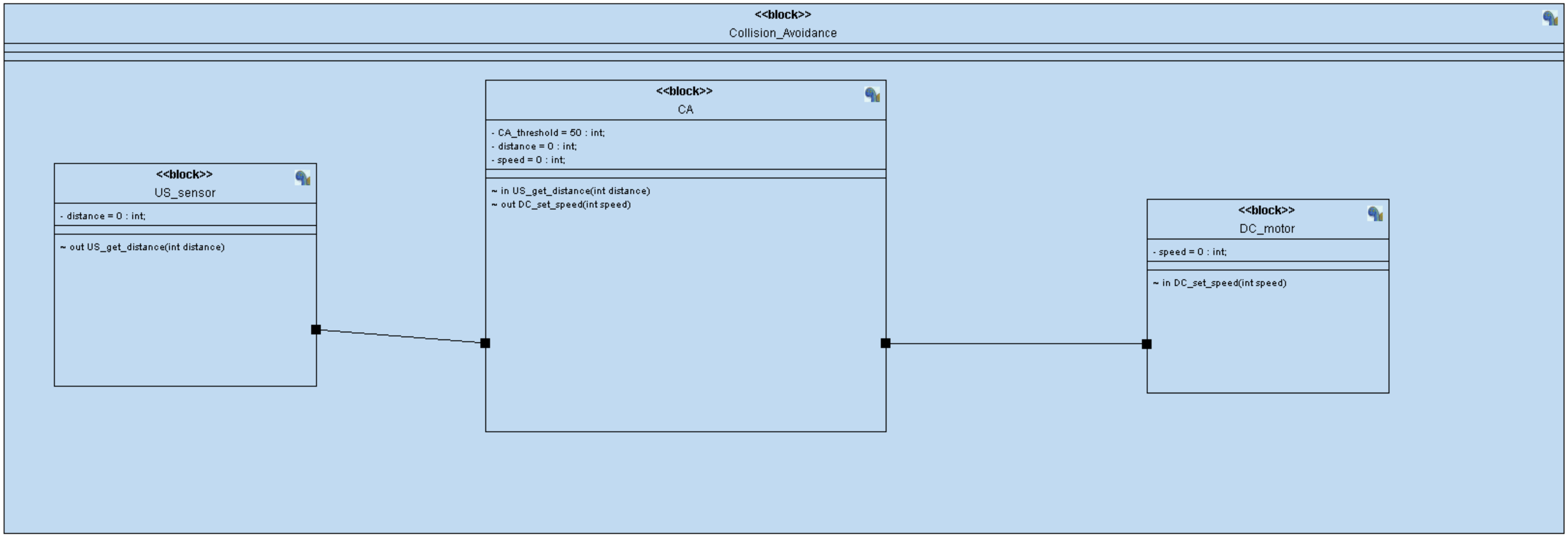
Collision Avoidance Lab

1. **Brief:** program to receive data from ultrasonic sensor and control DC motor based on sensor readings, using State Machine.

2. **Bolck Diagram:**

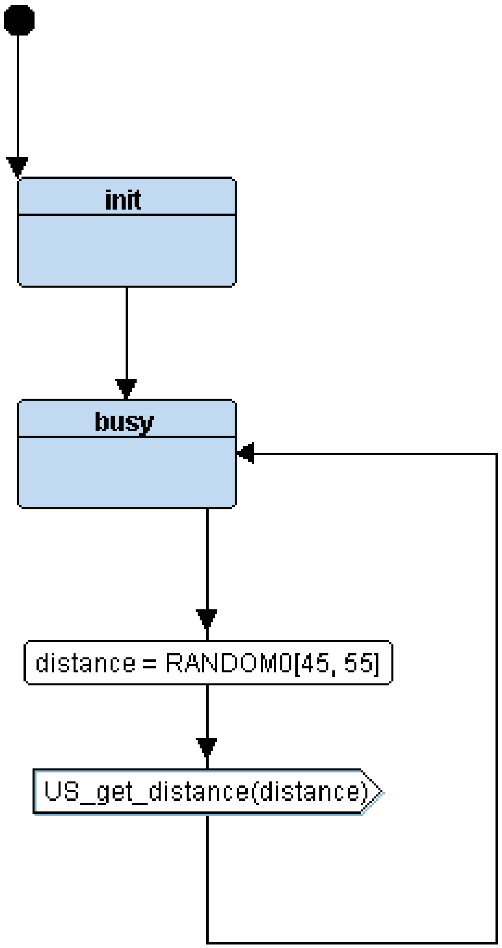

<<block>>
Collision_Avoidance
<<block>>
US_sensor
- distance = 0 : int;
~ out US_get_distance(int distance)
<<block>>
CA
- CA_threshold = 50 : int;
- distance = 0 : int;
- speed = 0 : int;
~ in US_get_distance(int distance)
~ out DC_set_speed(int speed)
<<block>>
DC_motor
- speed = 0 : int;
~ in DC_set_speed(int speed)

3. State Machine - Logical Design:

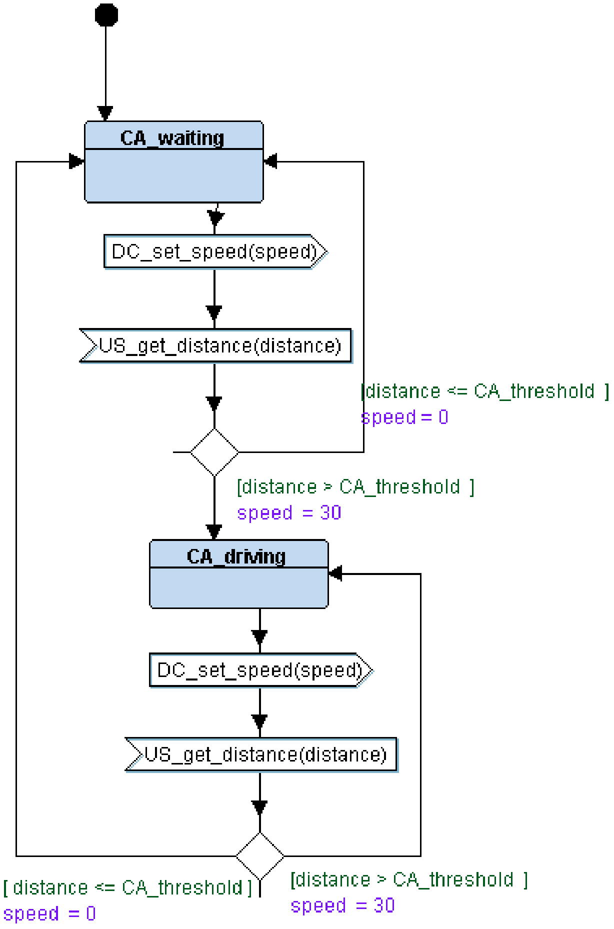
3.1. Ultra Sonic Sensor Module:

3.2. Collision Avoidance Module:

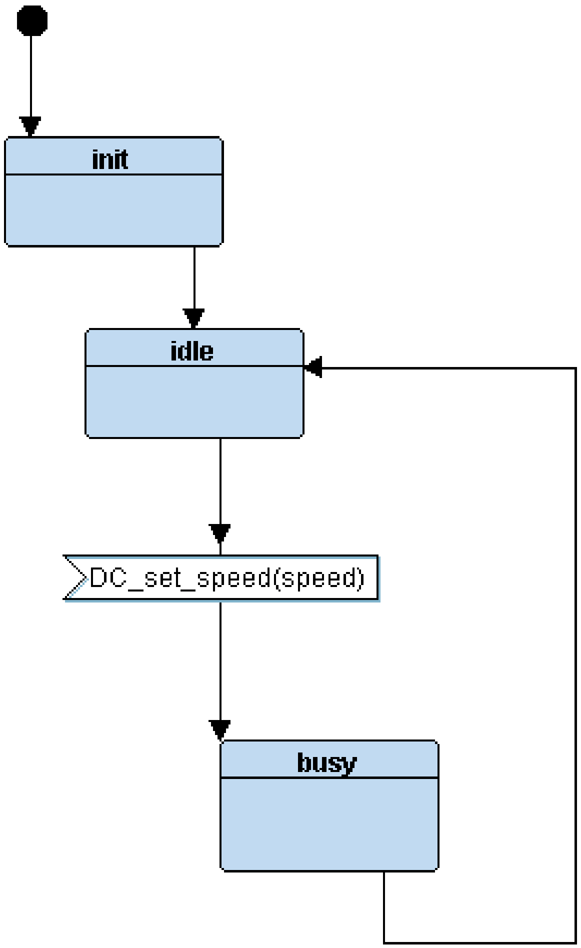
3.3. DC_motor Module:



