ULTRASONIC OBSTACLE-AVOIDING ROBOT

PROJECT

By Mohammed Hassan

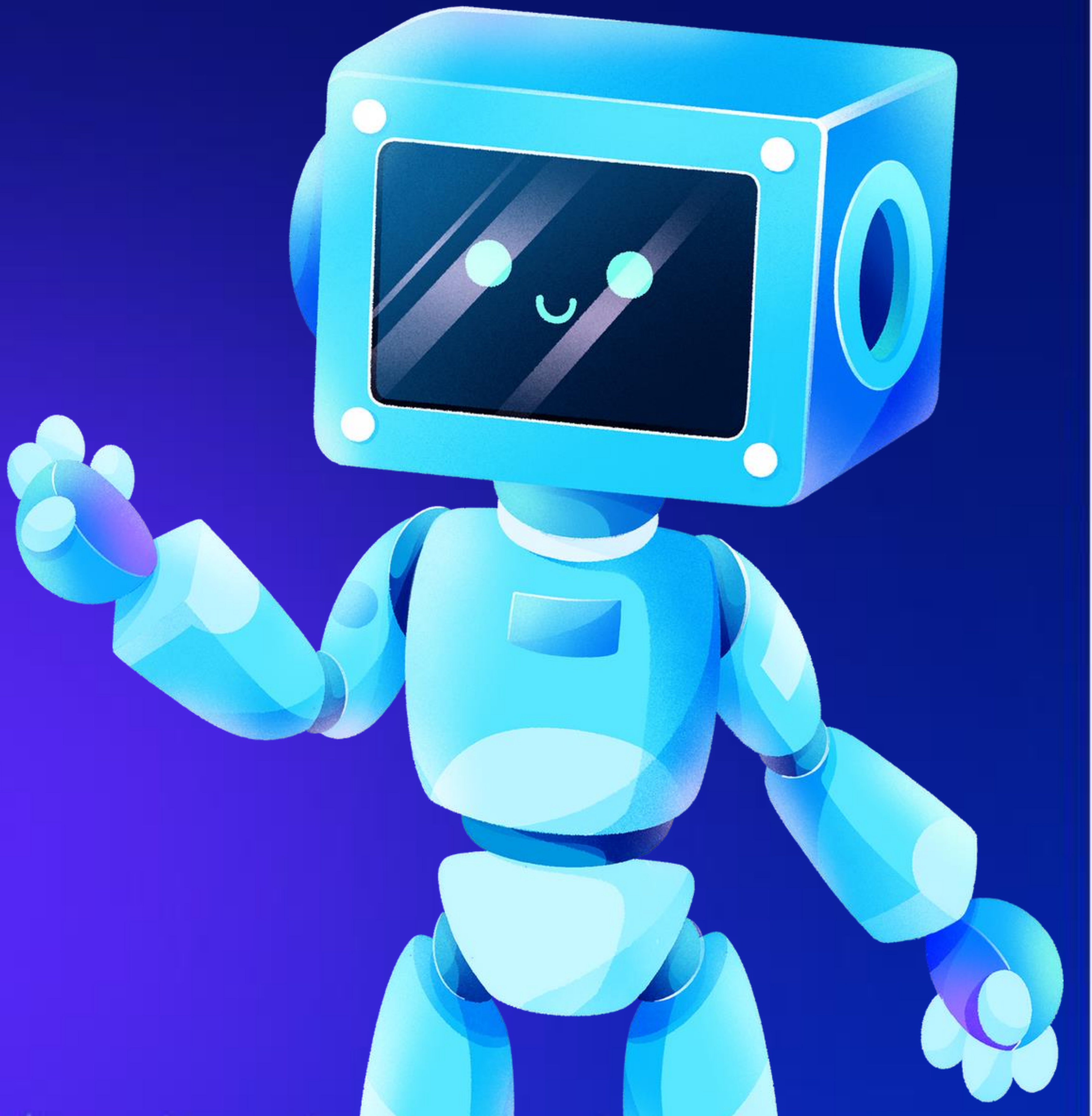
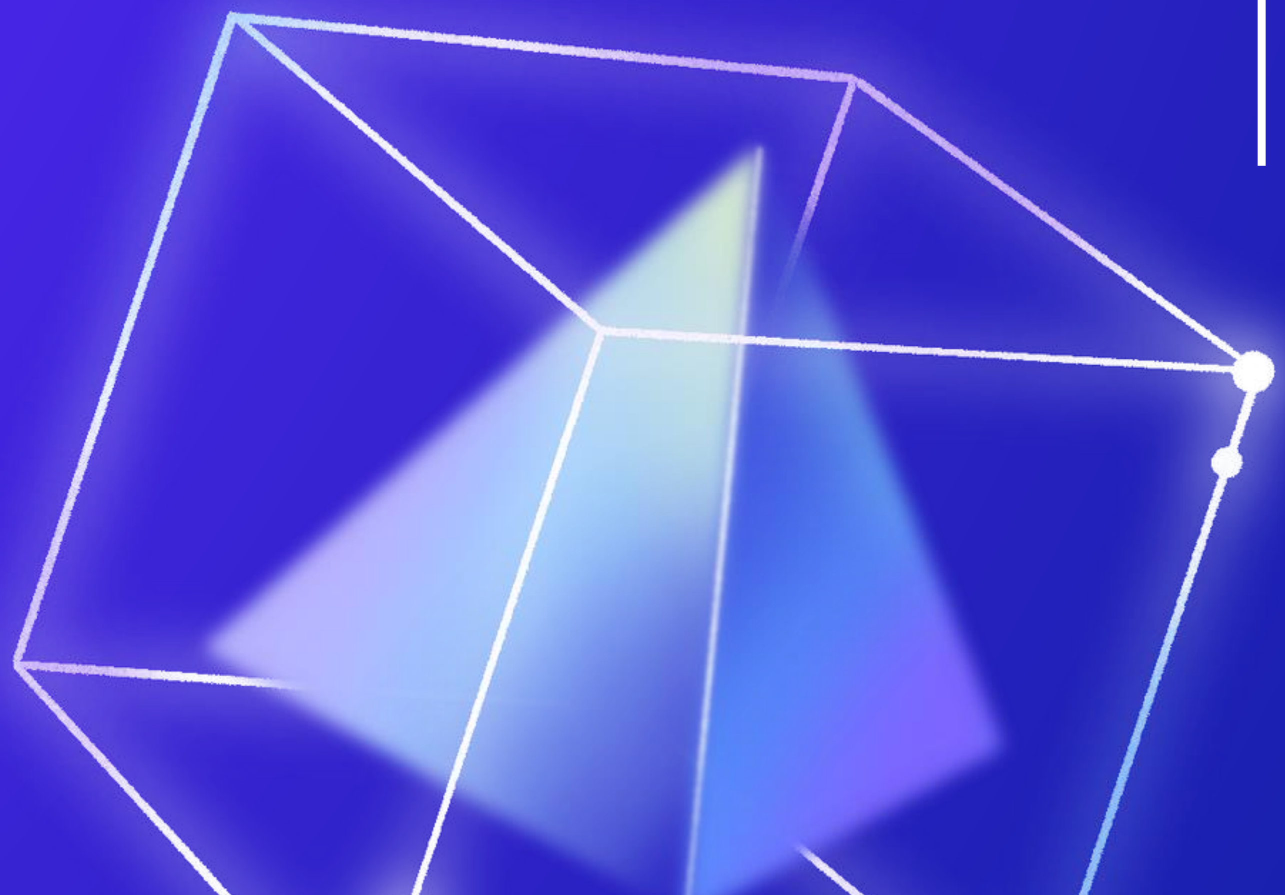
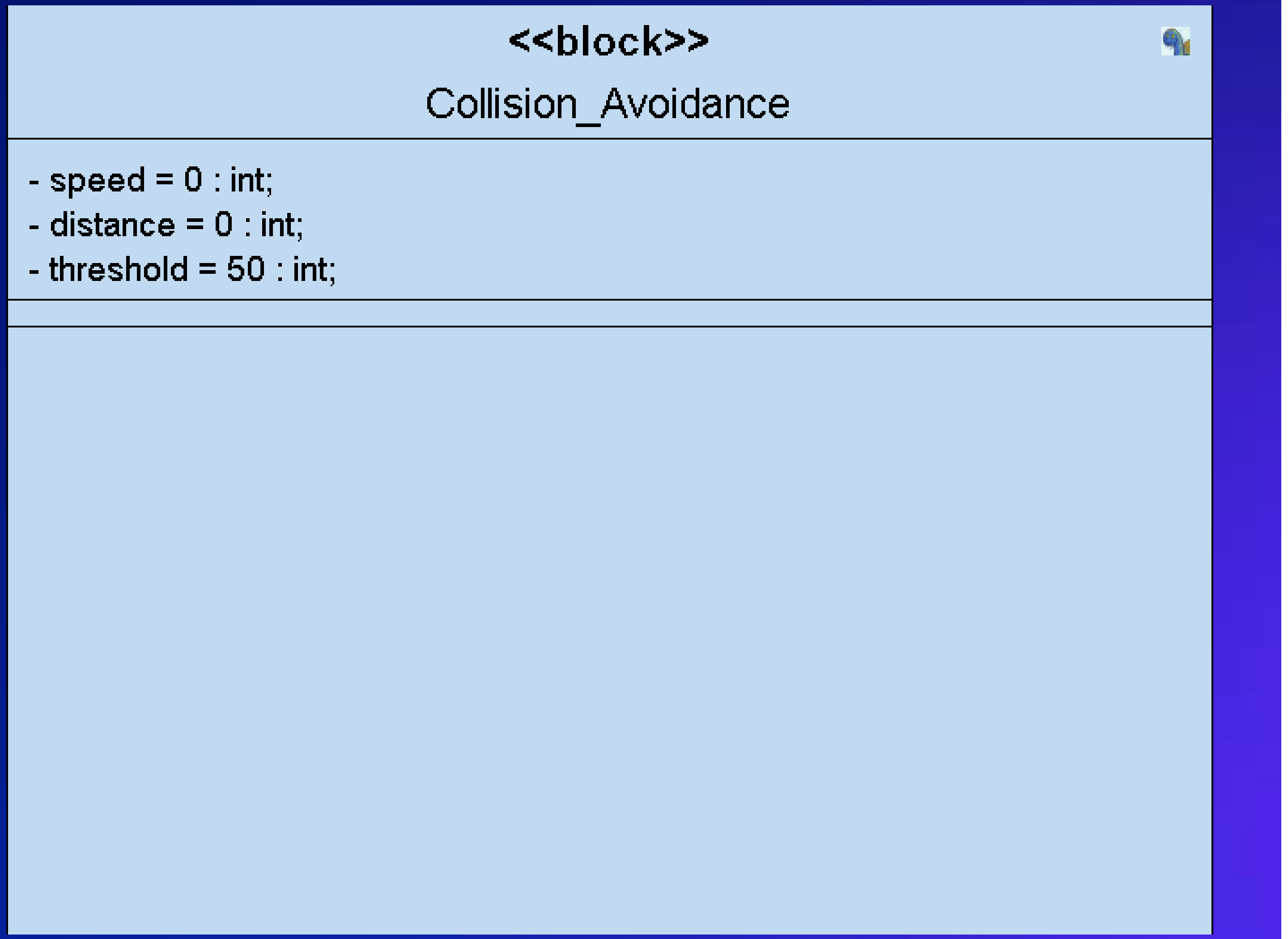


