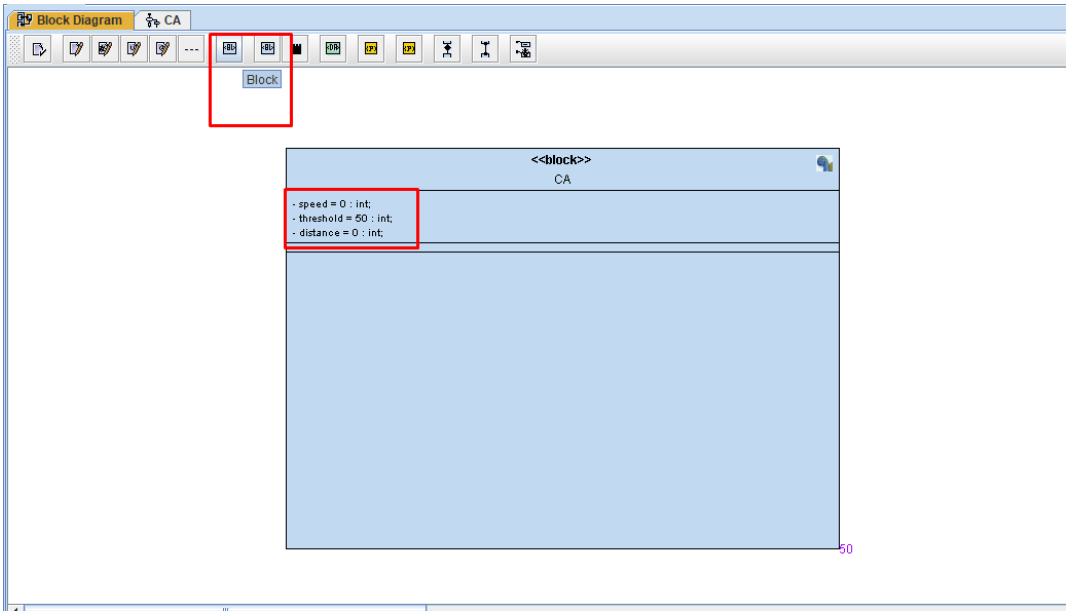
Unit 4 Lesson 2

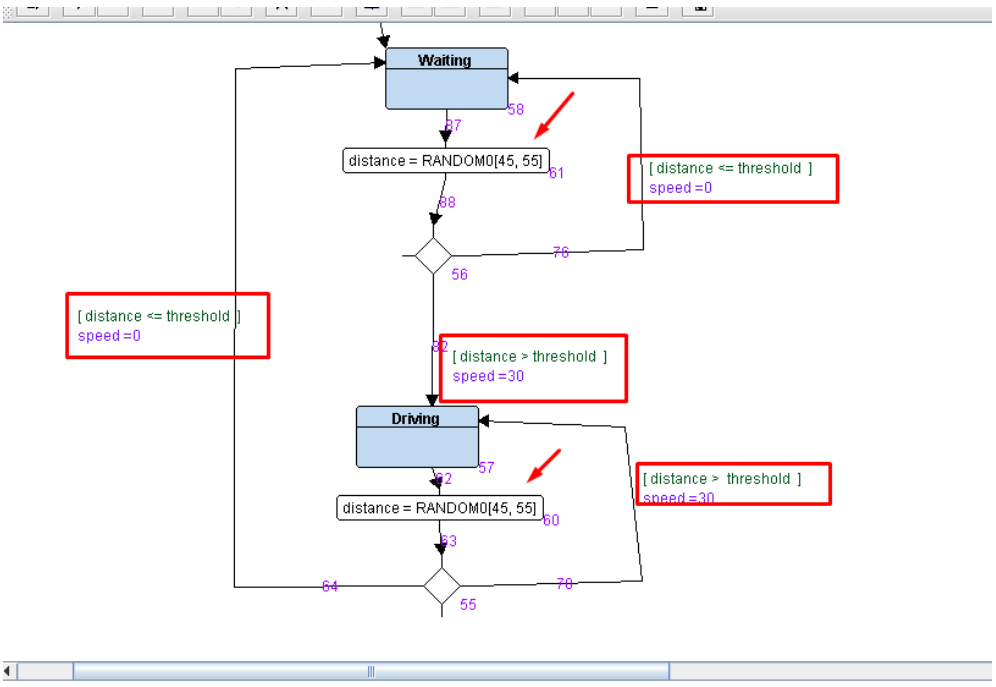
Guirguis Hedia

Version 1 :