3.1. Figure 1.


CA_waiting
DC_set_speed(speed)
US_get_distance(distance)
[distance <= CA_threshold]
speed = 0
[distance > CA_threshold]
speed = 30
CA_driving
DC_set_speed(speed)
US_get_distance(distance)
[distance <= CA_threshold]
speed = 0
[distance > CA_threshold]
speed = 30

3.2. Figure 2.



3.3. Figure 3.

4. Sequence Diagram:

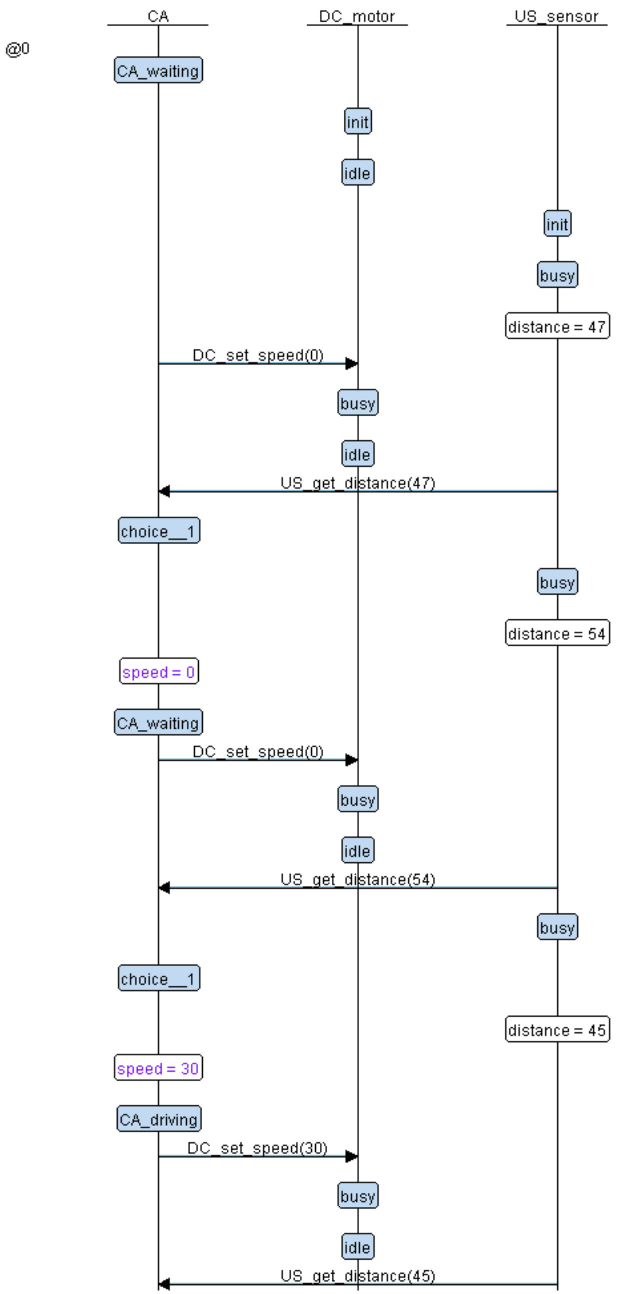


Figure 4.1

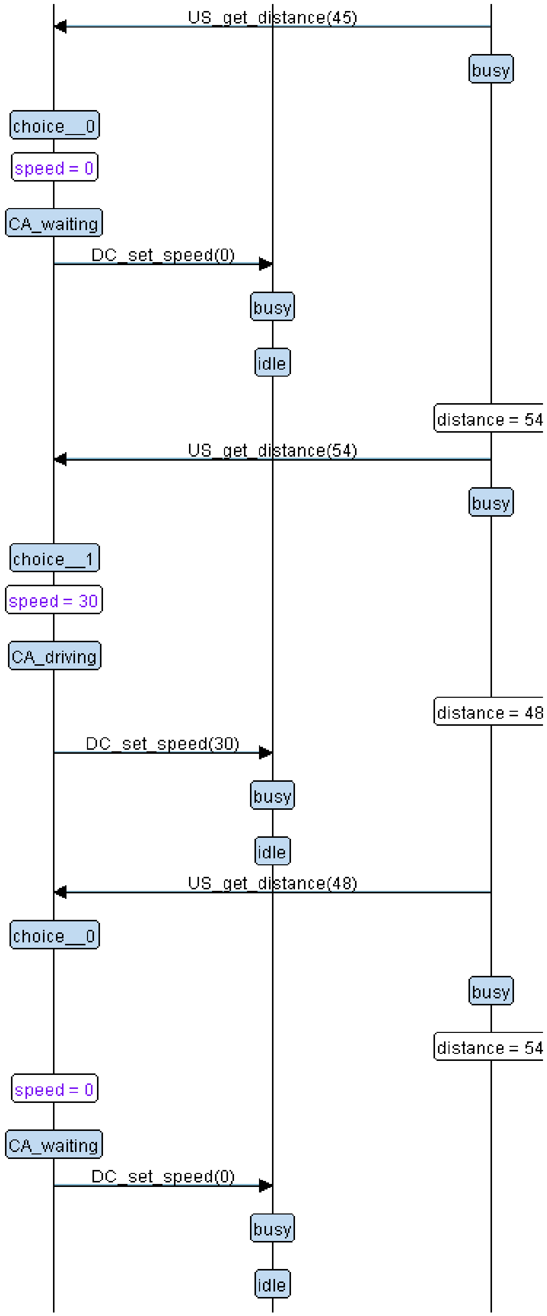
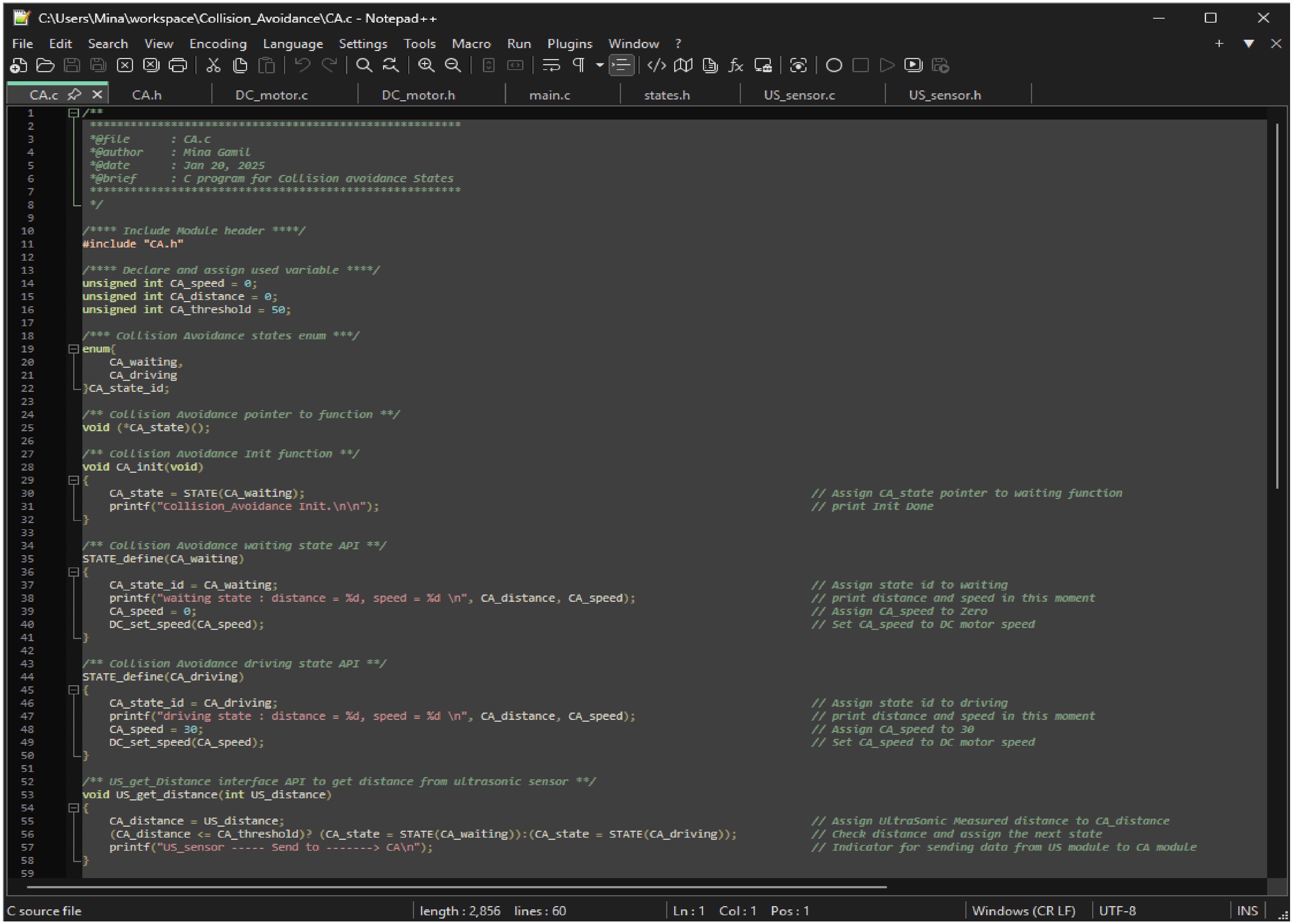