TABLE OF CONTENTS



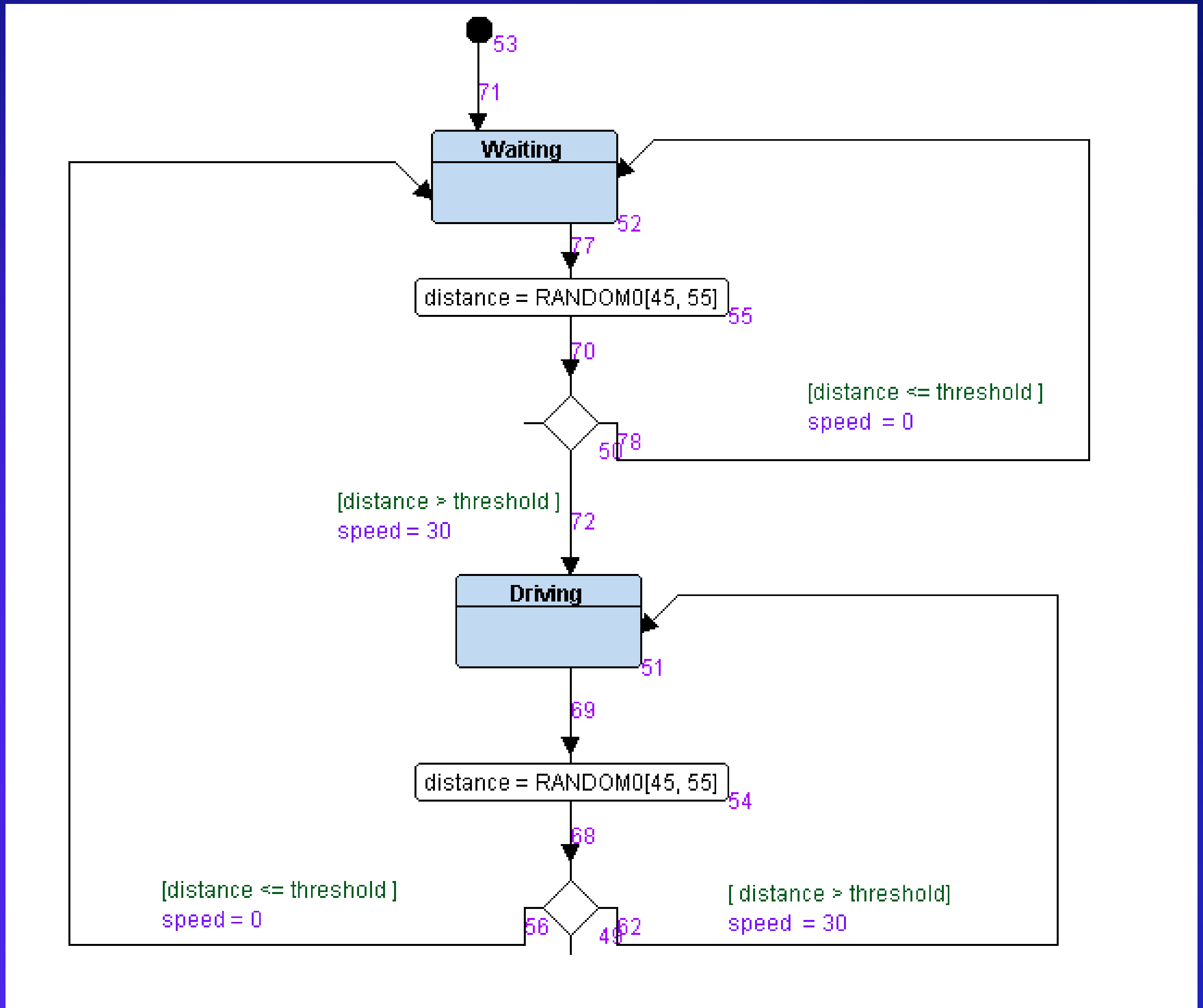
State Diagram

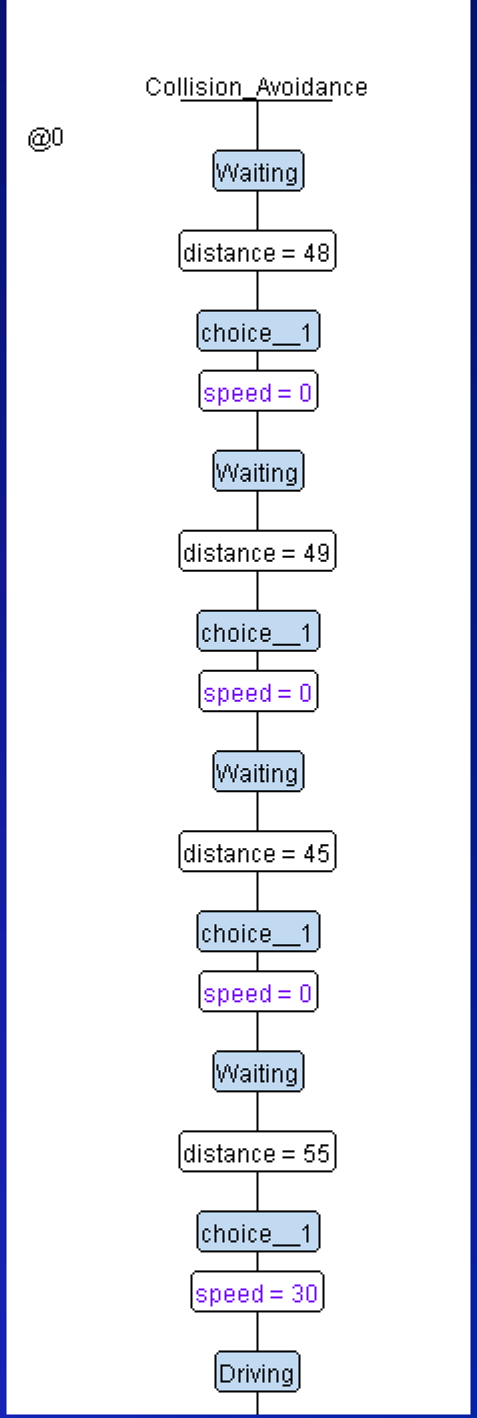
Collison Avoidance



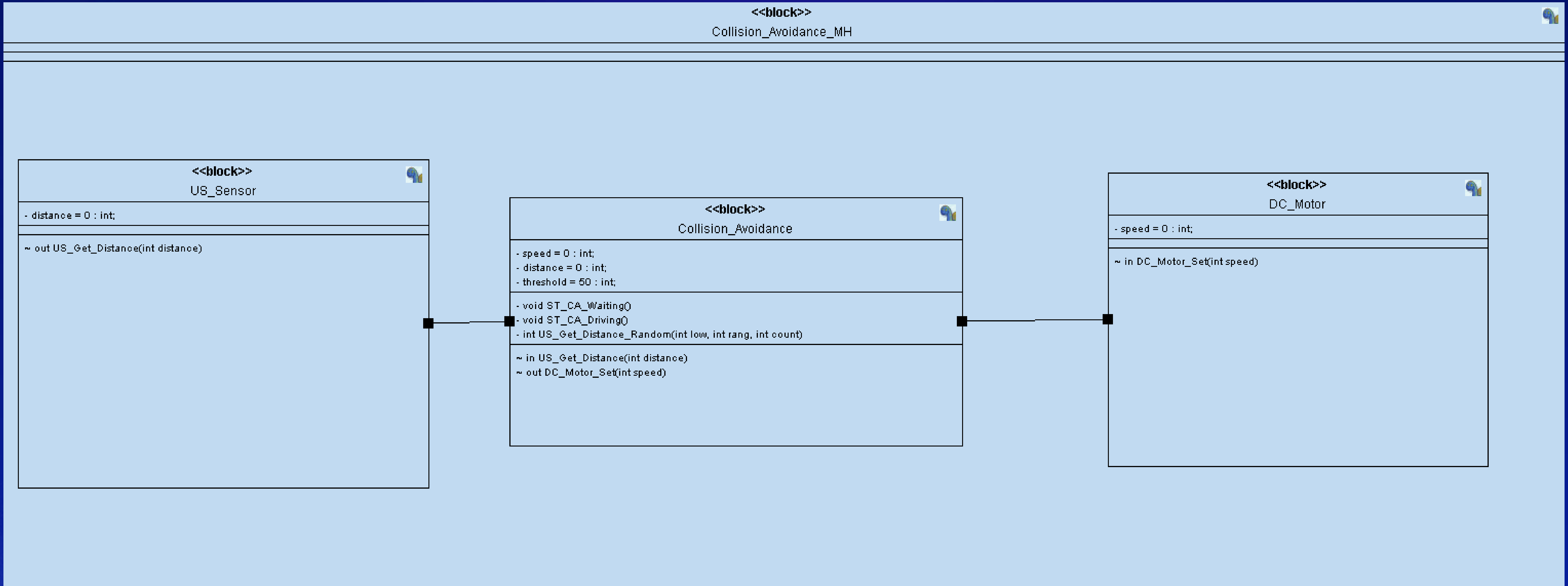
<<block>>
Collision_Avoidance

- speed = 0 : int;
- distance = 0 : int;
- threshold = 50 : int;





Modules level



<<block>>
Collision_Avoidance_MH

<<block>>
US_Sensor

- distance = 0 : int;

~ out US_Get_Distance(int distance)

<<block>>
Collision_Avoidance

- speed = 0 : int;
- distance = 0 : int;
- threshold = 50 : int;

- void ST_CA_Waiting()
- void ST_CA_Driving()
- int US_Get_Distance_Random(int low, int rang, int count)

~ in US_Get_Distance(int distance)
~ out DC_Motor_Set(int speed)