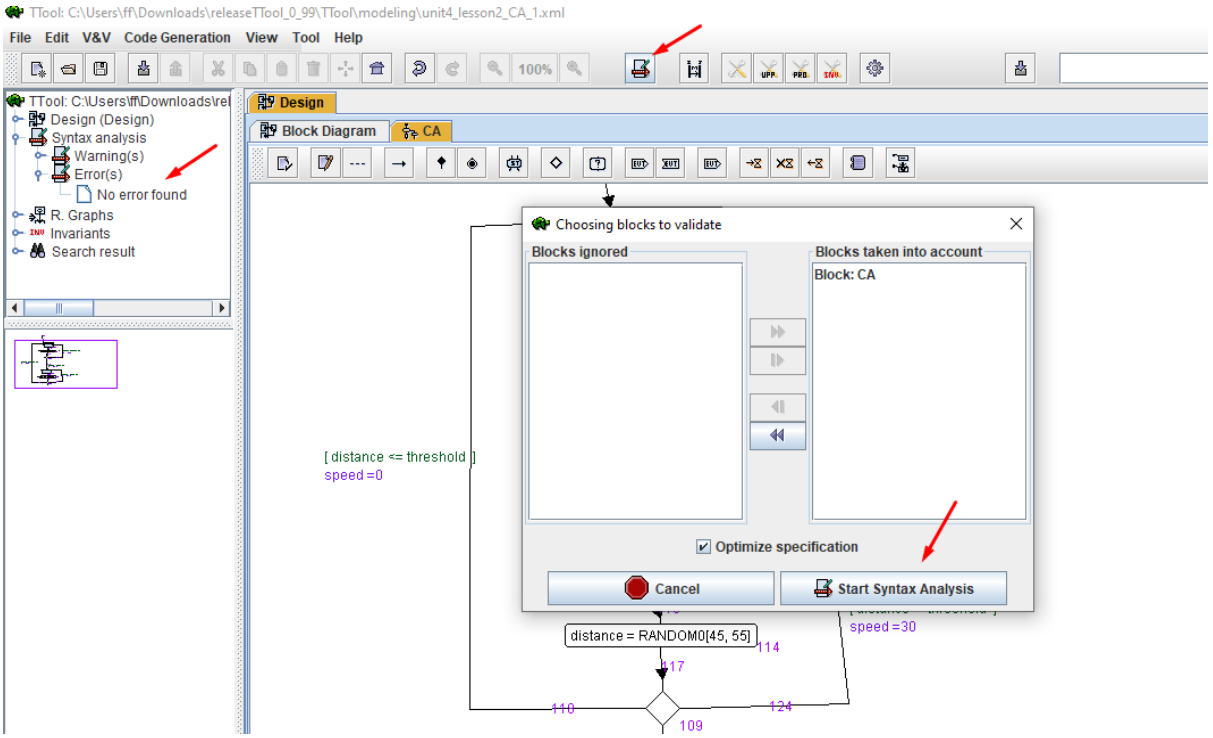
Design :



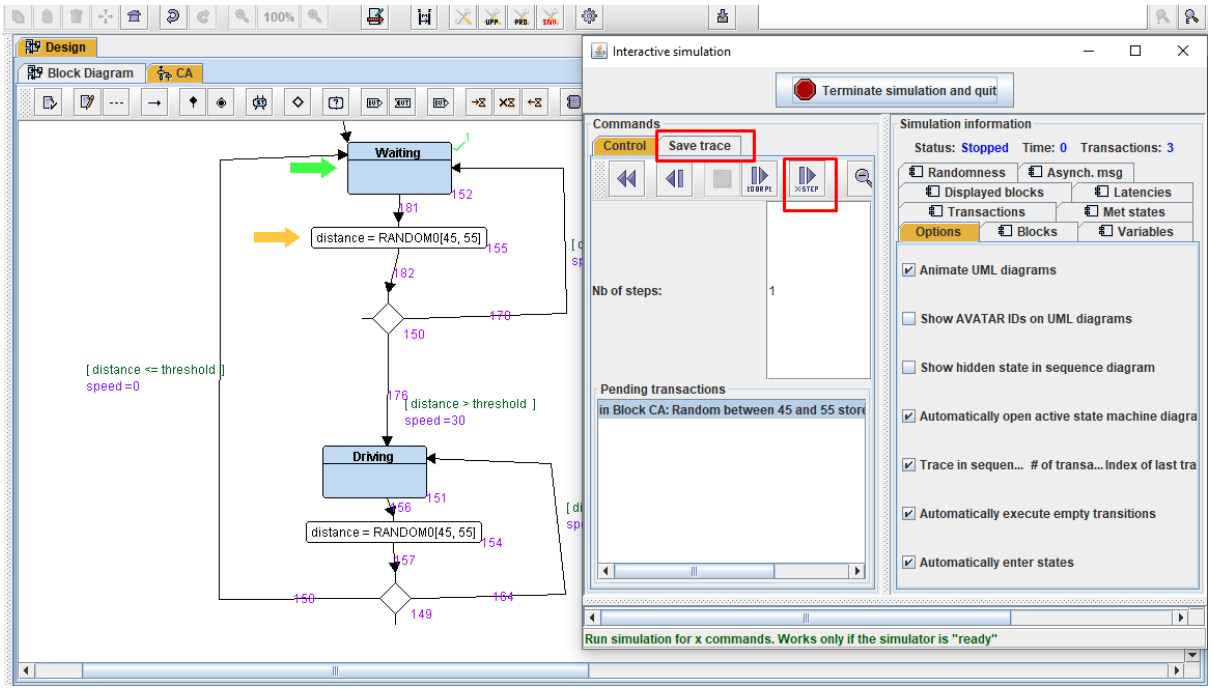
State Machine Diagram :



