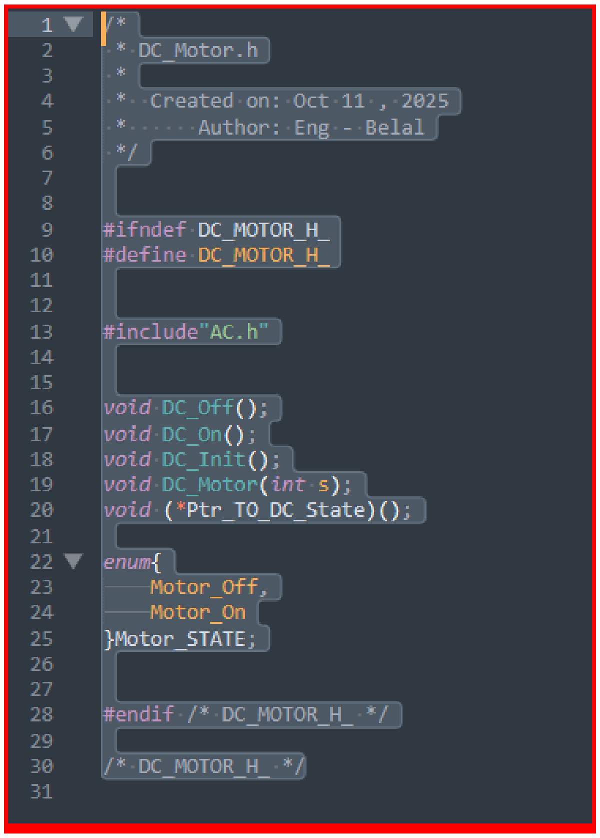
```c
/*
 * AC.c
 *
 *  Created on: Oct 11 , 2025
 *      Author: Eng - Belal
 */

#include"AC.h"

static unsigned int Distance= 0 ,Speed = 0 , Threshold = 50 ;

extern void (*Ptr_TO_STATEfunc)();

void US_Set_Speed(int d){
	Distance = d;
	if(Distance < Threshold )
		Ptr_TO_STATEfunc = ST_Waiting ;
	else
		Ptr_TO_STATEfunc = ST_Driving ;
	printf("\nUS Distance  : %d \n",Distance);

}


void ST_Waiting(){
	STATE=waiting;
	printf("\nwaiting state : speed %d distance %d \n",Speed,Distance);
	Speed =0;
	DC_Motor(Speed);

}

void ST_Driving(){
	STATE=driving;



	printf("\ndriving state : speed %d distance %d\n",Speed,Distance);
	Speed =30;
	DC_Motor(Speed);
}
```

AC.h:



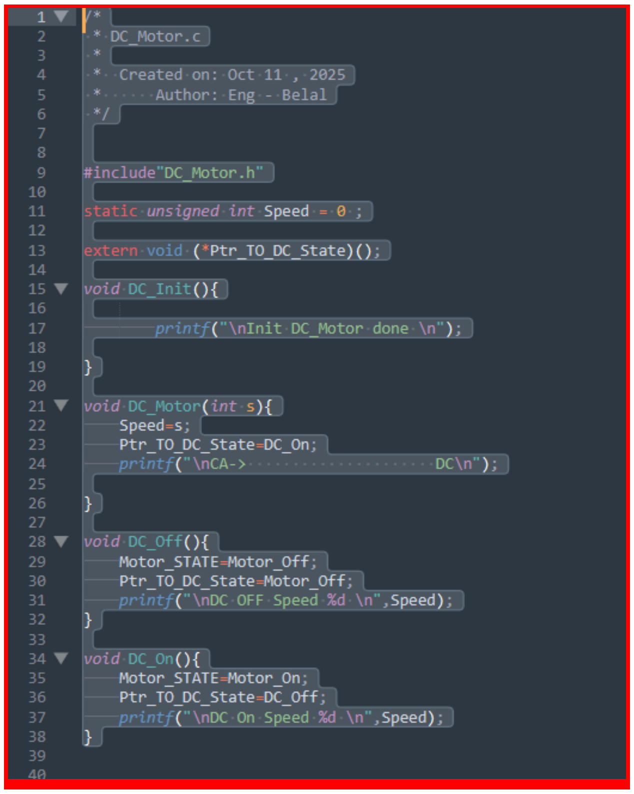
```c
/*
 * AC.h
 *
 *  Created on: Oct 11 , 2025
 *      Author: Eng - Belal
 */


#ifndef AC_H_
#define AC_H_


#include<stdio.h>
#include<stdlib.h>

void (*Ptr_TO_STATEfunc)();
void ST_Waiting();
void ST_Driving();
void US_Set_Speed(int d);

enum{
	waiting,
	driving
}STATE;




#endif /* AC_H_ */