<<block>>
DC_Motor

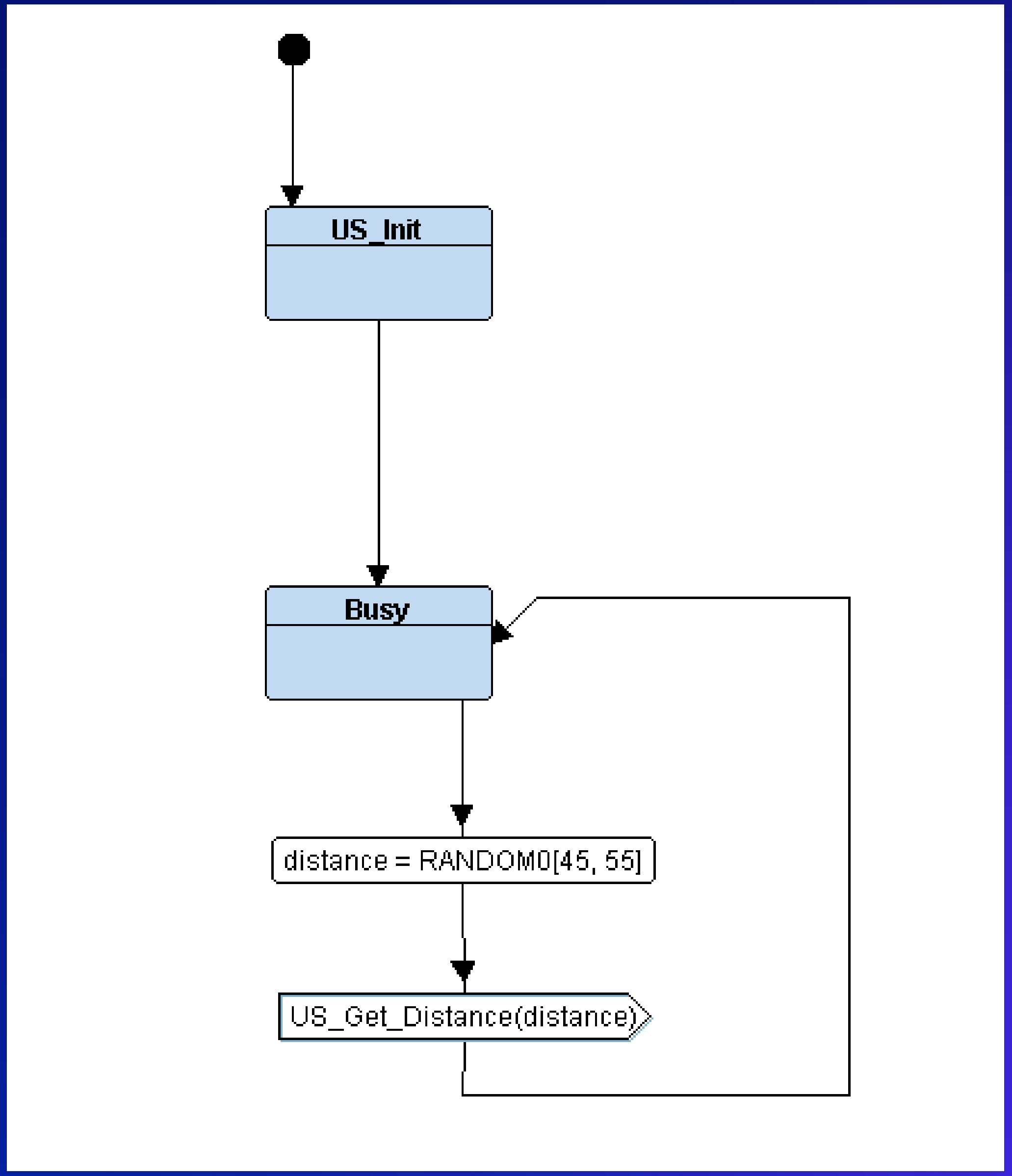
- speed = 0 : int;

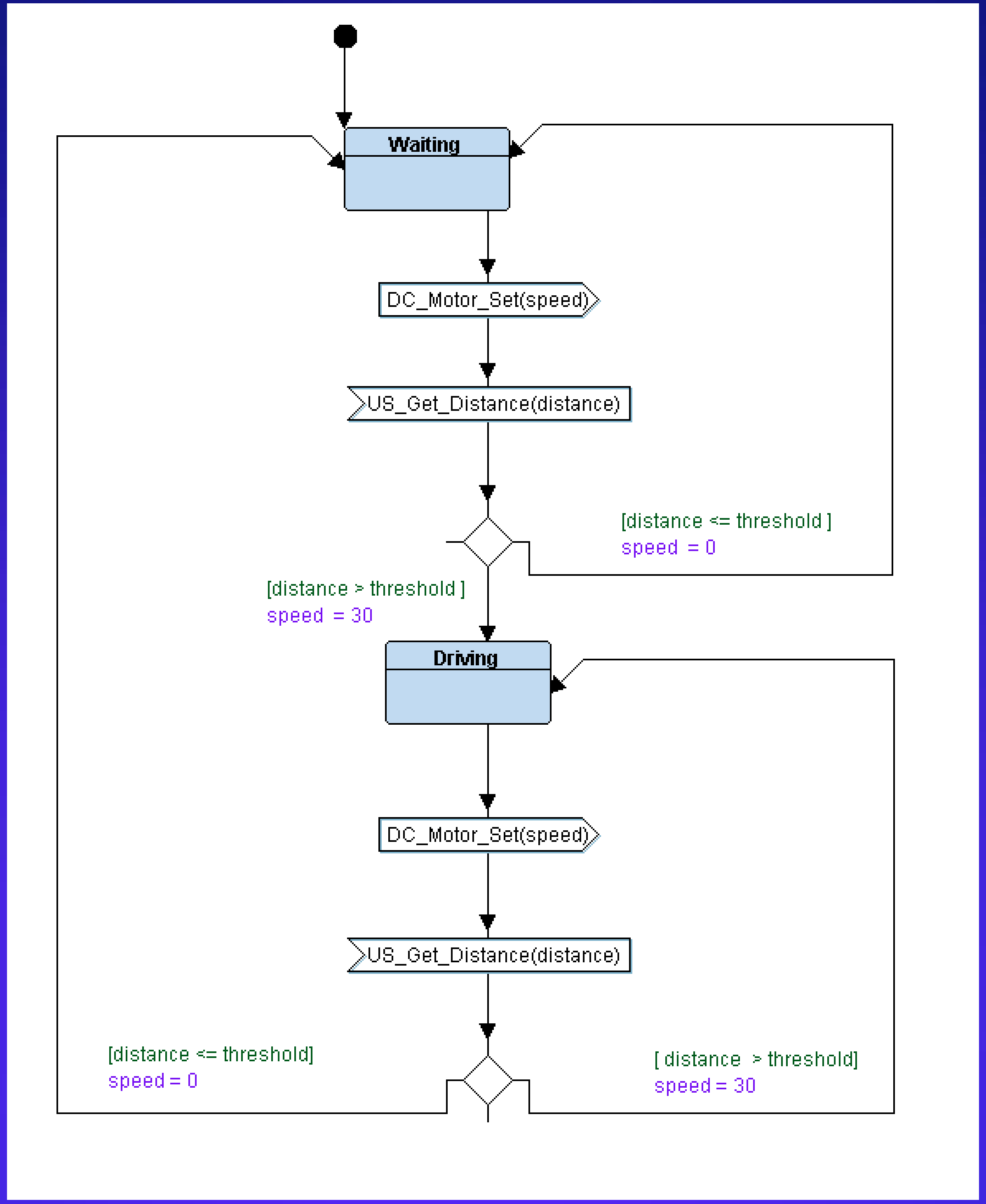
~ in DC_Motor_Set(int speed)