Start Test Syntax :



if you Found No Error in Syntax Start interactive Simulation:



Start Coding :

State.h File :

```c
/*
 * state.h
 */

#ifndef STATE_H_
#define STATE_H_
#include "stdio.h"
#include "stdlib.h"

//Automatic State Function Generated
#define STATE_define(_statFUN_) void ST_##_statFUN_()
#define STATE(_statFUN_)     ST_##_statFUN_


#endif /* STATE_H_ */

```

CA.h File :

```c
/*
 * CA.h
 */

#ifndef CA_H_
#define CA_H_

#include "state.h"
//Define States
enum {
    CA_waiting,
    CA_driving
}CA_state_id;

//STATE Pointer to Function
extern void (*CA_state )();

//Declare States Functions CA
STATE_define(CA_waiting);
STATE_define(CA_driving);
#endif /* CA_H_ */

```

CA.c file :

main.c | CA.h | state.h | CA.c

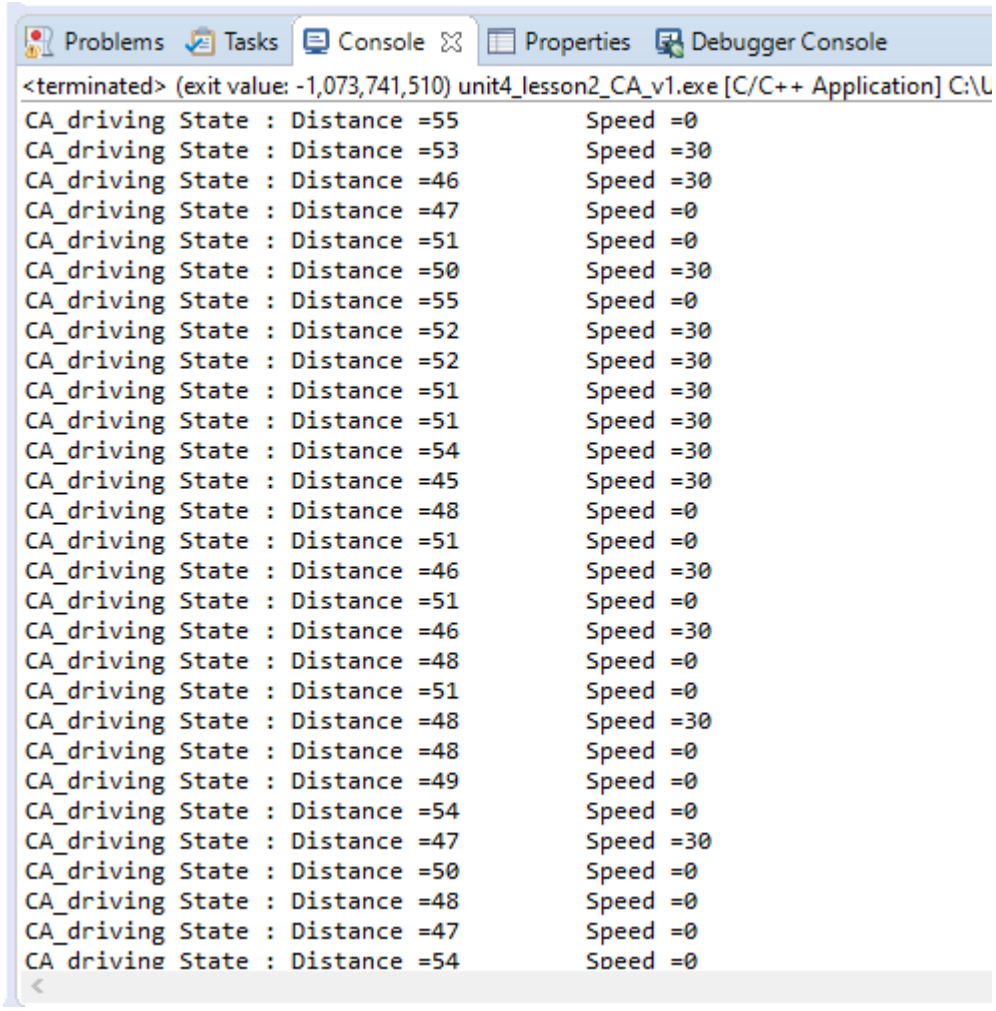
```c
/*
 * CA.c
 */

#include "CA.h"
//Variables
int CA_speed =0;
int CA_distance =0;
int CA_threshold =50;


//STATE Pointer to Function
void (*CA_state )();
int US_GET_distance_rondom(int l,int r,int count);

STATE_define(CA_waiting)
{
    //State_Name
    CA_state_id=CA_waiting;

    //State_Action
    CA_speed=0;

    //Event_Check
    //US_GET_Distance(CA_distance)
    CA_distance=US_GET_distance_rondom(45,55,1);;

    (CA_distance <=CA_threshold)?(CA_state=STATE(CA_waiting)):(CA_state=STATE(CA_driving));

    printf("CA_driving State : Distance =%d \tSpeed =%d\n",CA_distance,CA_speed);

}
```

```c
STATE_define(CA_driving)
{
    //State_Name
    CA_state_id=CA_driving;

    //State_Action
    CA_speed=30;

    //Event_Check
    //US_GET_Distance(CA_distance)
    CA_distance=US_GET_distance_rondom(45,55,1);
    (CA_distance <= CA_threshold)?(CA_state=STATE(CA_waiting)):(CA_state=STATE(CA_driving));

    printf("CA_driving State : Distance =%d \tSpeed =%d\n",CA_distance,CA_speed);
}

int US_GET_distance_rondom(int l,int r,int count)
{
    int i;

    for(i = 0;i < count; i++) {
        int rand_num = (rand() % (r - l + 1)) + l;
    return rand_num;
    }
}
```