```

DC_Motor.h:



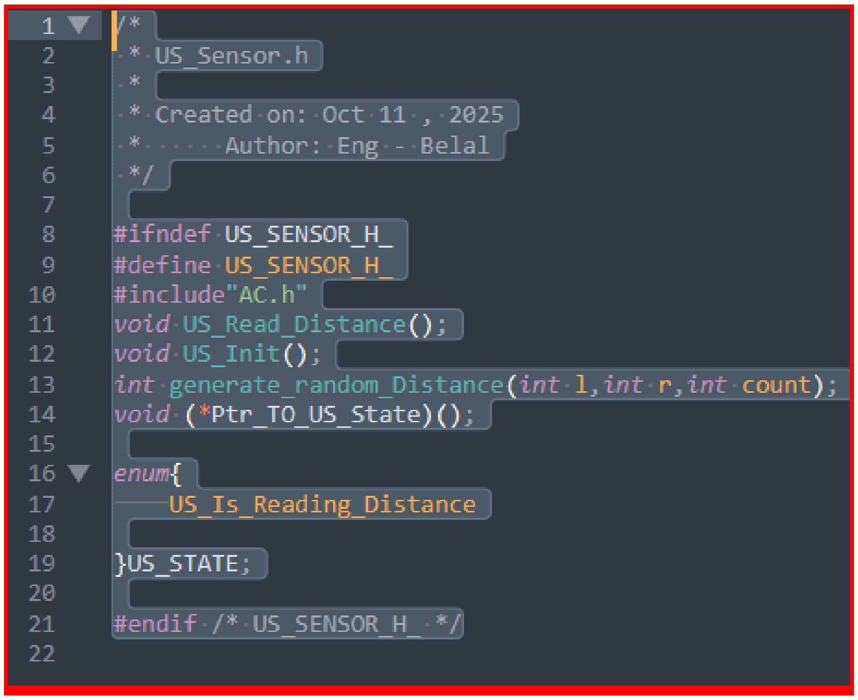
```c
/*
 * DC_Motor.h
 *
 *  Created on: Oct 11 , 2025
 *      Author: Eng - Belal
 */


#ifndef DC_MOTOR_H_
#define DC_MOTOR_H_


#include"AC.h"


void DC_Off();
void DC_On();
void DC_Init();
void DC_Motor(int s);
void (*Ptr_TO_DC_State)();

enum{
	Motor_Off,
	Motor_On
}Motor_STATE;


#endif /* DC_MOTOR_H_ */

/* DC_MOTOR_H_ */

```

DC_Motor.c:



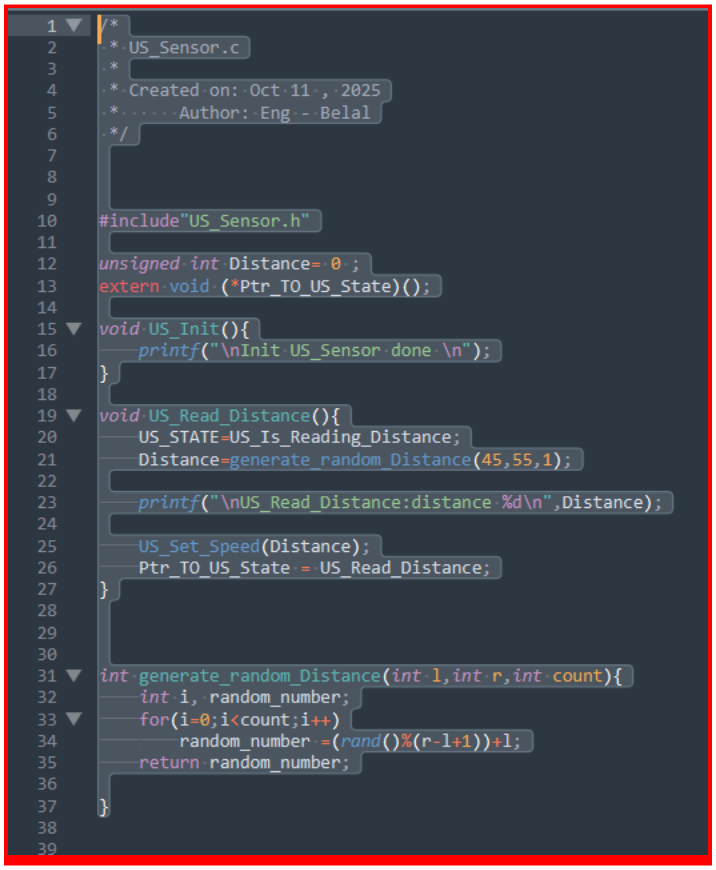
```c
/*
 * DC_Motor.c
 *
 *  Created on: Oct 11 , 2025
 *      Author: Eng - Belal
 */


#include"DC_Motor.h"

static unsigned int Speed = 0 ;

extern void (*Ptr_TO_DC_State)();

void DC_Init(){

        printf("\nInit DC_Motor done \n");

}

void DC_Motor(int s){
    Speed=s;
    Ptr_TO_DC_State=DC_On;
    printf("\nCA->                     DC\n");

}

void DC_Off(){
    Motor_STATE=Motor_Off;
    Ptr_TO_DC_State=Motor_Off;
    printf("\nDC OFF Speed %d \n",Speed);
}

void DC_On(){
    Motor_STATE=Motor_On;
    Ptr_TO_DC_State=DC_Off;
    printf("\nDC On Speed %d \n",Speed);
}