Logical Design

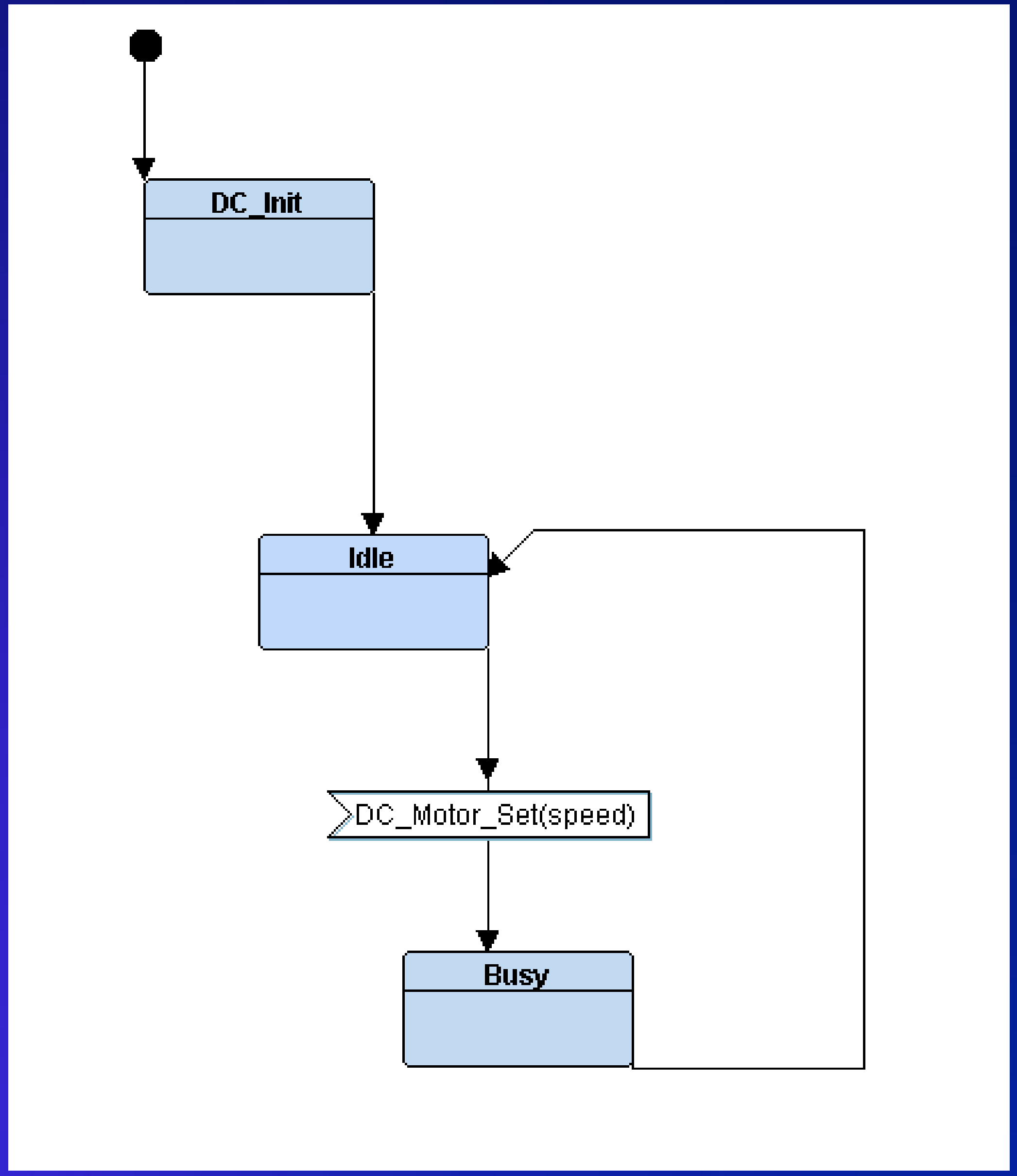
Collision Avoidance

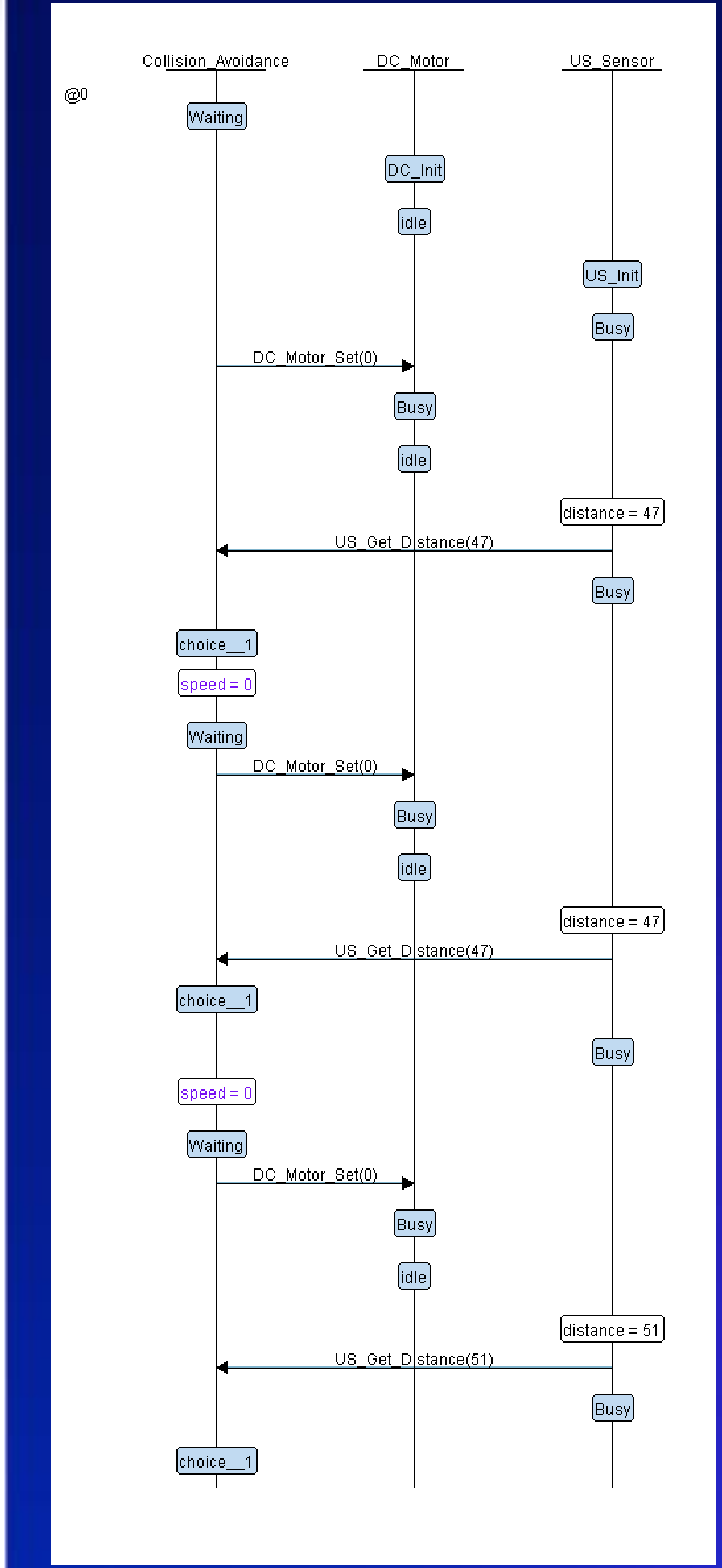
US



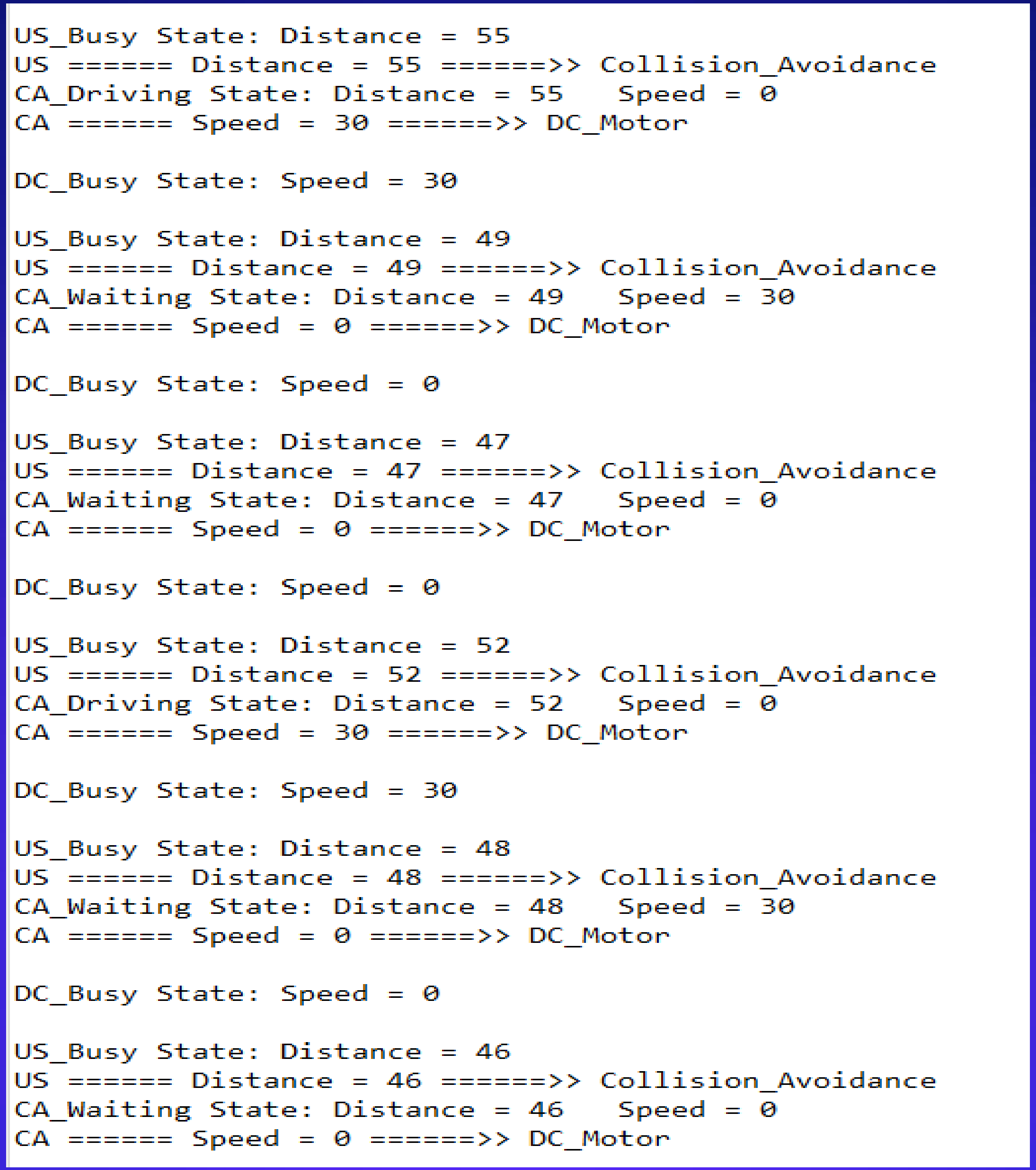


DC Motor

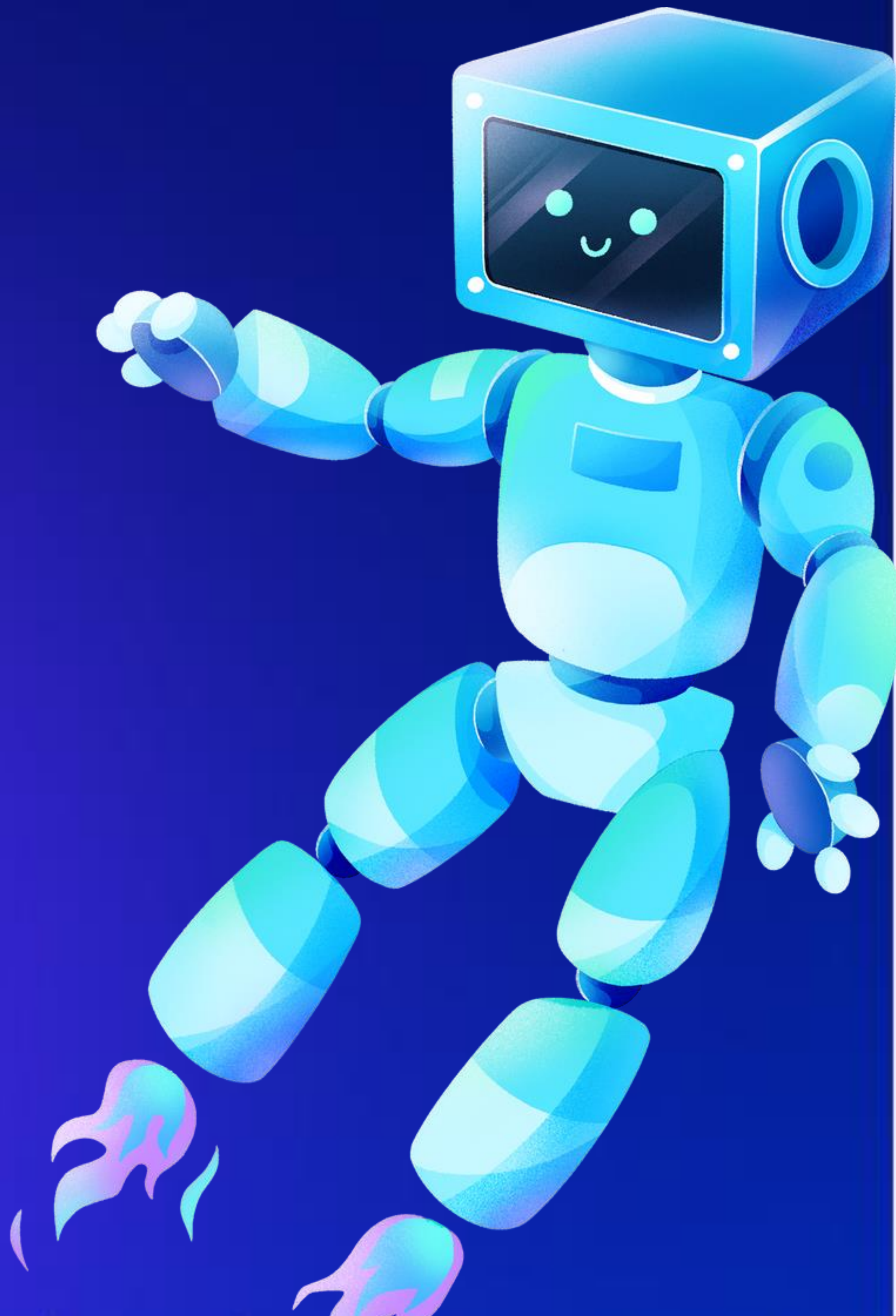




SW Logical verification

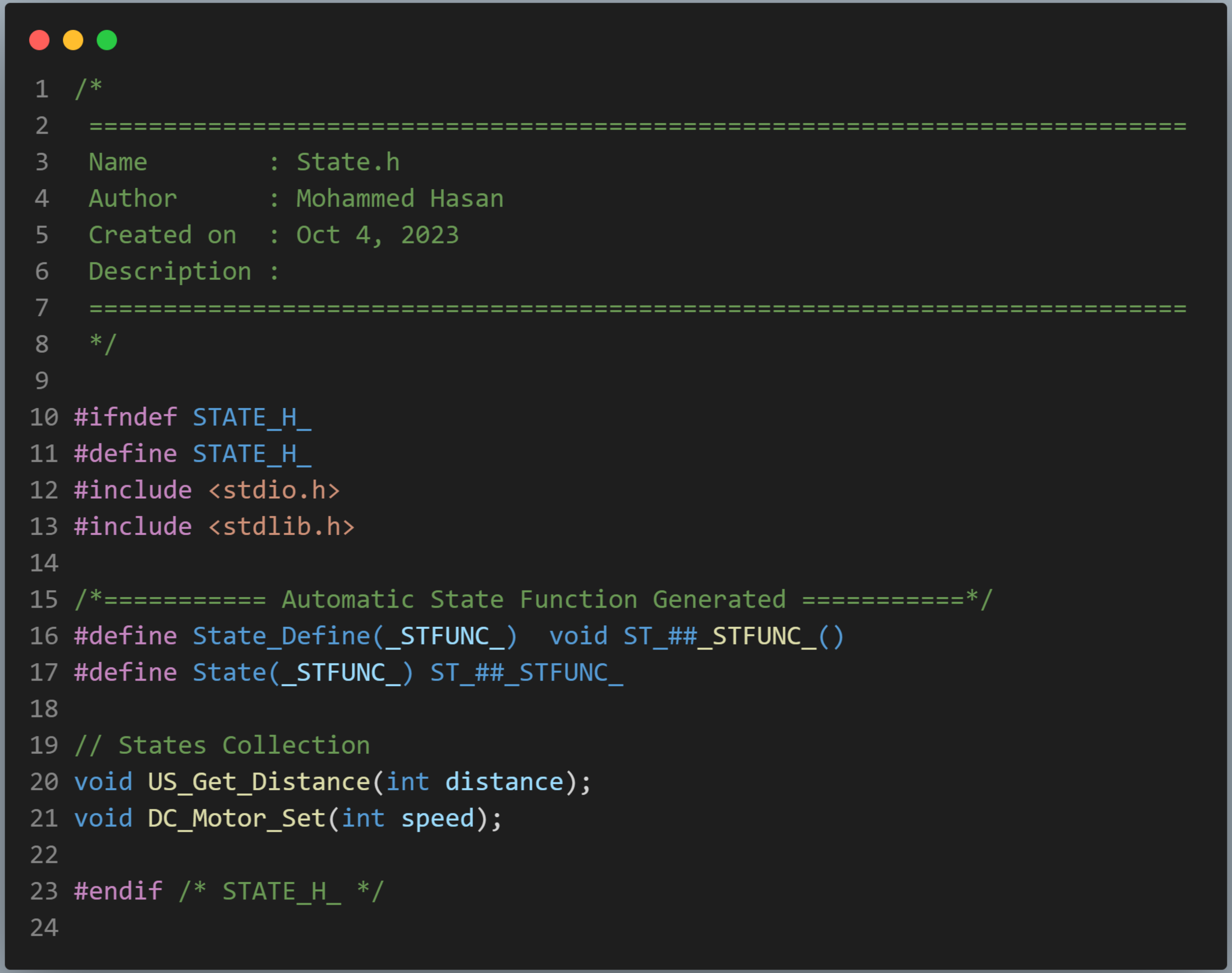


```
US_Busy State: Distance = 55
US ====== Distance = 55 ======>> Collision_Avoidance
CA_Driving State: Distance = 55    Speed = 0
CA ====== Speed = 30 ======>> DC_Motor

DC_Busy State: Speed = 30

US_Busy State: Distance = 49
US ====== Distance = 49 ======>> Collision_Avoidance
CA_Waiting State: Distance = 49    Speed = 30
CA ====== Speed = 0 ======>> DC_Motor

DC_Busy State: Speed = 0

US_Busy State: Distance = 47
US ====== Distance = 47 ======>> Collision_Avoidance
CA_Waiting State: Distance = 47    Speed = 0
CA ====== Speed = 0 ======>> DC_Motor

DC_Busy State: Speed = 0

US_Busy State: Distance = 52
US ====== Distance = 52 ======>> Collision_Avoidance
CA_Driving State: Distance = 52    Speed = 0
CA ====== Speed = 30 ======>> DC_Motor