main.c File :

```c
/*
 * main.c
 */

#include "CA.h"

void setup()
{
    //init all The Drivers;
    //inti IRQ ....
    //init HAL US_Driver DC_Driver
    //init Block
    //Set States Pointer for each Block

    CA_state=STATE(CA_waiting);
}
int main ()
{
    setup();

    while(1)
    {
        //Call State for each Block
        CA_state();
    }
    return 0;
}
```

Output Result :



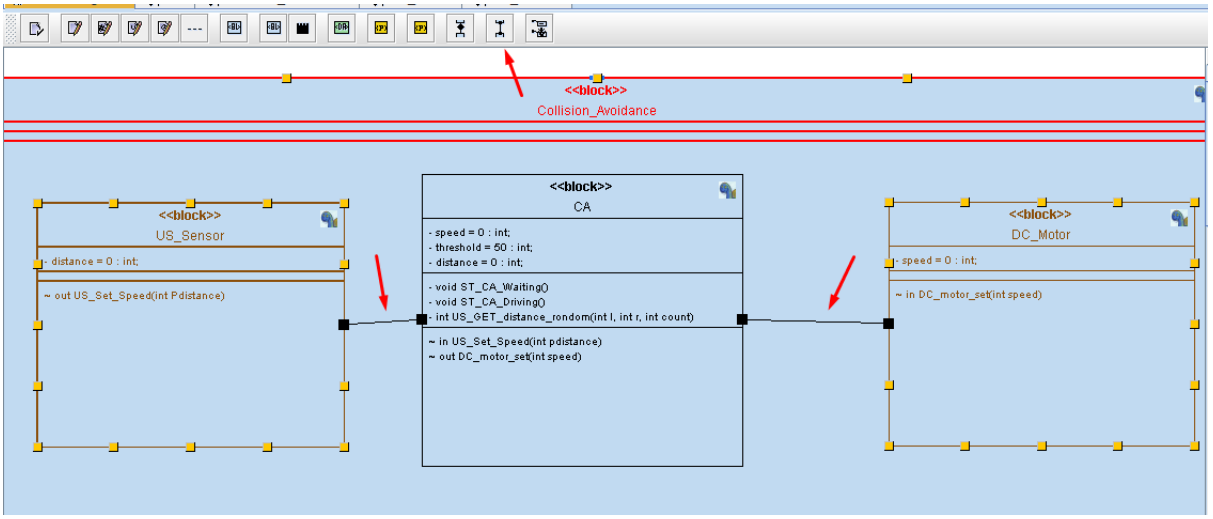
Problems | Tasks | Console | Properties | Debugger Console

<terminated> (exit value: -1,073,741,510) unit4_lesson2_CA_v1.exe [C/C++ Application] C:\U