Figure 4.2

5. C Implementation code:

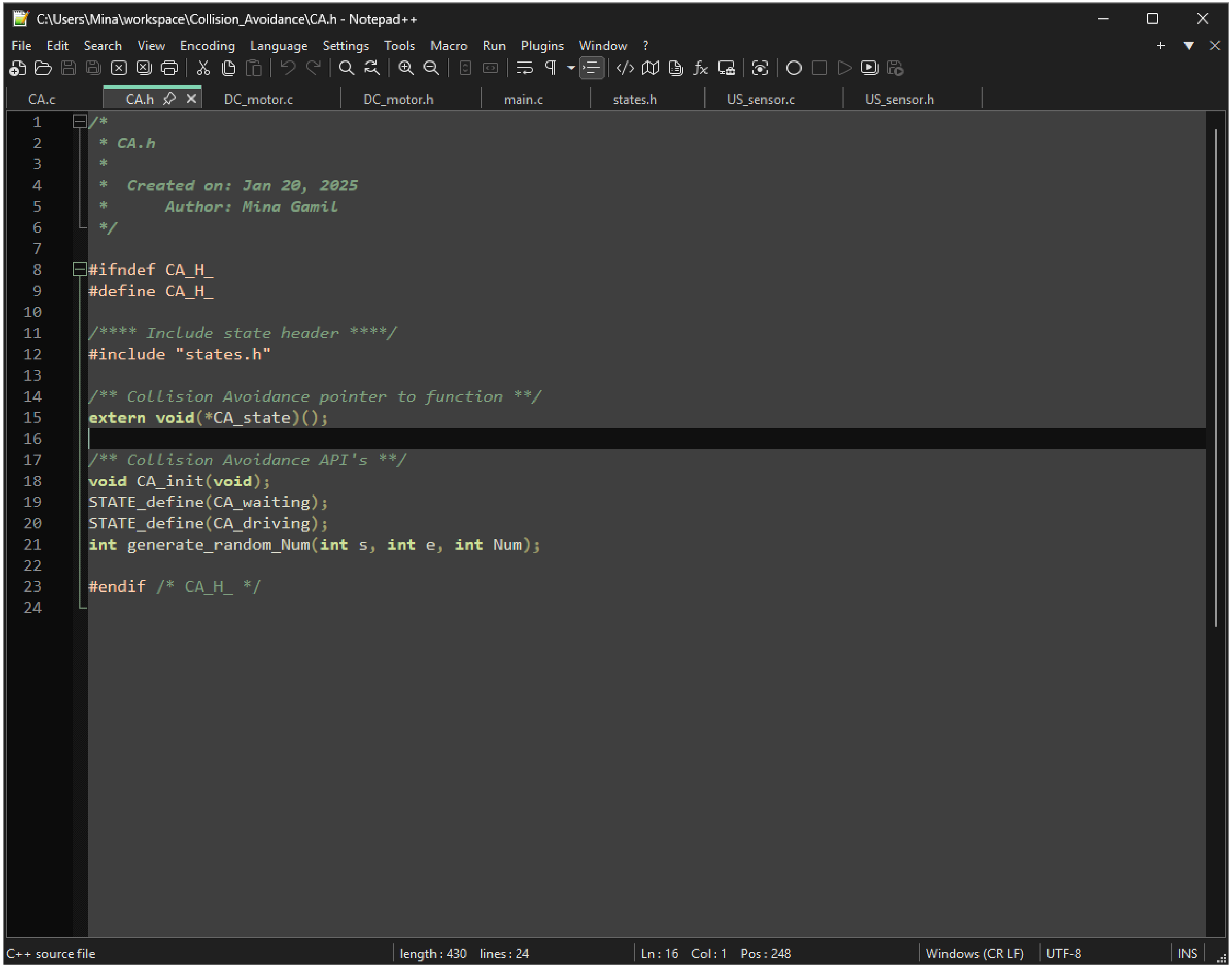
5.1. Collision Avoidance:

5.1.1. Source file:



```c
/**
 *******************************************************
 *@file      : CA.c
 *@author    : Mina Gamil
 *@date      : Jan 20, 2025
 *@brief     : C program for Collision avoidance States
 *******************************************************
 */

/**** Include Module header ****/
#include "CA.h"

/**** Declare and assign used variable ****/
unsigned int CA_speed = 0;
unsigned int CA_distance = 0;
unsigned int CA_threshold = 50;

/*** Collision Avoidance states enum ***/
enum{
    CA_waiting,
    CA_driving
}CA_state_id;

/** Collision Avoidance pointer to function **/
void (*CA_state)();

/** Collision Avoidance Init function **/
void CA_init(void)
{
    CA_state = STATE(CA_waiting);                                                                           // Assign CA_state pointer to waiting function
    printf("Collision_Avoidance Init.\n\n");                                                                // print Init Done
}

/** Collision Avoidance waiting state API **/
STATE_define(CA_waiting)
{
    CA_state_id = CA_waiting;                                                                               // Assign state id to waiting
    printf("waiting state : distance = %d, speed = %d \n", CA_distance, CA_speed);                          // print distance and speed in this moment
    CA_speed = 0;                                                                                           // Assign CA_speed to Zero
    DC_set_speed(CA_speed);                                                                                 // Set CA_speed to DC motor speed
}

/** Collision Avoidance driving state API **/
STATE_define(CA_driving)
{
    CA_state_id = CA_driving;                                                                               // Assign state id to driving
    printf("driving state : distance = %d, speed = %d \n", CA_distance, CA_speed);                          // print distance and speed in this moment
    CA_speed = 30;                                                                                          // Assign CA_speed to 30
    DC_set_speed(CA_speed);                                                                                 // Set CA_speed to DC motor speed
}

/** US_get_Distance interface API to get distance from ultrasonic sensor **/
void US_get_distance(int US_distance)
{
    CA_distance = US_distance;                                                                              // Assign UltraSonic Measured distance to CA_distance
    (CA_distance <= CA_threshold)? (CA_state = STATE(CA_waiting)):(CA_state = STATE(CA_driving));           // Check distance and assign the next state
    printf("US_sensor ----- Send to -------> CA\n");                                                        // Indicator for sending data from US module to CA module
}