DC_Busy State: Speed = 30

US_Busy State: Distance = 48
US ====== Distance = 48 ======>> Collision_Avoidance
CA_Waiting State: Distance = 48    Speed = 30
CA ====== Speed = 0 ======>> DC_Motor

DC_Busy State: Speed = 0

US_Busy State: Distance = 46
US ====== Distance = 46 ======>> Collision_Avoidance
CA_Waiting State: Distance = 46    Speed = 0
CA ====== Speed = 0 ======>> DC_Motor
```



C Implementation

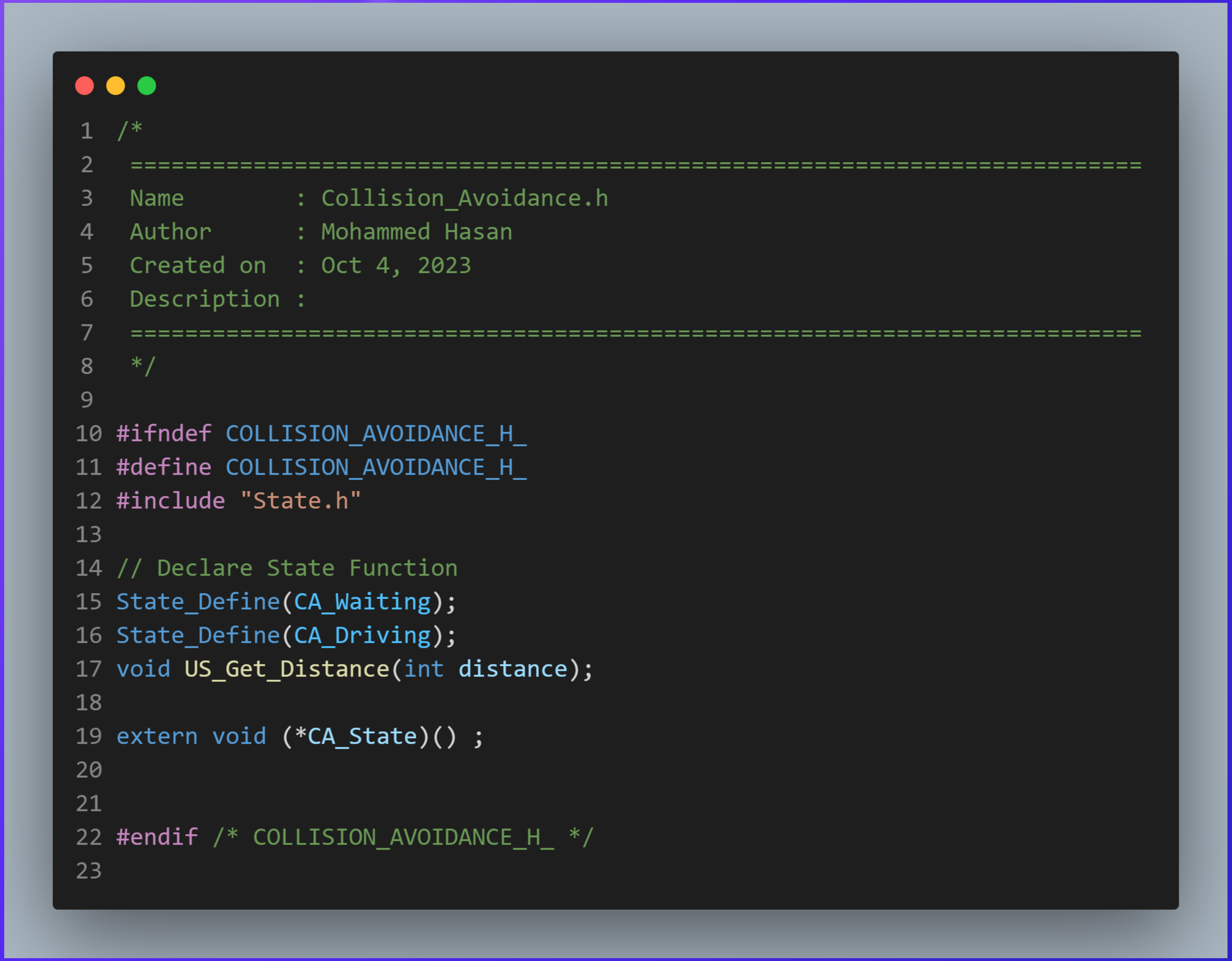
State.h



```c
/*
 ==========================================================================
 Name        : State.h
 Author      : Mohammed Hasan
 Created on  : Oct 4, 2023
 Description :
 ==========================================================================
 */

#ifndef STATE_H_
#define STATE_H_
#include <stdio.h>
#include <stdlib.h>

/*=========== Automatic State Function Generated ===========*/
#define State_Define(_STFUNC_)  void ST_##_STFUNC_()
#define State(_STFUNC_) ST_##_STFUNC_

// States Collection
void US_Get_Distance(int distance);
void DC_Motor_Set(int speed);

#endif /* STATE_H_ */