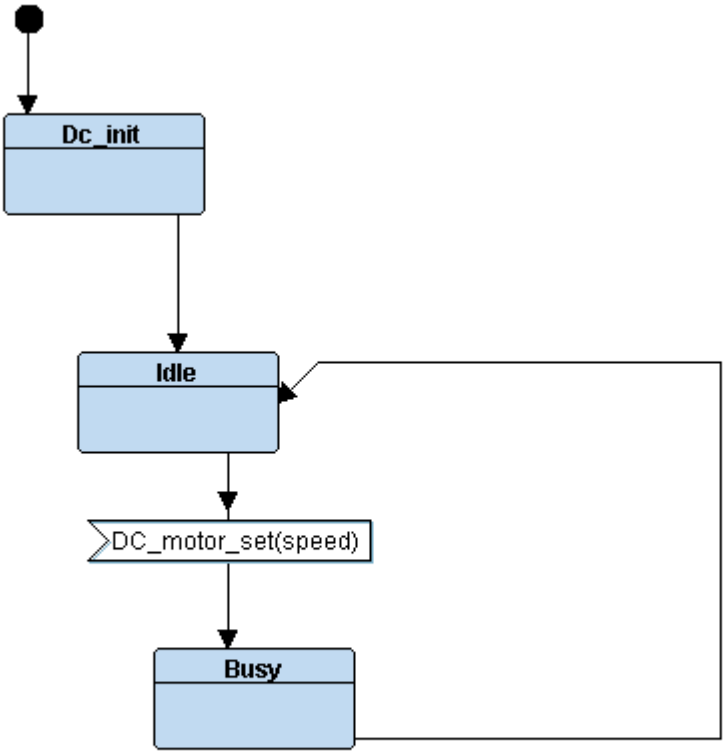
```
CA_driving State : Distance =55          Speed =0
CA_driving State : Distance =53          Speed =30
CA_driving State : Distance =46          Speed =30
CA_driving State : Distance =47          Speed =0
CA_driving State : Distance =51          Speed =0
CA_driving State : Distance =50          Speed =30
CA_driving State : Distance =55          Speed =0
CA_driving State : Distance =52          Speed =30
CA_driving State : Distance =52          Speed =30
CA_driving State : Distance =51          Speed =30
CA_driving State : Distance =51          Speed =30
CA_driving State : Distance =54          Speed =30
CA_driving State : Distance =45          Speed =30
CA_driving State : Distance =48          Speed =0
CA_driving State : Distance =51          Speed =0
CA_driving State : Distance =46          Speed =30
CA_driving State : Distance =51          Speed =0
CA_driving State : Distance =46          Speed =30
CA_driving State : Distance =48          Speed =0
CA_driving State : Distance =51          Speed =0
CA_driving State : Distance =48          Speed =30
CA_driving State : Distance =48          Speed =0
CA_driving State : Distance =49          Speed =0
CA_driving State : Distance =54          Speed =0
CA_driving State : Distance =47          Speed =30
CA_driving State : Distance =50          Speed =0
CA_driving State : Distance =48          Speed =0
CA_driving State : Distance =47          Speed =0
CA_driving State : Distance =54          Speed =0
```

Version 2 :

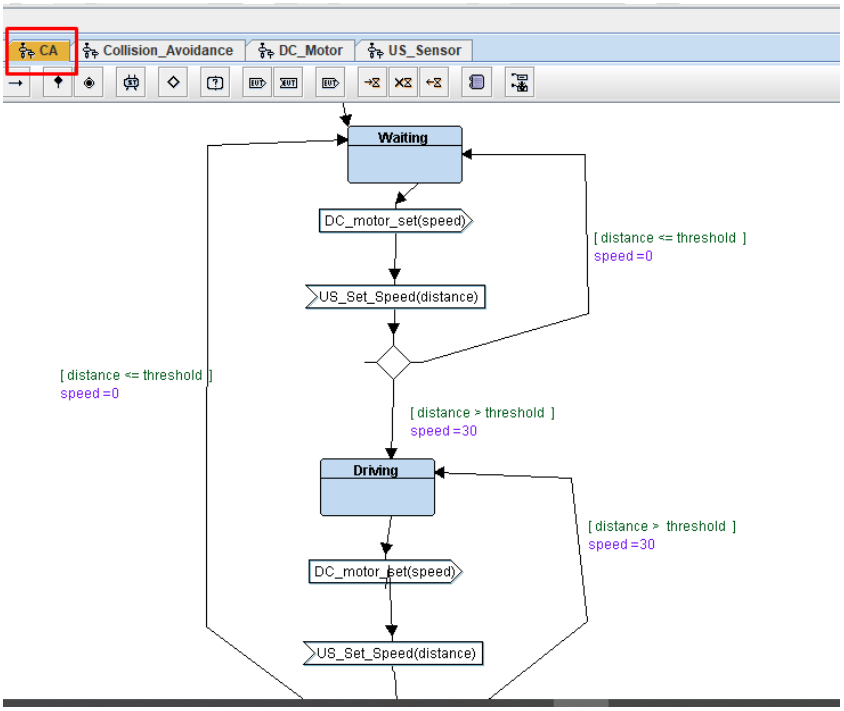
Design :