```

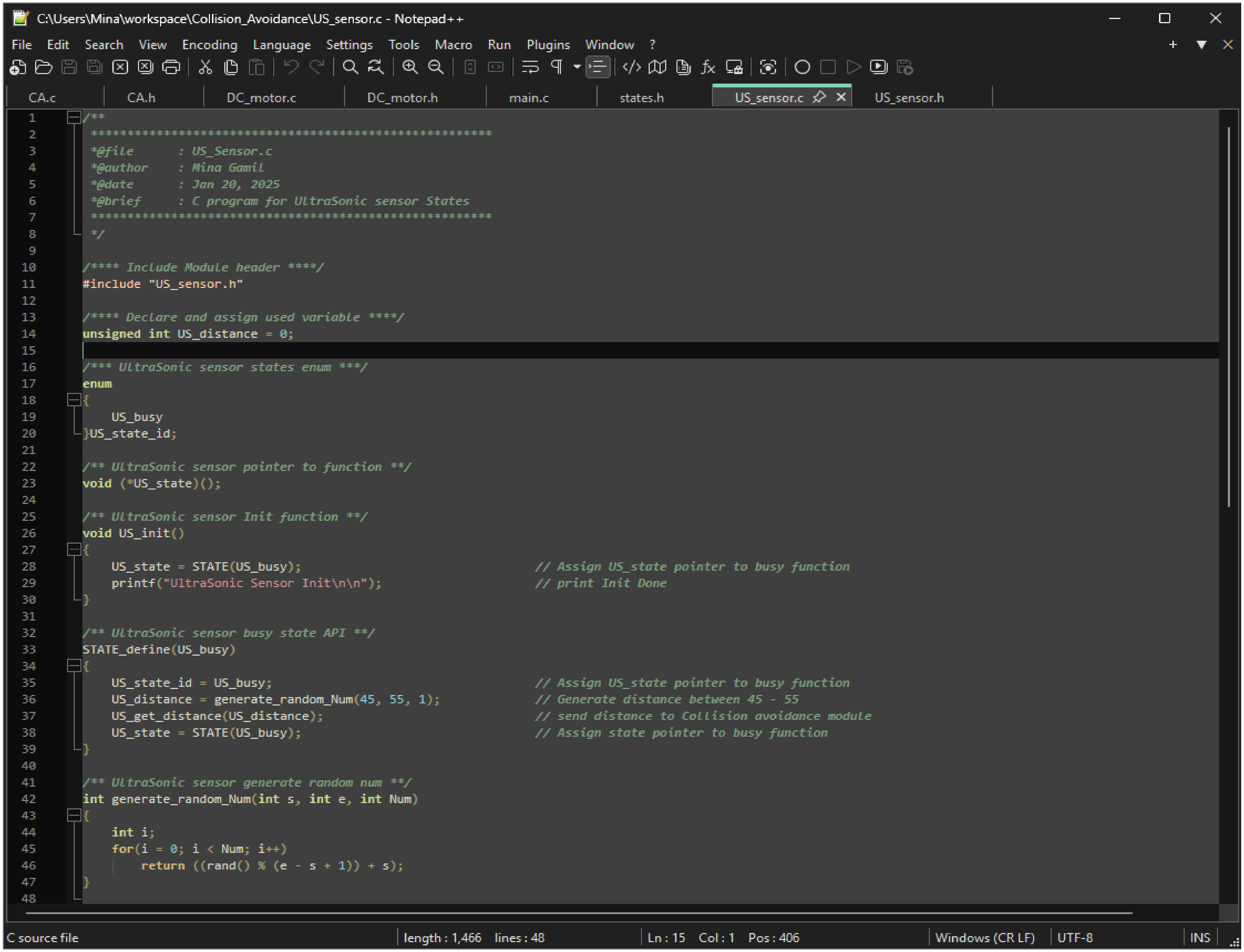
5.1.2. Header File:


C:\Users\Mina\workspace\Collision_Avoidance\CA.h - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
CA.c
CA.h
DC_motor.c
DC_motor.h
main.c
states.h
US_sensor.c
US_sensor.h
C++ source file
length : 430 lines : 24
Ln : 16 Col : 1 Pos : 248
Windows (CR LF)
UTF-8
INS

```c
/*
 * CA.h
 *
 *  Created on: Jan 20, 2025
 *      Author: Mina Gamil
 */

#ifndef CA_H_
#define CA_H_

/**** Include state header ****/
#include "states.h"

/** Collision Avoidance pointer to function **/
extern void(*CA_state)();

/** Collision Avoidance API's **/
void CA_init(void);
STATE_define(CA_waiting);
STATE_define(CA_driving);
int generate_random_Num(int s, int e, int Num);

#endif /* CA_H_ */

```

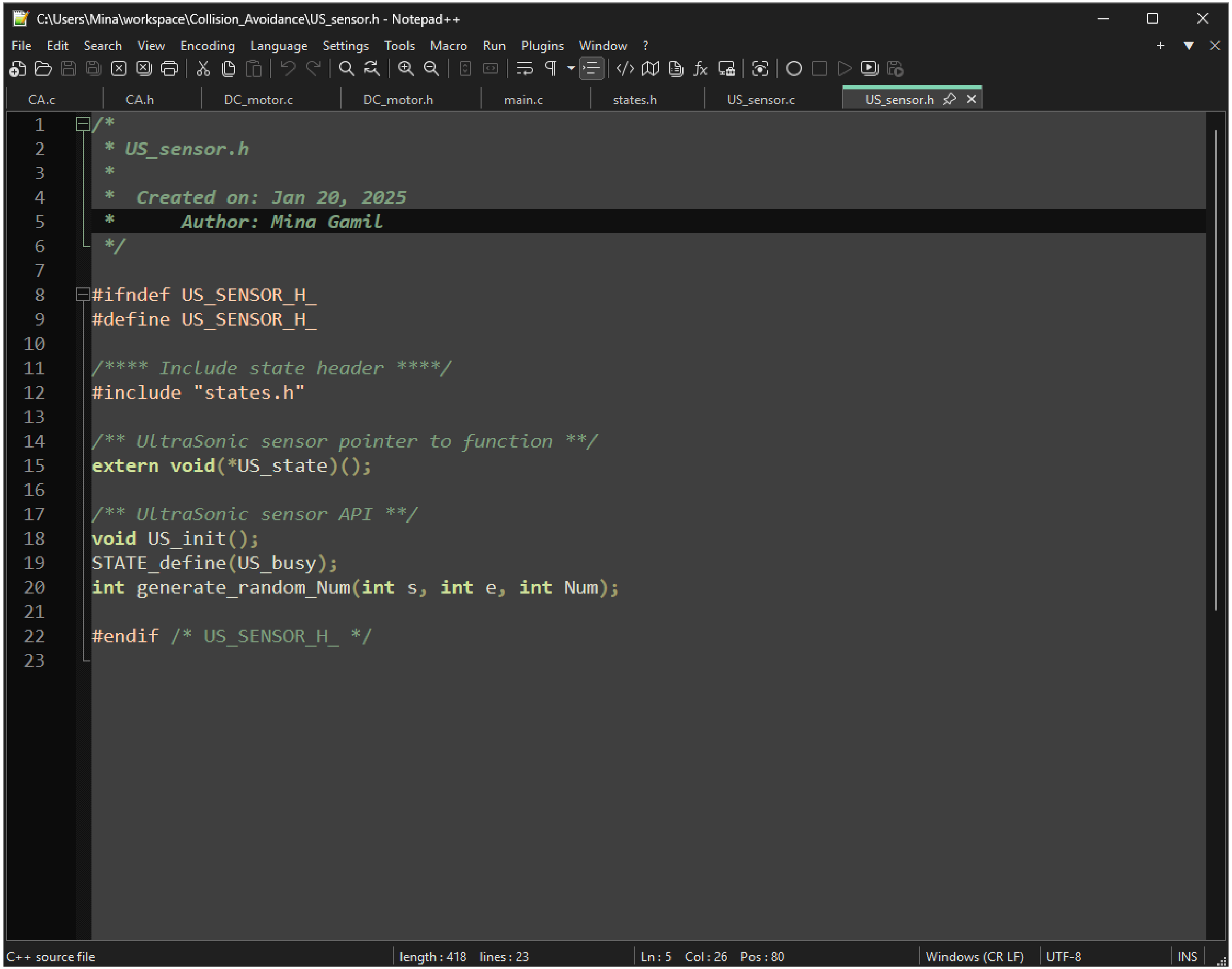
5.2. Ultra Sonic Sensor:

5.2.1. Source file:


C:\Users\Mina\workspace\Collision_Avoidance\US_sensor.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
CA.c
CA.h
DC_motor.c
DC_motor.h
main.c
states.h
US_sensor.c
US_sensor.h
C source file
length : 1,466 lines : 48
Ln : 15 Col : 1 Pos : 406
Windows (CR LF)
UTF-8
INS

```c
/**
 *******************************************************
 *@file      : US_Sensor.c
 *@author    : Mina Gamil
 *@date      : Jan 20, 2025
 *@brief     : C program for UltraSonic sensor States
 *******************************************************
 */

/**** Include Module header ****/
#include "US_sensor.h"

/**** Declare and assign used variable ****/
unsigned int US_distance = 0;

/*** UltraSonic sensor states enum ***/
enum
{
    US_busy
}US_state_id;

/** UltraSonic sensor pointer to function **/
void (*US_state)();

/** UltraSonic sensor Init function **/
void US_init()
{
    US_state = STATE(US_busy);                                // Assign US_state pointer to busy function
    printf("UltraSonic Sensor Init\n\n");                     // print Init Done
}

/** UltraSonic sensor busy state API **/
STATE_define(US_busy)
{
    US_state_id = US_busy;                                    // Assign US_state pointer to busy function
    US_distance = generate_random_Num(45, 55, 1);             // Generate distance between 45 - 55
    US_get_distance(US_distance);                             // send distance to Collision avoidance module
    US_state = STATE(US_busy);                                // Assign state pointer to busy function
}

/** UltraSonic sensor generate random num **/
int generate_random_Num(int s, int e, int Num)
{
    int i;
    for(i = 0; i < Num; i++)
        return ((rand() % (e - s + 1)) + s);
}

```

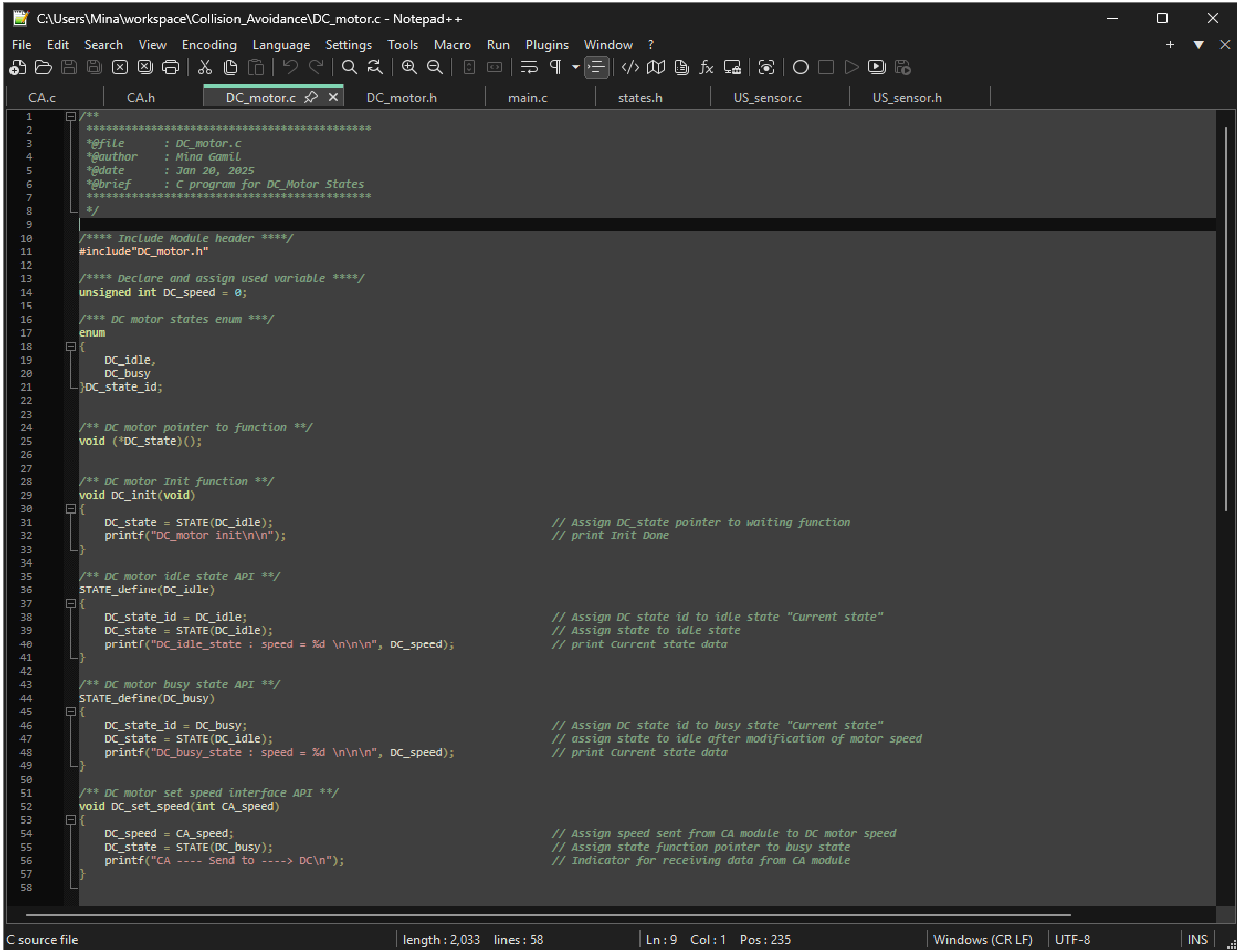
5.2.2. Header File:



```c
/*
 * US_sensor.h
 *
 *  Created on: Jan 20, 2025
 *      Author: Mina Gamil
 */

#ifndef US_SENSOR_H_
#define US_SENSOR_H_

/**** Include state header ****/
#include "states.h"

/** UltraSonic sensor pointer to function **/
extern void(*US_state)();

/** UltraSonic sensor API **/
void US_init();
STATE_define(US_busy);
int generate_random_Num(int s, int e, int Num);

#endif /* US_SENSOR_H_ */

```

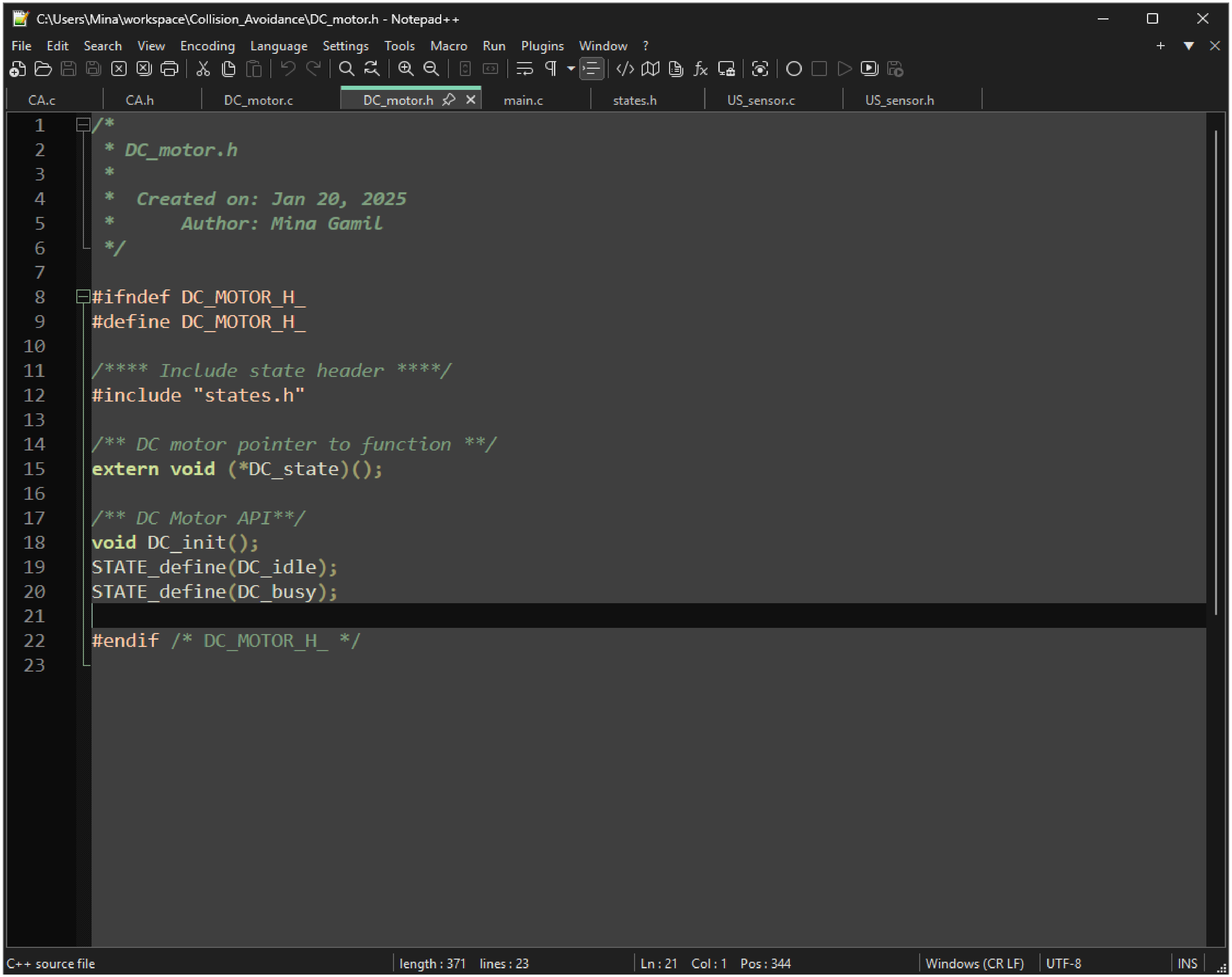
5.3. DC motor:

5.3.1. Source file:


C:\Users\Mina\workspace\Collision_Avoidance\DC_motor.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
CA.c
CA.h
DC_motor.c
DC_motor.h
main.c
states.h
US_sensor.c
US_sensor.h
C source file
length : 2,033 lines : 58
Ln : 9 Col : 1 Pos : 235
Windows (CR LF)
UTF-8
INS

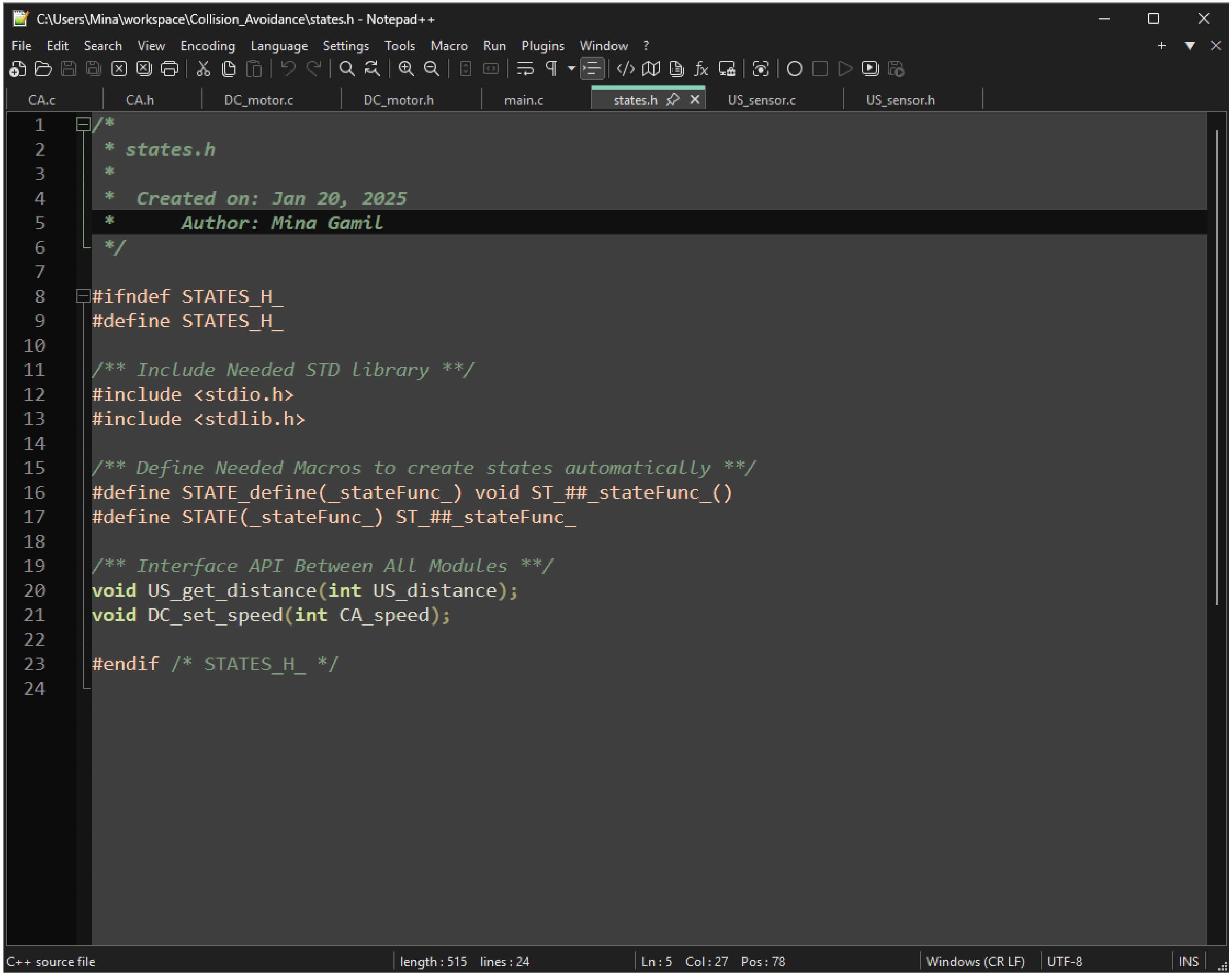
```c
/**
 ********************************************
 *@file      : DC_motor.c
 *@author    : Mina Gamil
 *@date      : Jan 20, 2025
 *@brief     : C program for DC_Motor States
 ********************************************
 */

/**** Include Module header ****/
#include"DC_motor.h"

/**** Declare and assign used variable ****/
unsigned int DC_speed = 0;

/*** DC motor states enum ***/
enum
{
    DC_idle,
    DC_busy
}DC_state_id;


/** DC motor pointer to function **/
void (*DC_state)();


/** DC motor Init function **/
void DC_init(void)
{
    DC_state = STATE(DC_idle);                                          // Assign DC_state pointer to waiting function
    printf("DC_motor init\n\n");                                        // print Init Done
}

/** DC motor idle state API **/
STATE_define(DC_idle)
{
    DC_state_id = DC_idle;                                              // Assign DC state id to idle state "Current state"
    DC_state = STATE(DC_idle);                                          // Assign state to idle state
    printf("DC_idle_state : speed = %d \n\n\n", DC_speed);              // print Current state data
}

/** DC motor busy state API **/
STATE_define(DC_busy)
{
    DC_state_id = DC_busy;                                              // Assign DC state id to busy state "Current state"
    DC_state = STATE(DC_idle);                                          // assign state to idle after modification of motor speed
    printf("DC_busy_state : speed = %d \n\n\n", DC_speed);              // print Current state data
}

/** DC motor set speed interface API **/
void DC_set_speed(int CA_speed)
{
    DC_speed = CA_speed;                                                // Assign speed sent from CA module to DC motor speed
    DC_state = STATE(DC_busy);                                          // Assign state function pointer to busy state
    printf("CA ---- Send to ----> DC\n");                               // Indicator for receiving data from CA module
}

```

5.3.2. Header File:


C:\Users\Mina\workspace\Collision_Avoidance\DC_motor.h - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
CA.c
CA.h
DC_motor.c
DC_motor.h
main.c
states.h
US_sensor.c
US_sensor.h
C++ source file
length : 371 lines : 23
Ln : 21 Col : 1 Pos : 344
Windows (CR LF)
UTF-8
INS

```c
/*
 * DC_motor.h
 *
 *  Created on: Jan 20, 2025
 *      Author: Mina Gamil
 */

#ifndef DC_MOTOR_H_
#define DC_MOTOR_H_

/**** Include state header ****/
#include "states.h"

/** DC motor pointer to function **/
extern void (*DC_state)();

/** DC Motor API**/
void DC_init();
STATE_define(DC_idle);
STATE_define(DC_busy);

#endif /* DC_MOTOR_H_ */