```

C Implementation

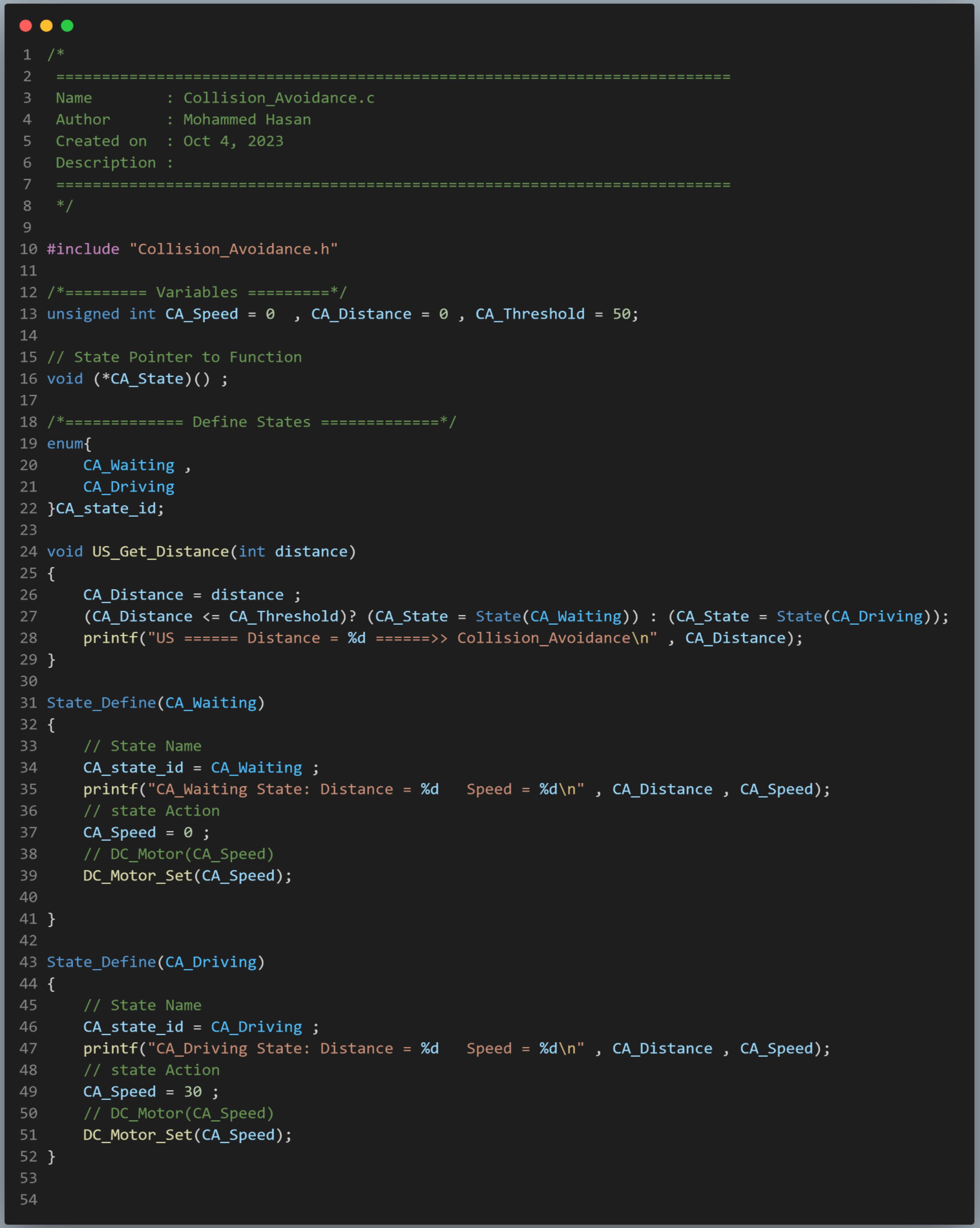
CA.h



```c
/*
 ==========================================================================
 Name        : Collision_Avoidance.h
 Author      : Mohammed Hasan
 Created on  : Oct 4, 2023
 Description :
 ==========================================================================
 */

#ifndef COLLISION_AVOIDANCE_H_
#define COLLISION_AVOIDANCE_H_
#include "State.h"

// Declare State Function
State_Define(CA_Waiting);
State_Define(CA_Driving);
void US_Get_Distance(int distance);

extern void (*CA_State)() ;


#endif /* COLLISION_AVOIDANCE_H_ */

```

C Implementation

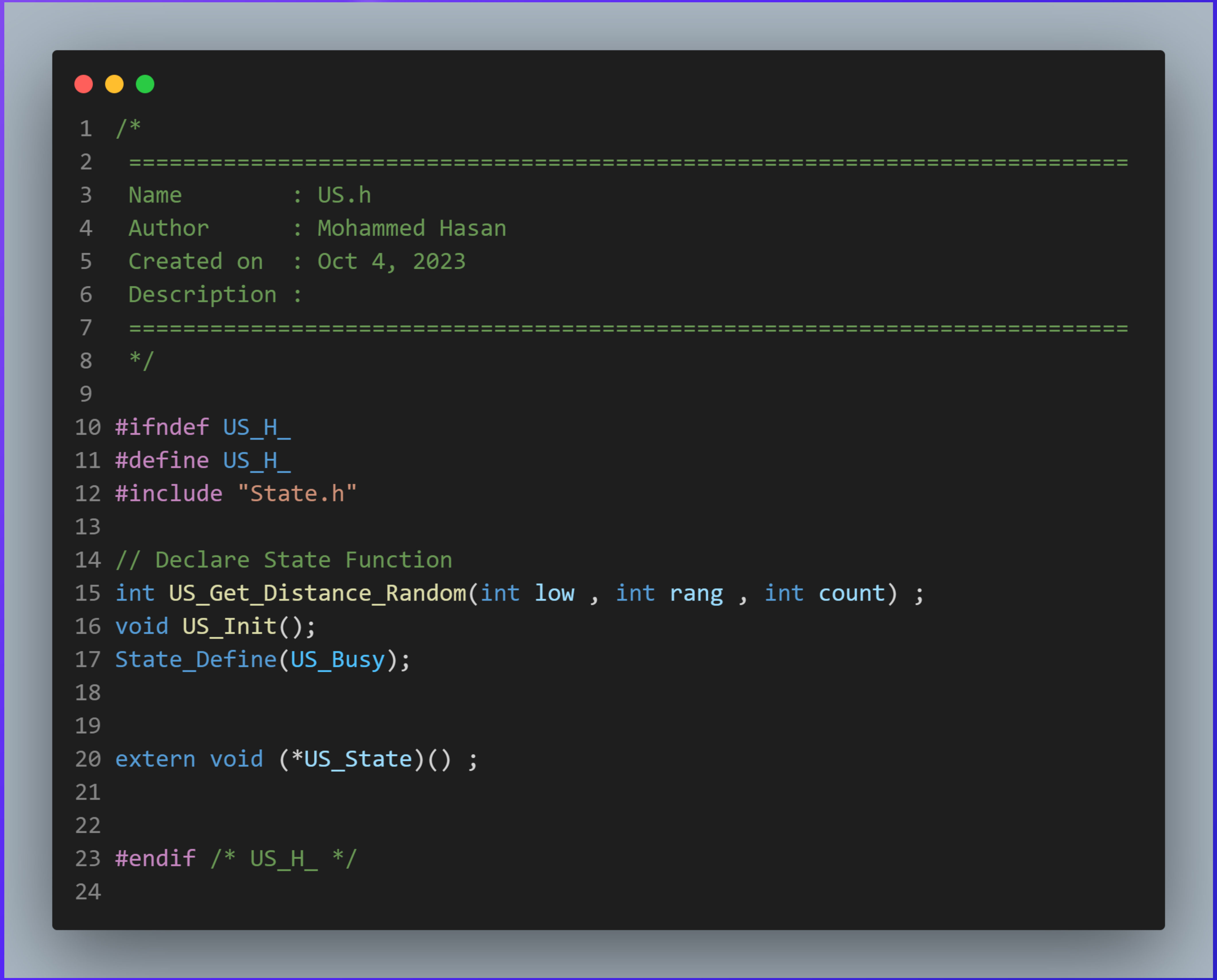
CA.c



```c
/*
 ==========================================================================
 Name        : Collision_Avoidance.c
 Author      : Mohammed Hasan
 Created on  : Oct 4, 2023
 Description :
 ==========================================================================
 */

#include "Collision_Avoidance.h"

/*========= Variables =========*/
unsigned int CA_Speed = 0  , CA_Distance = 0 , CA_Threshold = 50;

// State Pointer to Function
void (*CA_State)() ;

/*============= Define States =============*/
enum{
    CA_Waiting ,
    CA_Driving
}CA_state_id;

void US_Get_Distance(int distance)
{
    CA_Distance = distance ;
    (CA_Distance <= CA_Threshold)? (CA_State = State(CA_Waiting)) : (CA_State = State(CA_Driving));
    printf("US ====== Distance = %d ======>> Collision_Avoidance\n" , CA_Distance);
}

State_Define(CA_Waiting)
{
    // State Name
    CA_state_id = CA_Waiting ;
    printf("CA_Waiting State: Distance = %d   Speed = %d\n" , CA_Distance , CA_Speed);
    // state Action
    CA_Speed = 0 ;
    // DC_Motor(CA_Speed)
    DC_Motor_Set(CA_Speed);

}

State_Define(CA_Driving)
{
    // State Name
    CA_state_id = CA_Driving ;
    printf("CA_Driving State: Distance = %d   Speed = %d\n" , CA_Distance , CA_Speed);
    // state Action
    CA_Speed = 30 ;
    // DC_Motor(CA_Speed)
    DC_Motor_Set(CA_Speed);
}


```

C Implementation

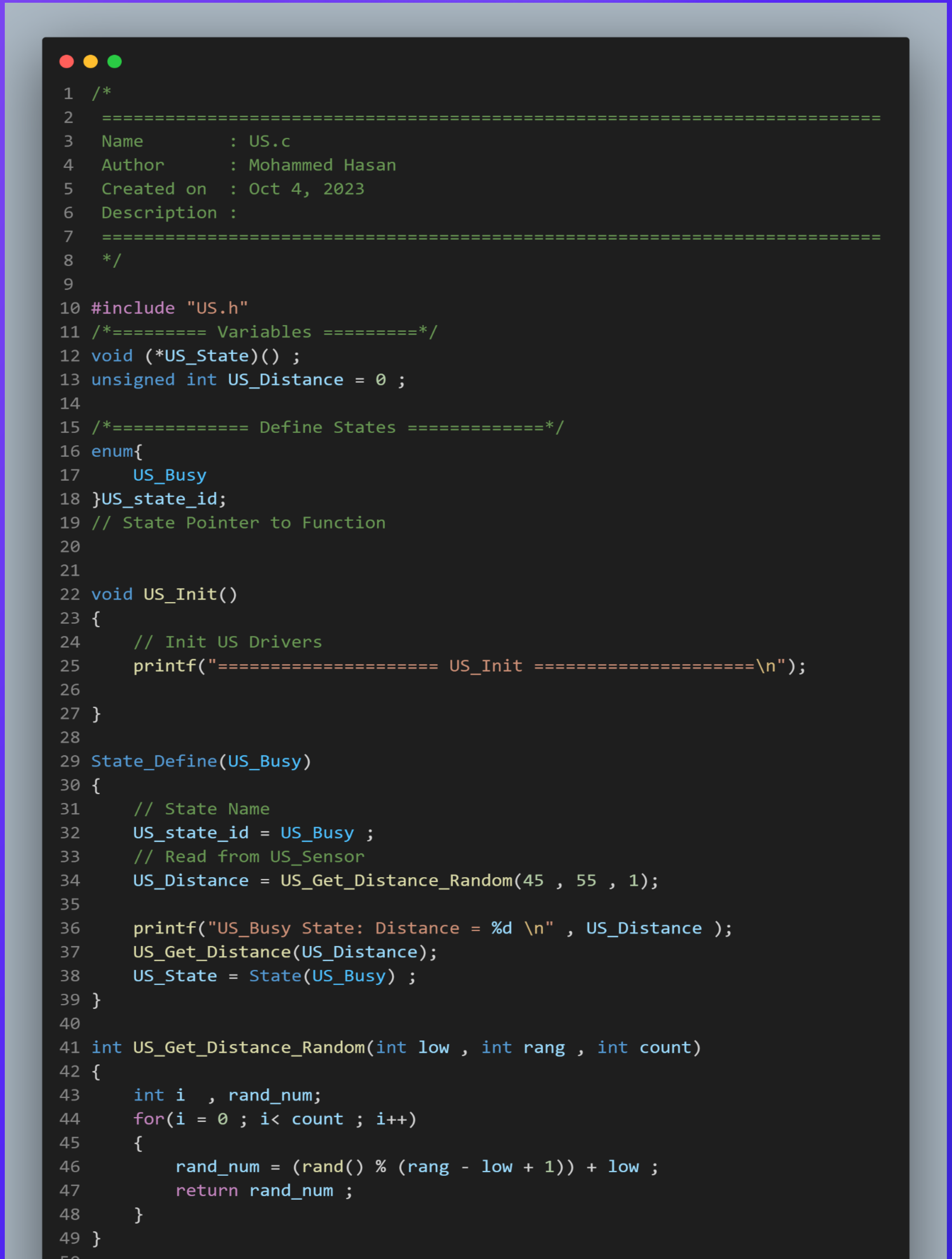
US.h



```c
/*
 ==========================================================================
 Name        : US.h
 Author      : Mohammed Hasan
 Created on  : Oct 4, 2023
 Description :
 ==========================================================================
 */

#ifndef US_H_
#define US_H_
#include "State.h"

// Declare State Function
int US_Get_Distance_Random(int low , int rang , int count) ;
void US_Init();
State_Define(US_Busy);


extern void (*US_State)() ;


#endif /* US_H_ */