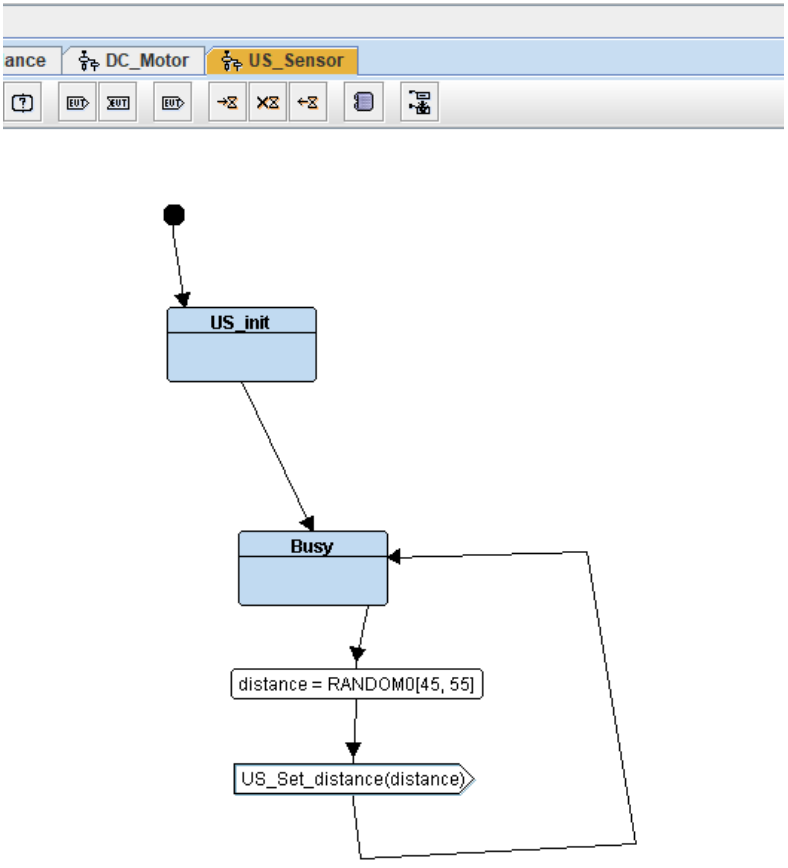
DC Motor State Diagram :



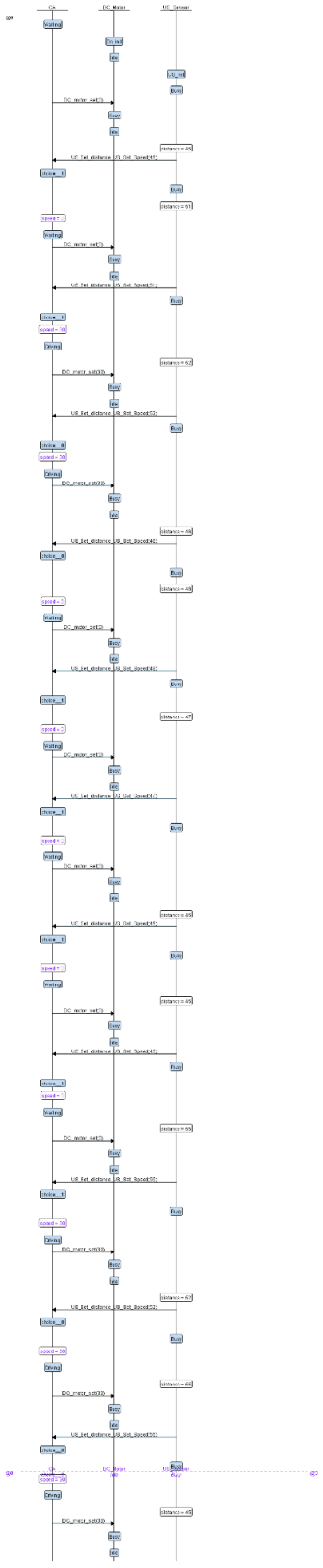
CA State Diagram :



US Sensor State Diagram :



Output of TTool :



Start Coding :

State.h File :

```
 2  * state.h
 7
 8  #ifndef STATE_H_
 9  #define STATE_H_
10  #include "stdio.h"
11  #include "stdlib.h"
12
13  //Automatic State Function Generated
14  #define STATE_define(_statFUN_) void ST_##_statFUN_()
15  #define STATE(_statFUN_)     ST_##_statFUN_
16
17
18
19
20  //States Connection
21  void US_Set_distance (int d);
22  void DC_motor (int s);
23
24  #endif /* STATE_H_ */
25
```

US.h file :

```
 2  * US.h
 7
 8  #ifndef US_H_
 9  #define US_H_
10
11  #include "state.h"
12  //Define States
13  enum {
14      US_busy,
15  }US_state_id;
16
17  //STATE Pointer to Function
18  extern void (*US_state )();
19
20  void US_init();
21
22  //Declare States Functions US
23  STATE_define(US_busy);
24  #endif /* US_H_ */
25
```

US.c file :

US.c | US.h | state.h | main.c | DC.h | DC.c | CA.h

```
10 int US_distance =0;
11
12
13
14 //STATE Pointer to Function
15 void (*US_state )();
16 int US_GET_distance_rondom(int l,int r);
17
18
19
20 void US_init()
21 {
22     //Initialize US Driver
23     printf("US init \n");
24 }
25
26 STATE_define(US_busy)
27 {
28     //State_Name
29     US_state_id=US_busy;
30
31     //State_Action
32     US_distance=US_GET_distance_rondom(45,55);
33
34     printf("\nUS_busy State : Distance =%d\n",US_distance);
35     US_Set_distance(US_distance);
36     US_state=STATE(US_busy);
37 }
38
39 int US_GET_distance_rondom(int l,int r)
40 {
41
42     int rand_num = (rand() % (r - l + 1)) + l;
43     return rand_num;
```