```

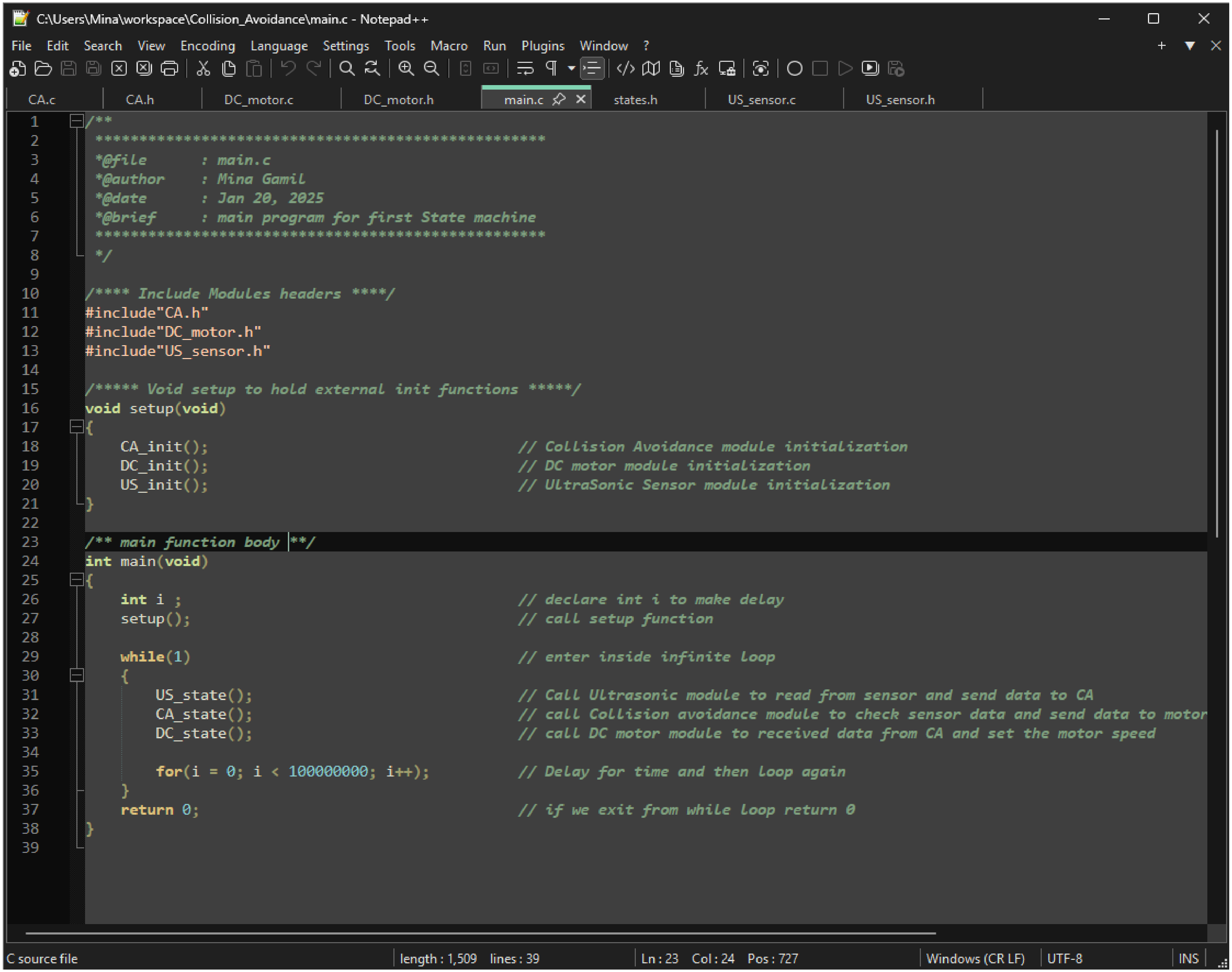
5.4. State Header File:



C:\Users\Mina\workspace\Collision_Avoidance\states.h - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
CA.c
CA.h
DC_motor.c
DC_motor.h
main.c
states.h
US_sensor.c
US_sensor.h

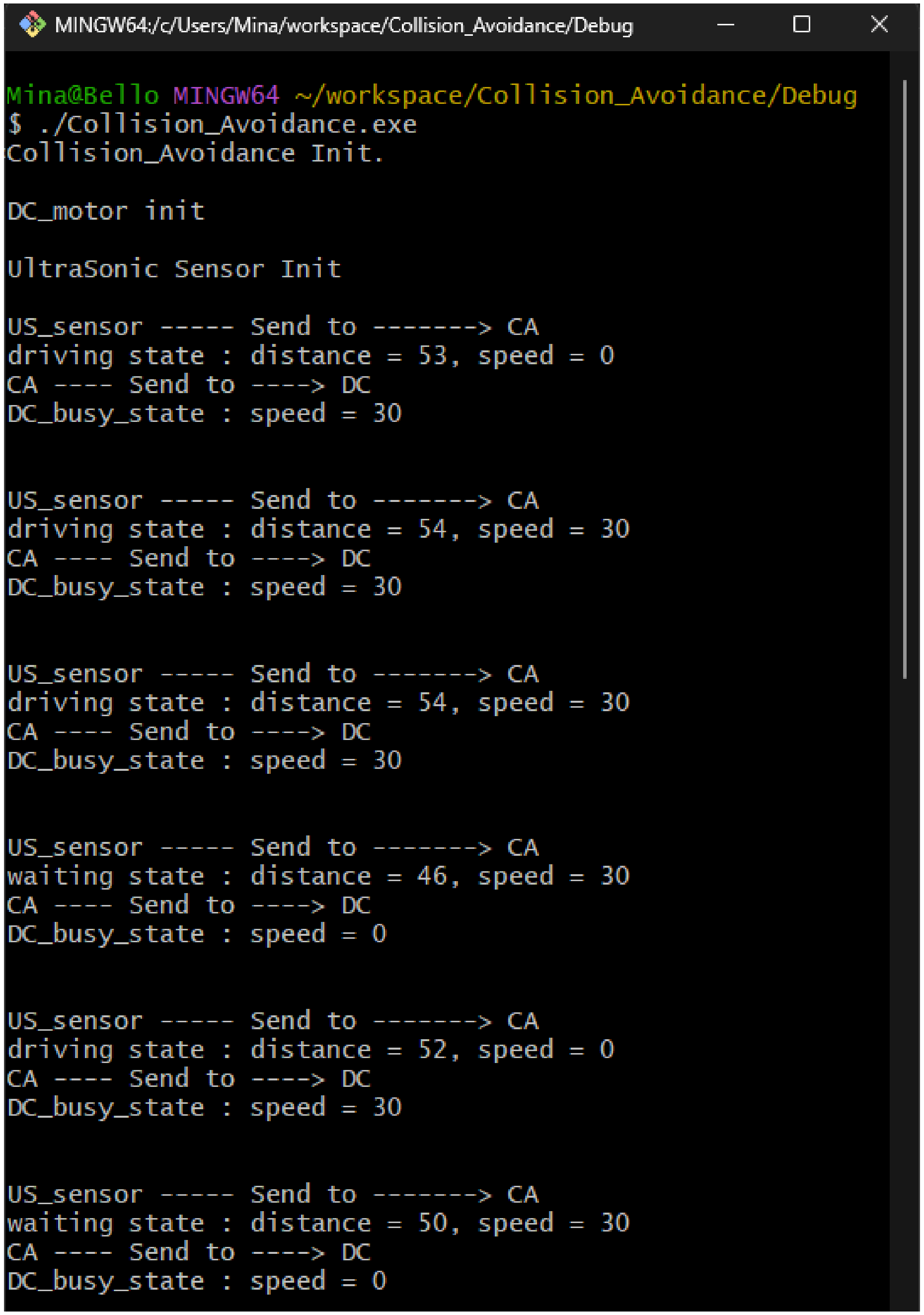
```c
/*
 * states.h
 *
 *  Created on: Jan 20, 2025
 *      Author: Mina Gamil
 */

#ifndef STATES_H_
#define STATES_H_

/** Include Needed STD library **/
#include <stdio.h>
#include <stdlib.h>

/** Define Needed Macros to create states automatically **/
#define STATE_define(_stateFunc_) void ST_##_stateFunc_()
#define STATE(_stateFunc_) ST_##_stateFunc_

/** Interface API Between All Modules **/
void US_get_distance(int US_distance);
void DC_set_speed(int CA_speed);

#endif /* STATES_H_ */