```

C Implementation

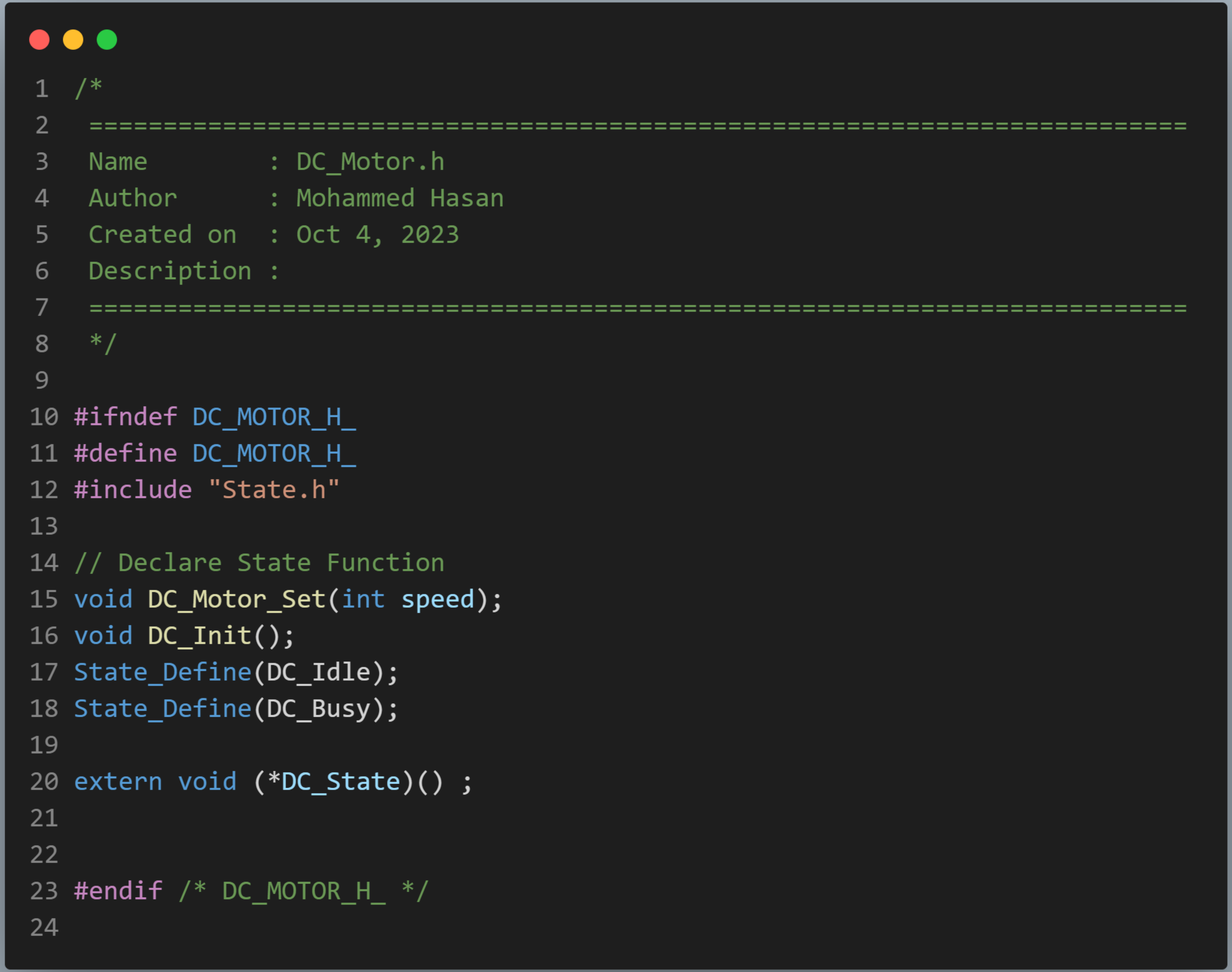
US.c



```c
/*
 ===========================================================================
 Name        : US.c
 Author      : Mohammed Hasan
 Created on  : Oct 4, 2023
 Description :
 ===========================================================================
 */

#include "US.h"
/*========= Variables =========*/
void (*US_State)() ;
unsigned int US_Distance = 0 ;

/*============= Define States =============*/
enum{
    US_Busy
}US_state_id;
// State Pointer to Function


void US_Init()
{
    // Init US Drivers
    printf("===================== US_Init =====================\n");

}

State_Define(US_Busy)
{
    // State Name
    US_state_id = US_Busy ;
    // Read from US_Sensor
    US_Distance = US_Get_Distance_Random(45 , 55 , 1);

    printf("US_Busy State: Distance = %d \n" , US_Distance );
    US_Get_Distance(US_Distance);
    US_State = State(US_Busy) ;
}

int US_Get_Distance_Random(int low , int rang , int count)
{
    int i  , rand_num;
    for(i = 0 ; i< count ; i++)
    {
        rand_num = (rand() % (rang - low + 1)) + low ;
        return rand_num ;
    }
}
```

C Implementation

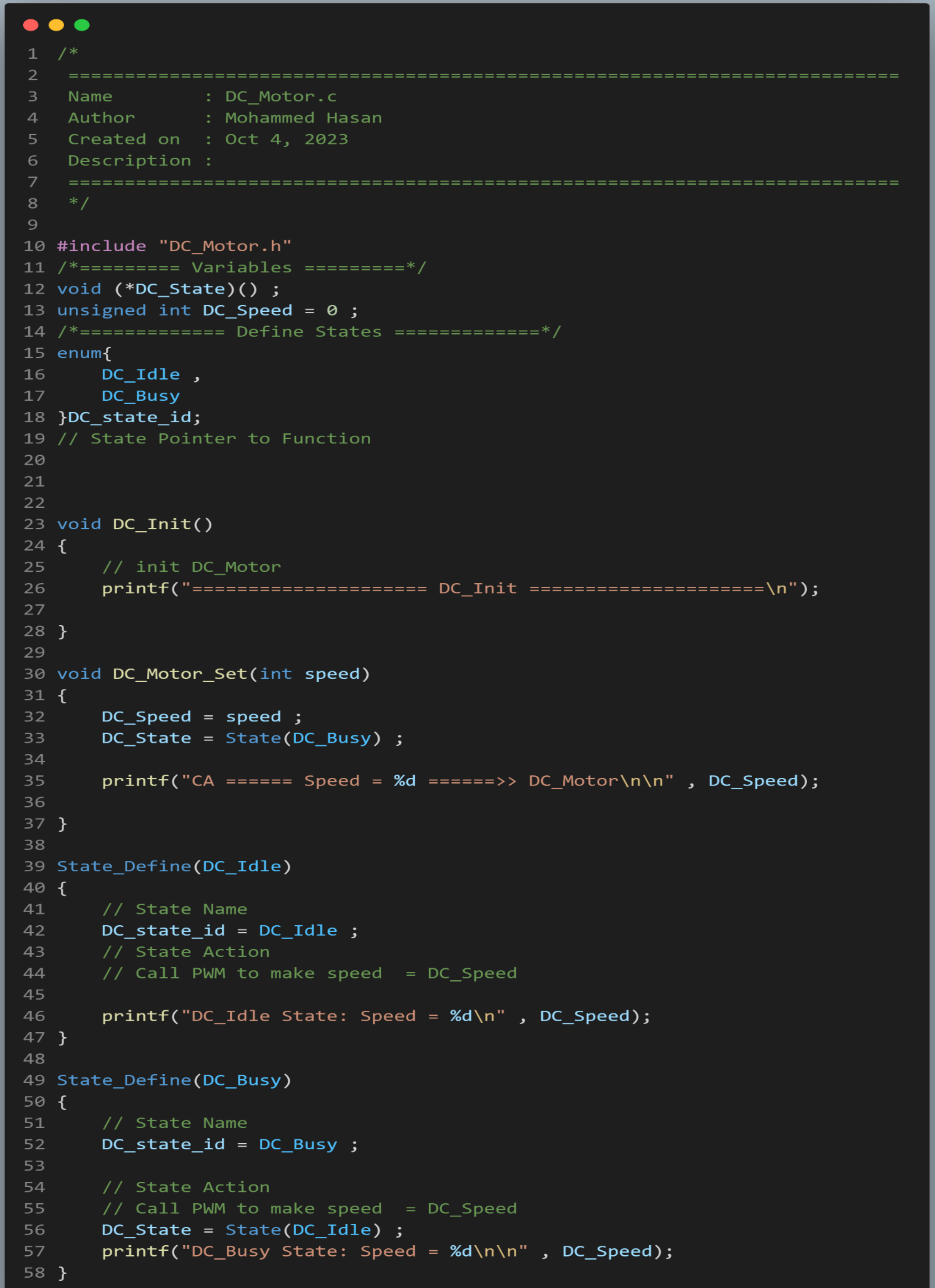
DC.h



```c
/*
 ==============================================================================
 Name        : DC_Motor.h
 Author      : Mohammed Hasan
 Created on  : Oct 4, 2023
 Description :
 ==============================================================================
 */

#ifndef DC_MOTOR_H_
#define DC_MOTOR_H_
#include "State.h"

// Declare State Function
void DC_Motor_Set(int speed);
void DC_Init();
State_Define(DC_Idle);
State_Define(DC_Busy);

extern void (*DC_State)() ;


#endif /* DC_MOTOR_H_ */