```

US_Sensor.h:



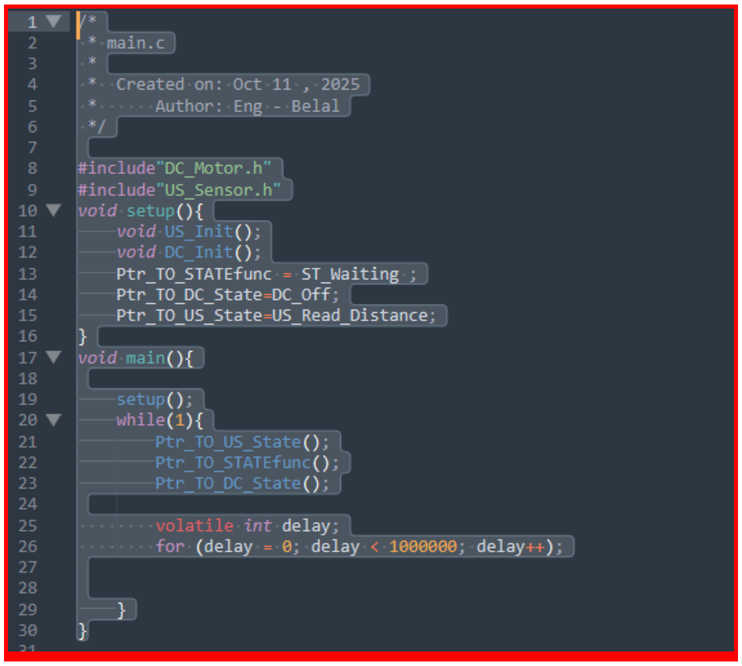
```c
/*
 * US_Sensor.h
 *
 * Created on: Oct 11 , 2025
 *      Author: Eng - Belal
 */

#ifndef US_SENSOR_H_
#define US_SENSOR_H_
#include"AC.h"
void US_Read_Distance();
void US_Init();
int generate_random_Distance(int l,int r,int count);
void (*Ptr_TO_US_State)();

enum{
    US_Is_Reading_Distance

}US_STATE;

#endif /* US_SENSOR_H_ */

```

US_Sensor.c:



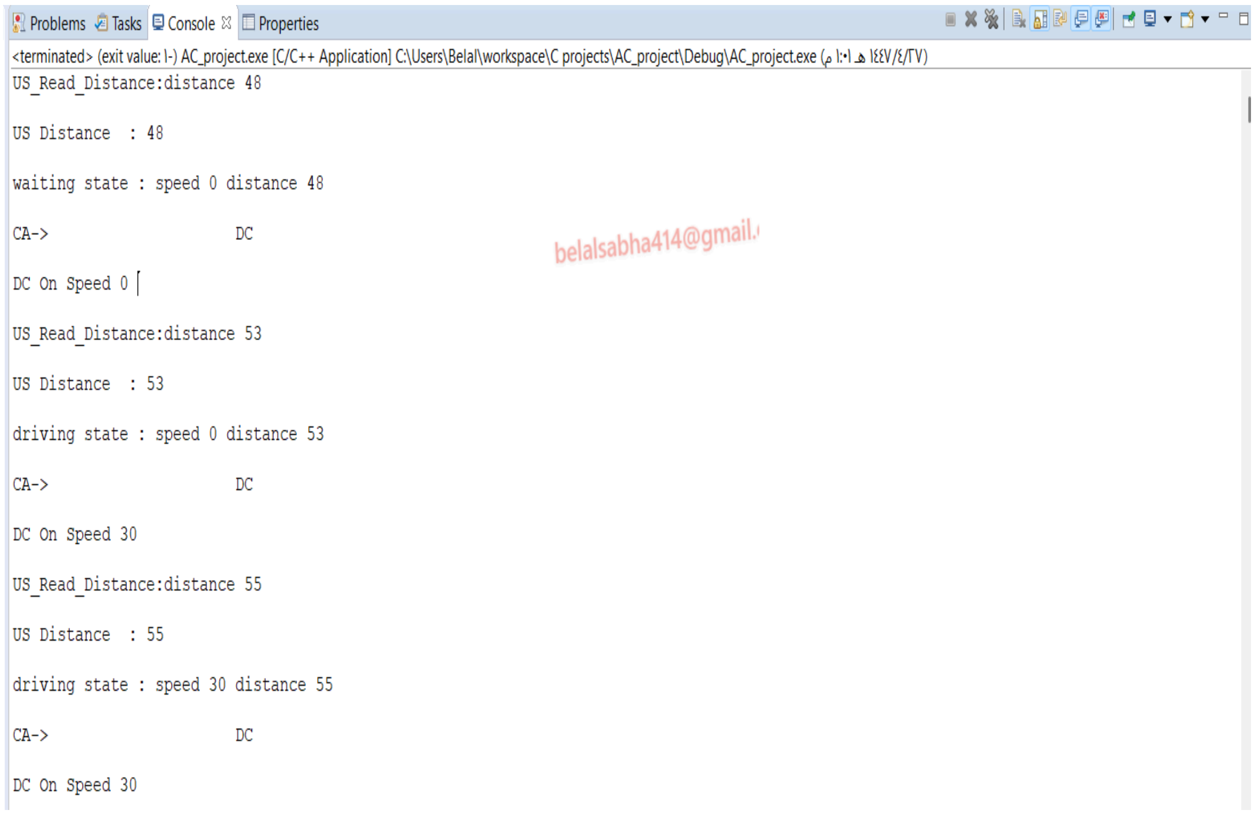
```c
/*
 * US_Sensor.c
 *
 * Created on: Oct 11 , 2025
 *      Author: Eng - Belal
 */



#include"US_Sensor.h"

unsigned int Distance= 0 ;
extern void (*Ptr_TO_US_State)();

void US_Init(){
	printf("\nInit US_Sensor done \n");
}

void US_Read_Distance(){
	US_STATE=US_Is_Reading_Distance;
	Distance=generate_random_Distance(45,55,1);

	printf("\nUS_Read_Distance:distance %d\n",Distance);

	US_Set_Speed(Distance);
	Ptr_TO_US_State = US_Read_Distance;
}



int generate_random_Distance(int l,int r,int count){
	int i, random_number;
	for(i=0;i<count;i++)
		random_number =(rand()%(r-l+1))+l;
	return random_number;

}
```

Main.c:



```c
/*
 * main.c
 *
 *  Created on: Oct 11 , 2025
 *      Author: Eng - Belal
 */

#include"DC_Motor.h"
#include"US_Sensor.h"
void setup(){
	void US_Init();
	void DC_Init();
	Ptr_TO_STATEfunc = ST_Waiting ;
	Ptr_TO_DC_State=DC_Off;
	Ptr_TO_US_State=US_Read_Distance;
}
void main(){

	setup();
	while(1){
		Ptr_TO_US_State();
		Ptr_TO_STATEfunc();
		Ptr_TO_DC_State();

		volatile int delay;
		for (delay = 0; delay < 1000000; delay++);


	}
}
```

<u>Output:</u>



```
<terminated> (exit value: ١-) AC_project.exe [C/C++ Application] C:\Users\Belal\workspace\C projects\AC_project\Debug\AC_project.exe (١٤٤٧/٤/٢٧ هـ ١:٠١ م)
US_Read_Distance:distance 48

US Distance  : 48

waiting state : speed 0 distance 48

CA->                    DC

DC On Speed 0

US_Read_Distance:distance 53

US Distance  : 53

driving state : speed 0 distance 53

CA->                    DC

DC On Speed 30

US_Read_Distance:distance 55

US Distance  : 55

driving state : speed 30 distance 55

CA->                    DC

DC On Speed 30
```