```

5.5. Main Source file:


C:\Users\Mina\workspace\Collision_Avoidance\main.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
CA.c
CA.h
DC_motor.c
DC_motor.h
main.c
states.h
US_sensor.c
US_sensor.h
C source file
length : 1,509 lines : 39
Ln : 23 Col : 24 Pos : 727
Windows (CR LF)
UTF-8
INS

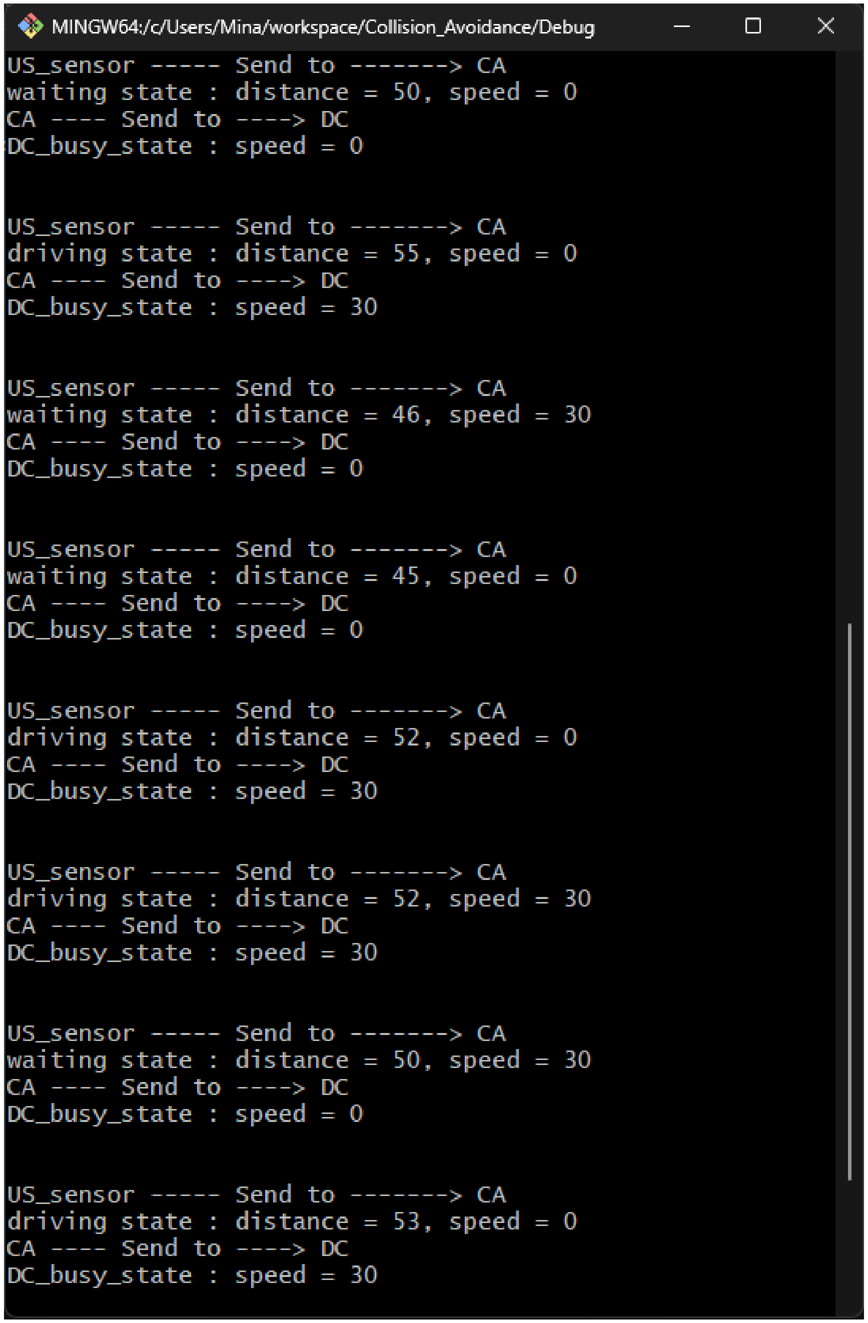
```c
/**
 ***************************************************
 *@file      : main.c
 *@author    : Mina Gamil
 *@date      : Jan 20, 2025
 *@brief     : main program for first State machine
 ***************************************************
 */

/**** Include Modules headers ****/
#include"CA.h"
#include"DC_motor.h"
#include"US_sensor.h"

/***** Void setup to hold external init functions *****/
void setup(void)
{
    CA_init();                                  // Collision Avoidance module initialization
    DC_init();                                  // DC motor module initialization
    US_init();                                  // UltraSonic Sensor module initialization
}

/** main function body **/
int main(void)
{
    int i ;                                     // declare int i to make delay
    setup();                                    // call setup function

    while(1)                                    // enter inside infinite loop
    {
        US_state();                             // Call Ultrasonic module to read from sensor and send data to CA
        CA_state();                             // call Collision avoidance module to check sensor data and send data to motor
        DC_state();                             // call DC motor module to received data from CA and set the motor speed

        for(i = 0; i < 100000000; i++);         // Delay for time and then loop again
    }
    return 0;                                   // if we exit from while loop return 0
}

```

6. C implementation result:



```
Mina@Bello MINGW64 ~/workspace/Collision_Avoidance/Debug
$ ./Collision_Avoidance.exe
Collision_Avoidance Init.

DC_motor init

UltraSonic Sensor Init

US_sensor ----- Send to -------> CA
driving state : distance = 53, speed = 0
CA ---- Send to ----> DC
DC_busy_state : speed = 30


US_sensor ----- Send to -------> CA
driving state : distance = 54, speed = 30
CA ---- Send to ----> DC
DC_busy_state : speed = 30


US_sensor ----- Send to -------> CA
driving state : distance = 54, speed = 30
CA ---- Send to ----> DC
DC_busy_state : speed = 30


US_sensor ----- Send to -------> CA
waiting state : distance = 46, speed = 30
CA ---- Send to ----> DC
DC_busy_state : speed = 0


US_sensor ----- Send to -------> CA
driving state : distance = 52, speed = 0
CA ---- Send to ----> DC
DC_busy_state : speed = 30


US_sensor ----- Send to -------> CA
waiting state : distance = 50, speed = 30
CA ---- Send to ----> DC
DC_busy_state : speed = 0
```

Figure 6.1



```
US_sensor ----- Send to -------> CA
waiting state : distance = 50, speed = 0
CA ---- Send to ----> DC
DC_busy_state : speed = 0


US_sensor ----- Send to -------> CA
driving state : distance = 55, speed = 0
CA ---- Send to ----> DC
DC_busy_state : speed = 30


US_sensor ----- Send to -------> CA
waiting state : distance = 46, speed = 30
CA ---- Send to ----> DC
DC_busy_state : speed = 0


US_sensor ----- Send to -------> CA
waiting state : distance = 45, speed = 0
CA ---- Send to ----> DC
DC_busy_state : speed = 0


US_sensor ----- Send to -------> CA
driving state : distance = 52, speed = 0
CA ---- Send to ----> DC
DC_busy_state : speed = 30


US_sensor ----- Send to -------> CA
driving state : distance = 52, speed = 30
CA ---- Send to ----> DC
DC_busy_state : speed = 30


US_sensor ----- Send to -------> CA
waiting state : distance = 50, speed = 30
CA ---- Send to ----> DC
DC_busy_state : speed = 0


US_sensor ----- Send to -------> CA
driving state : distance = 53, speed = 0
CA ---- Send to ----> DC
DC_busy_state : speed = 30
```

Figure 6.2