```

C Implementation

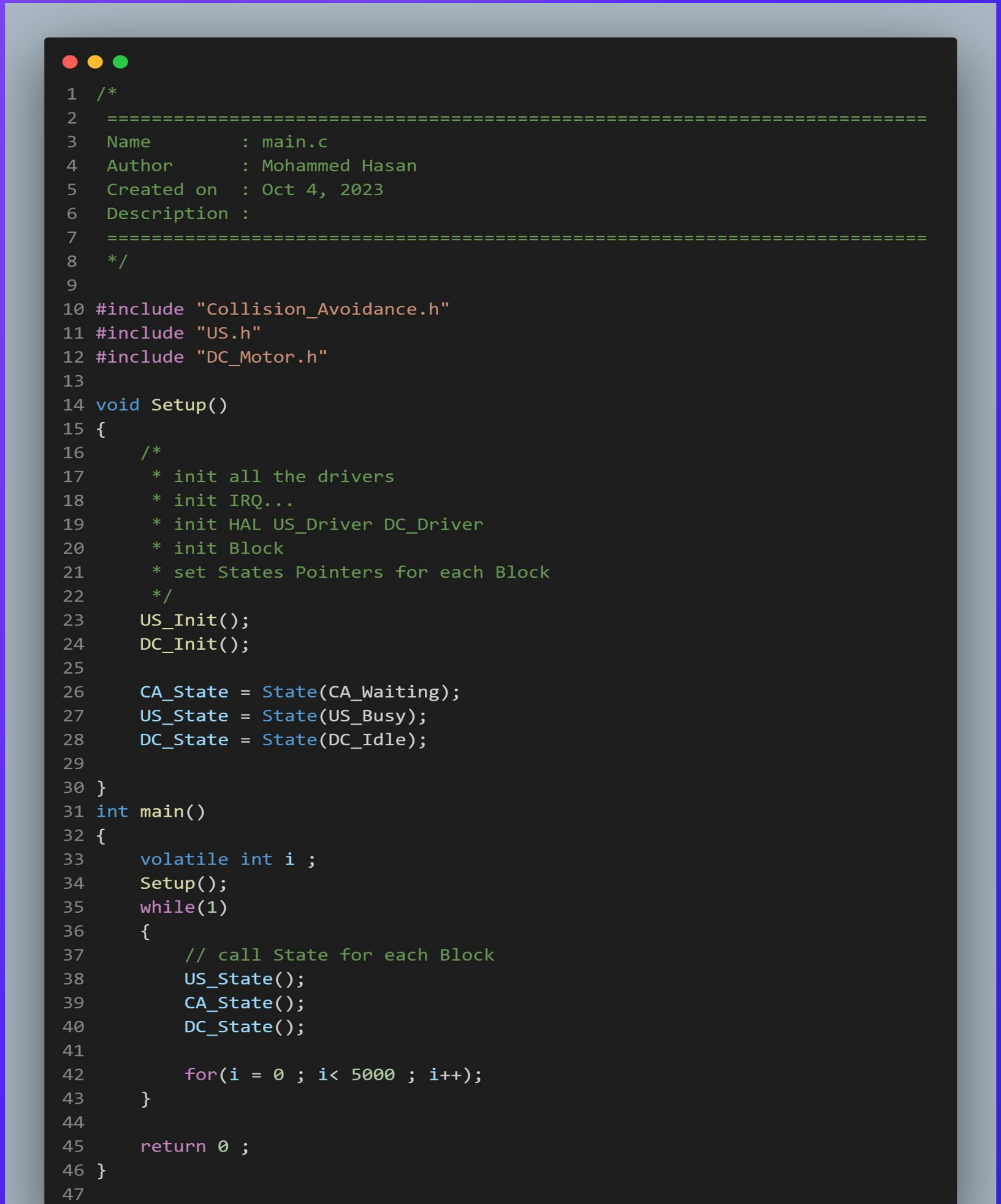
DC.c



```c
/*
 ==========================================================================
 Name         : DC_Motor.c
 Author       : Mohammed Hasan
 Created on   : Oct 4, 2023
 Description :
 ==========================================================================
 */

#include "DC_Motor.h"
/*========= Variables =========*/
void (*DC_State)() ;
unsigned int DC_Speed = 0 ;
/*============= Define States =============*/
enum{
    DC_Idle ,
    DC_Busy
}DC_state_id;
// State Pointer to Function



void DC_Init()
{
    // init DC_Motor
    printf("===================== DC_Init =====================\n");

}

void DC_Motor_Set(int speed)
{
    DC_Speed = speed ;
    DC_State = State(DC_Busy) ;

    printf("CA ====== Speed = %d ======>> DC_Motor\n\n" , DC_Speed);

}

State_Define(DC_Idle)
{
    // State Name
    DC_state_id = DC_Idle ;
    // State Action
    // Call PWM to make speed  = DC_Speed

    printf("DC_Idle State: Speed = %d\n" , DC_Speed);
}

State_Define(DC_Busy)
{
    // State Name
    DC_state_id = DC_Busy ;

    // State Action
    // Call PWM to make speed  = DC_Speed
    DC_State = State(DC_Idle) ;
    printf("DC_Busy State: Speed = %d\n\n" , DC_Speed);
}
```

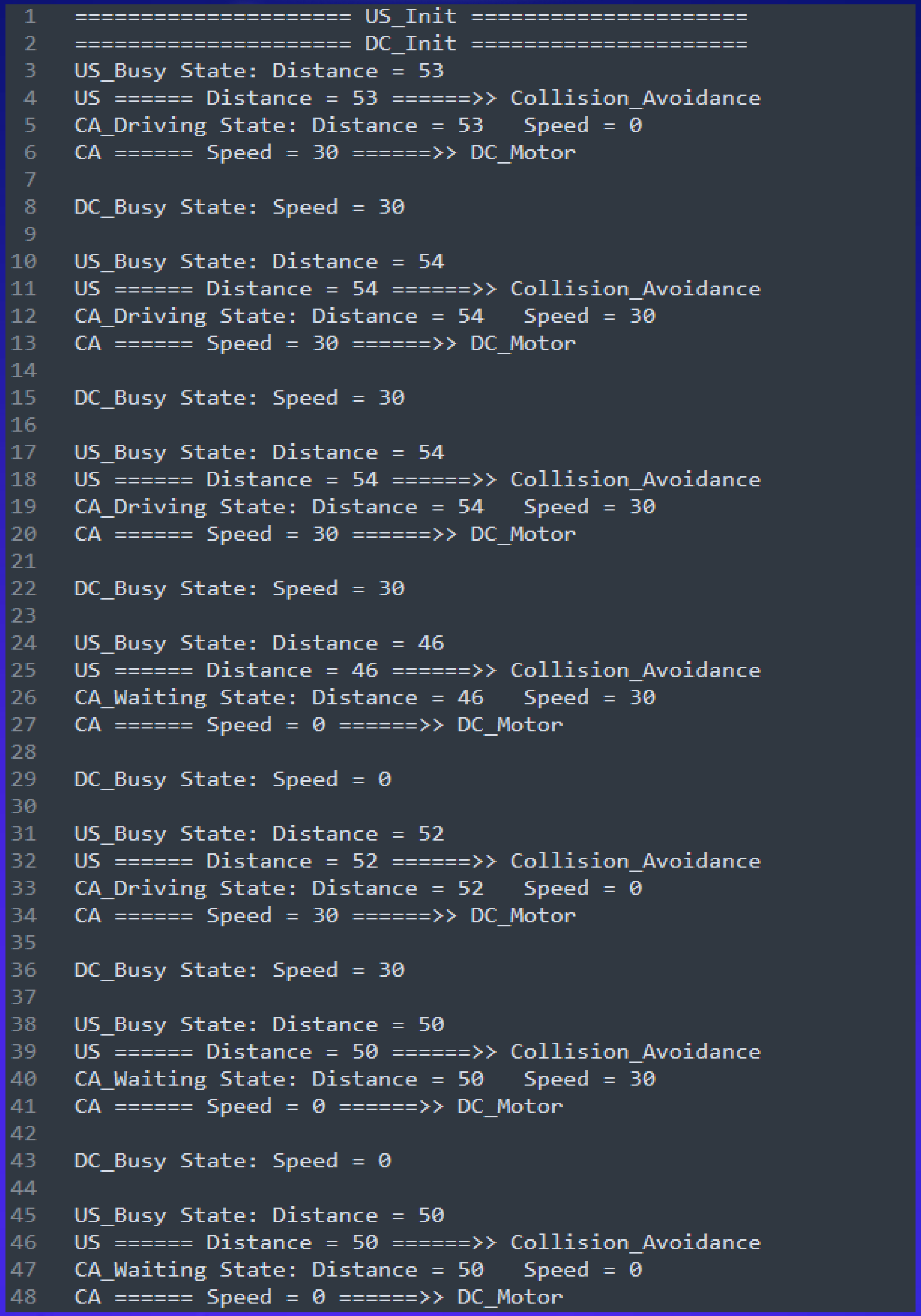
C Implementation

Main.c



```c
/*
 ==========================================================================
 Name        : main.c
 Author      : Mohammed Hasan
 Created on  : Oct 4, 2023
 Description :
 ==========================================================================
 */

#include "Collision_Avoidance.h"
#include "US.h"
#include "DC_Motor.h"

void Setup()
{
    /*
     * init all the drivers
     * init IRQ...
     * init HAL US_Driver DC_Driver
     * init Block
     * set States Pointers for each Block
     */
    US_Init();
    DC_Init();

    CA_State = State(CA_Waiting);
    US_State = State(US_Busy);
    DC_State = State(DC_Idle);

}
int main()
{
    volatile int i ;
    Setup();
    while(1)
    {
        // call State for each Block
        US_State();
        CA_State();
        DC_State();

        for(i = 0 ; i< 5000 ; i++);
    }

    return 0 ;
}

```

Log.txt



```
===================== US_Init =====================
===================== DC_Init =====================
US_Busy State: Distance = 53
US ====== Distance = 53 ======>> Collision_Avoidance
CA_Driving State: Distance = 53   Speed = 0
CA ====== Speed = 30 ======>> DC_Motor

DC_Busy State: Speed = 30

US_Busy State: Distance = 54
US ====== Distance = 54 ======>> Collision_Avoidance
CA_Driving State: Distance = 54   Speed = 30
CA ====== Speed = 30 ======>> DC_Motor

DC_Busy State: Speed = 30

US_Busy State: Distance = 54
US ====== Distance = 54 ======>> Collision_Avoidance
CA_Driving State: Distance = 54   Speed = 30
CA ====== Speed = 30 ======>> DC_Motor

DC_Busy State: Speed = 30

US_Busy State: Distance = 46
US ====== Distance = 46 ======>> Collision_Avoidance
CA_Waiting State: Distance = 46   Speed = 30
CA ====== Speed = 0 ======>> DC_Motor

DC_Busy State: Speed = 0

US_Busy State: Distance = 52
US ====== Distance = 52 ======>> Collision_Avoidance
CA_Driving State: Distance = 52   Speed = 0
CA ====== Speed = 30 ======>> DC_Motor

DC_Busy State: Speed = 30

US_Busy State: Distance = 50
US ====== Distance = 50 ======>> Collision_Avoidance
CA_Waiting State: Distance = 50   Speed = 30
CA ====== Speed = 0 ======>> DC_Motor

DC_Busy State: Speed = 0

US_Busy State: Distance = 50
US ====== Distance = 50 ======>> Collision_Avoidance
CA_Waiting State: Distance = 50   Speed = 0
CA ====== Speed = 0 ======>> DC_Motor
```

THANK YOU!