DC.h file :

US.c | US.h | state.h | main.c | DC.h

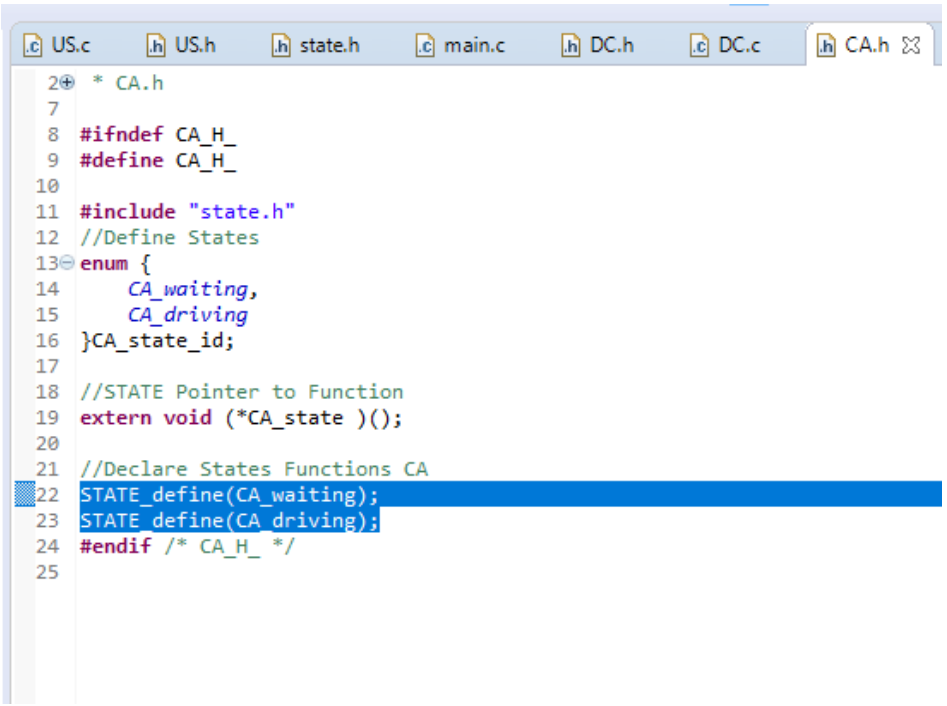
```
2  * CA.h
7
8  #ifndef DC_H_
9  #define DC_H_
10
11 #include "state.h"
12 //Define States
13 enum {
14     DC_idle,
15     DC_busy
16 }DC_state_id;
17
18 //STATE Pointer to Function
19 extern void (*DC_state )();
20
21
22
23 void DC_init();
24 //Declare States Functions CA
25 STATE_define(DC_idle);
26 STATE_define(DC_busy);
27 #endif /* DC_H_ */
28
```

DC.c file :

US.c | US.h | state.h | main.c | DC.h | DC.c | CA.h | CA.c

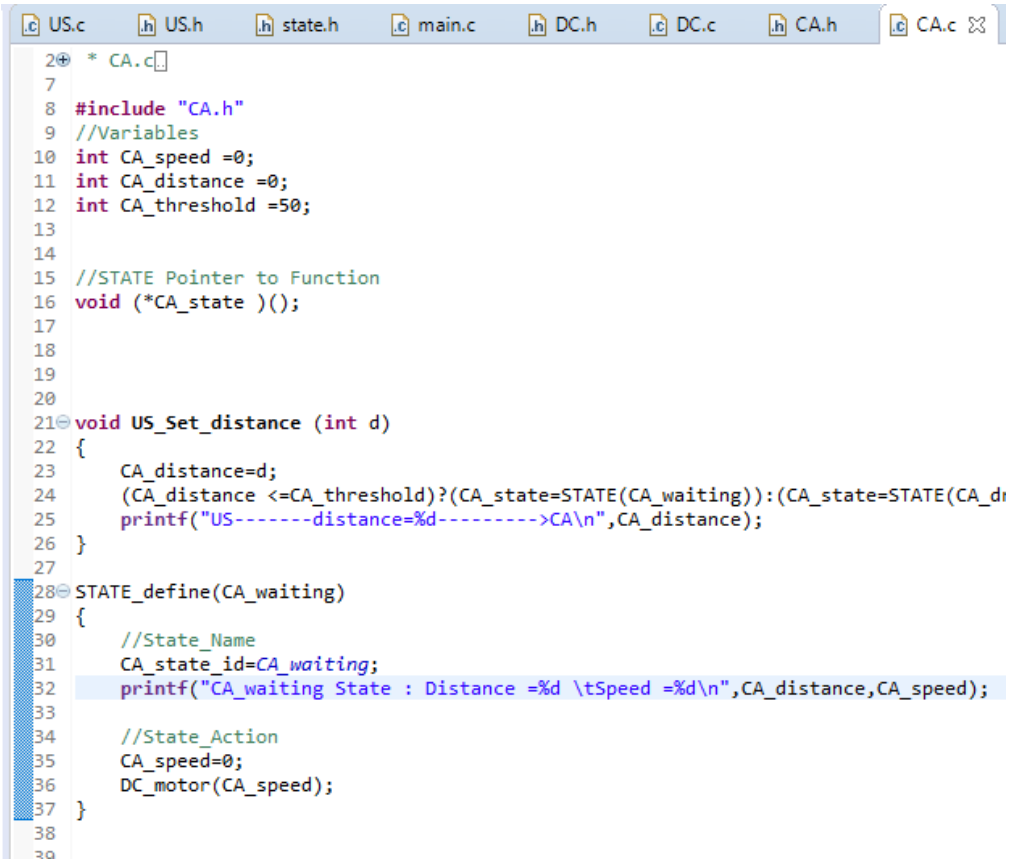
```c
/*
 * DC.c
 */

#include "DC.h"
//Variables
int DC_speed;

//STATE Pointer to Function
void (*DC_state )();


void DC_init()
{
    //init PWM
    printf("DC_init \n");

}

void DC_motor (int s)
{
    DC_speed=s;
    (DC_speed ==0)?(DC_state=STATE(DC_idle)):(DC_state=STATE(DC_busy));
    printf("CA-------speed=%d--------->DC\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

    printf("DC_busy State :Speed =%d\n",DC_speed);
    DC_state=STATE(DC_idle);

}
```

CA.h file :



```
 2  * CA.h
 7
 8  #ifndef CA_H_
 9  #define CA_H_
10
11  #include "state.h"
12  //Define States
13  enum {
14      CA_waiting,
15      CA_driving
16  }CA_state_id;
17
18  //STATE Pointer to Function
19  extern void (*CA_state )();
20
21  //Declare States Functions CA
22  STATE_define(CA_waiting);
23  STATE_define(CA_driving);
24  #endif /* CA_H_ */
25
```

CA.c file :



```
 2  * CA.c
 7
 8  #include "CA.h"
 9  //Variables
10  int CA_speed =0;
11  int CA_distance =0;
12  int CA_threshold =50;
13
14
15  //STATE Pointer to Function
16  void (*CA_state )();
17
18
19
20
21  void US_Set_distance (int d)
22  {
23      CA_distance=d;
24      (CA_distance <=CA_threshold)?(CA_state=STATE(CA_waiting)):(CA_state=STATE(CA_d
25      printf("US-------distance=%d--------->CA\n",CA_distance);
26  }
27
28  STATE_define(CA_waiting)
29  {
30      //State_Name
31      CA_state_id=CA_waiting;
32      printf("CA_waiting State : Distance =%d \tSpeed =%d\n",CA_distance,CA_speed);
33
34      //State_Action
35      CA_speed=0;
36      DC_motor(CA_speed);
37  }
38
```

```
39
40 STATE_define(CA_driving)
41 {
42     //State_Name
43     CA_state_id=CA_driving;
44     CA_speed=30;
45
46     printf("CA_driving State : Distance =%d \tSpeed =%d\n",CA_distance,CA_speed);
47     //State_Action
48
49     DC_motor(CA_speed);
50 }
51
52
```

Problems | Tasks | Console | Properties | Debugger Console

main.c file :

US.c | US.h | state.h | main.c | DC.h | DC.c

```
 2  * main.c
 7
 8 #include "CA.h"
 9 #include "US.h"
10 #include "DC.h"
11
12 void setup()
13 {
14     //init all The Drivers;
15     //inti IRQ ....
16     //init HAL US_Driver DC_Driver
17     //init Block
18     US_init();
19     DC_init();
20     //Set States Pointer for each Block
21     US_state=STATE(US_busy);
22     CA_state=STATE(CA_waiting);
23     DC_state=STATE(DC_idle);
24 }
25 int main ()
26 {
27     //volatile int i ;
28     setup();
29
30     while(1)
31     {
32         //Call State for each Block
33         US_state();
34         CA_state();
35         DC_state();
36 //      for(i=0;i<50000;i++);
37     }
38     return 0;
39 }
```

Output Of C Code :

```
US_busy State : Distance =51
US-------distance=51--------->CA
CA_driving State : Distance =51         Speed =30
CA-------speed=30--------->DC
DC_busy State :Speed =30

US_busy State : Distance =55
US-------distance=55--------->CA
CA_driving State : Distance =55         Speed =30
CA-------speed=30--------->DC
DC_busy State :Speed =30

US_busy State : Distance =51
US-------distance=51--------->CA
CA_driving State : Distance =51         Speed =30
CA-------speed=30--------->DC
DC_busy State :Speed =30

US_busy State : Distance =54
US-------distance=54--------->CA
CA_driving State : Distance =54         Speed =30
CA-------speed=30--------->DC
DC_busy State :Speed =30

US_busy State : Distance =47
US-------distance=47--------->CA
CA_waiting State : Distance =47         Speed =30
CA-------speed=0--------->DC
DC_idle State :Speed =0
```