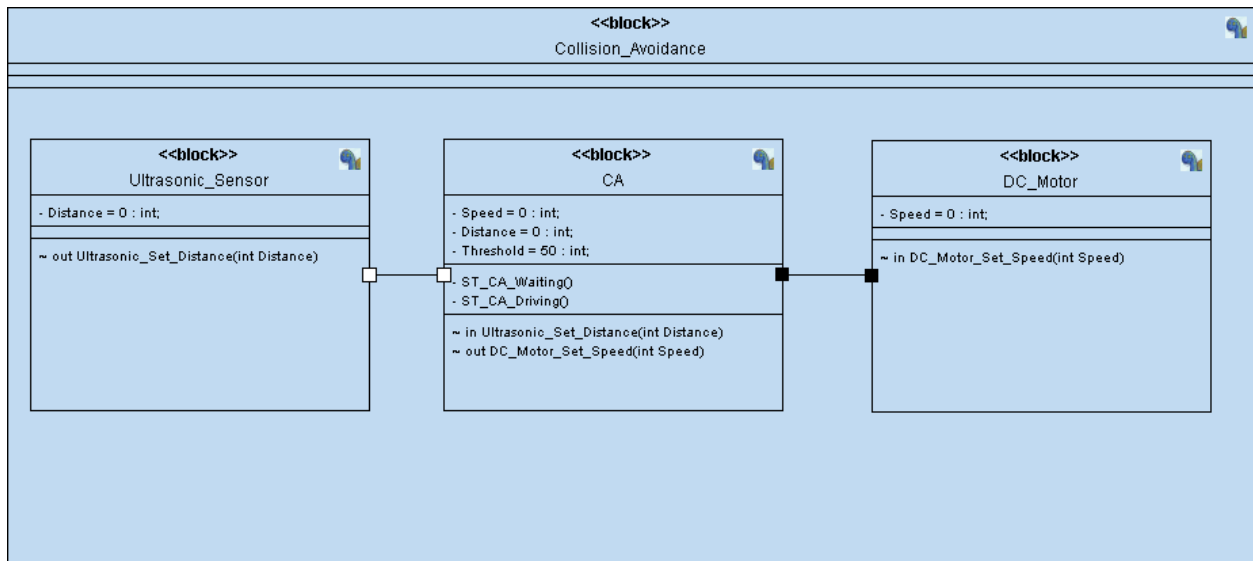


# Collision Avoidance State Machine

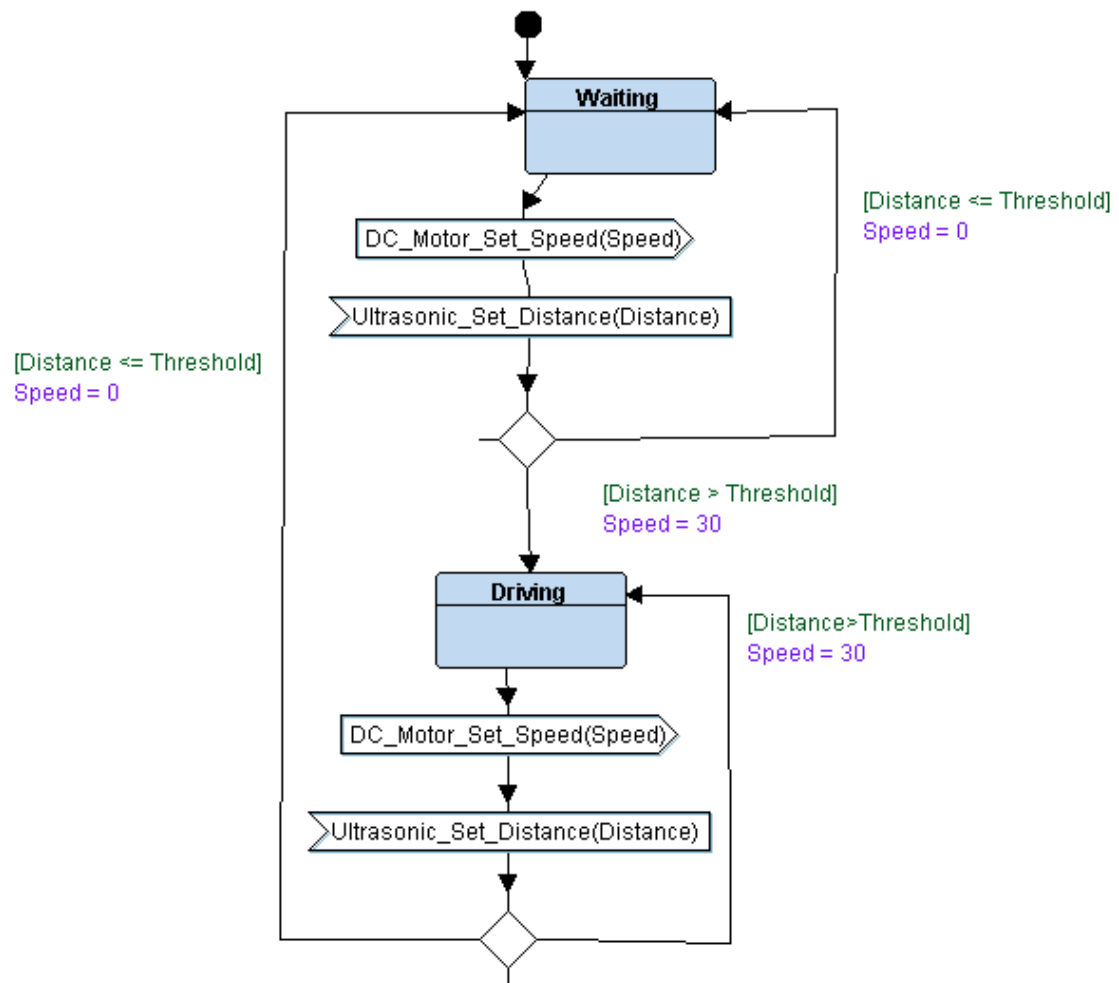
## Report by: Ebram Habib

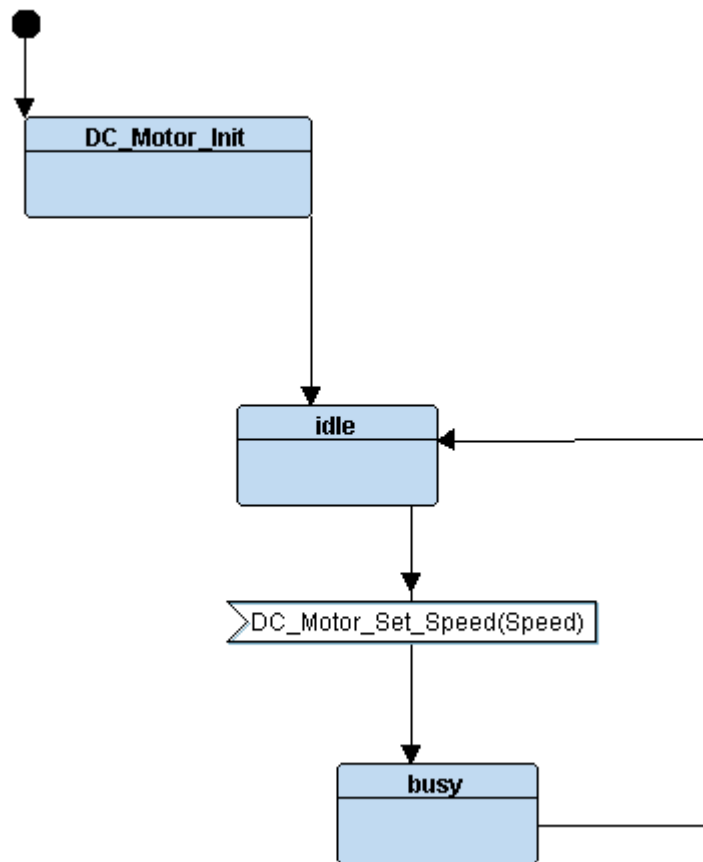
---

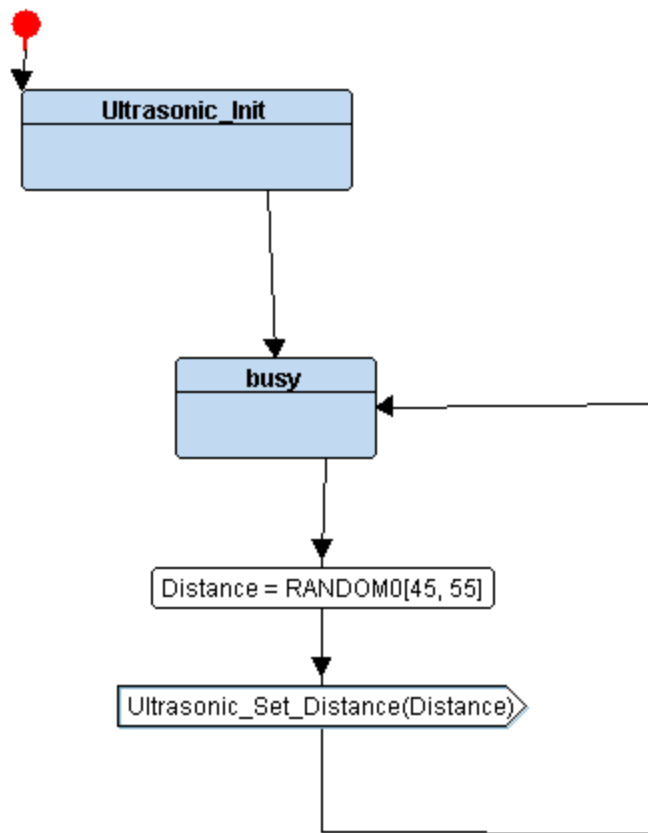
### Modules level



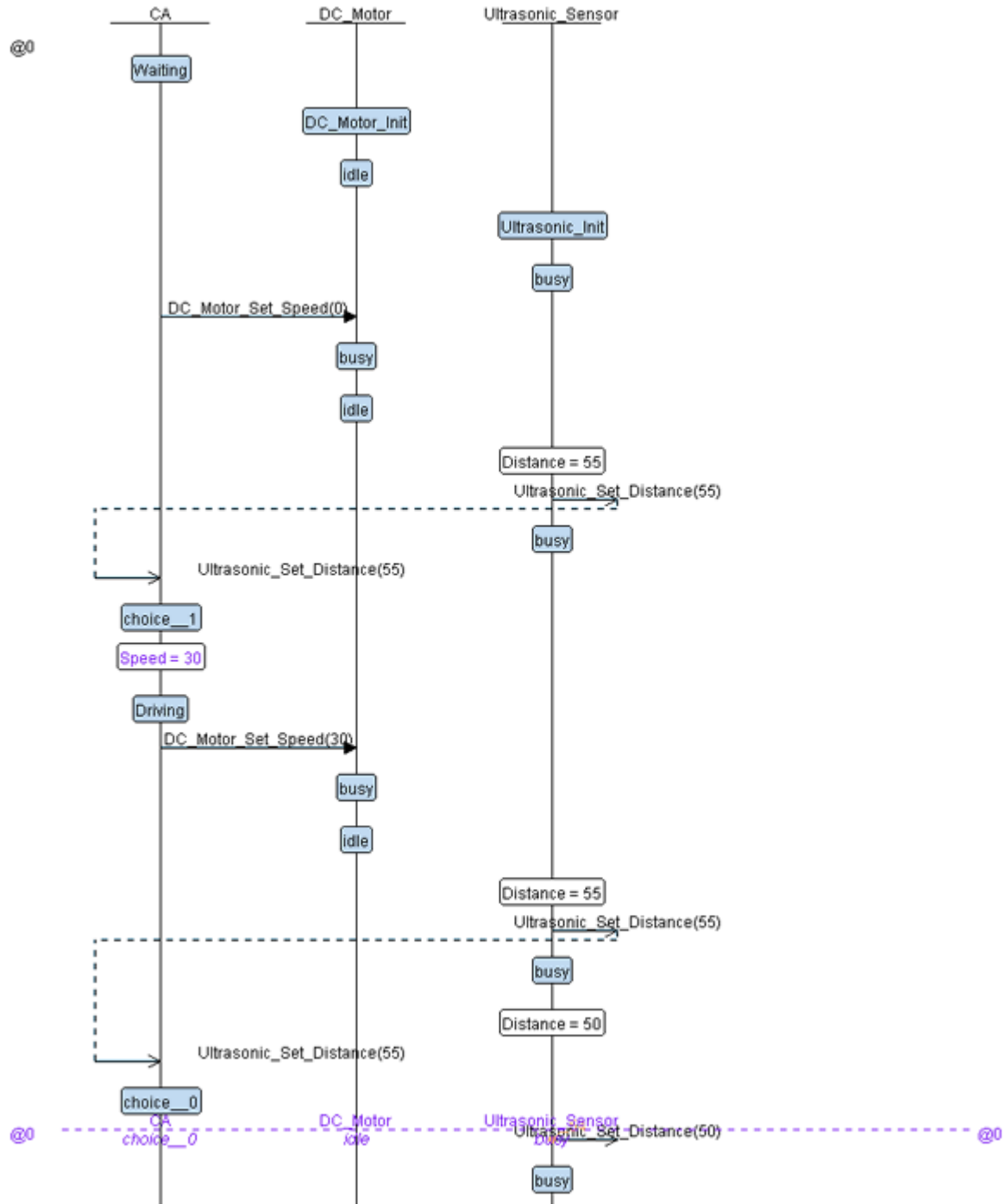
## Logical design







@0



```
0 bunchid:0 @@/ 0: start ID=3080 / top level operator
  nexts= 0:avatar transition/ ID=3111 in block CA (silent)
attributes=0 0 50
1 bunchid:0 @@/ 0: start ID=3121 / top level operator
  nexts= 0:avatar transition/ ID=3123 in block DC_Motor (silent)
attributes=0
2 bunchid:0 @@/ 0: start ID=3131 / top level operator
  nexts= 0:avatar transition/ ID=3133 in block Ultrasonic_Sensor (silent)
attributes=0
3 bunchid:0 @@/ 0: avatar transition ID=3111 / top level operator
  value:Empty transition
  nexts= 0:Waiting/ ID=3079 in block CA (silent)
attributes=0 0 50
4 bunchid:0 @@/ 0: Waiting ID=3079 / top level operator
  nexts= 0:avatar transition/ ID=3112 in block CA (silent)
attributes=0 0 50
5 bunchid:0 @@/ 0: avatar transition ID=3123 / top level operator
  value:Empty transition
  nexts= 0:DC_Motor_Init/ ID=3120 in block DC_Motor (silent)
attributes=0
6 bunchid:0 @@/ 0: DC_Motor_Init ID=3120 / top level operator
  nexts= 0:avatar transition/ ID=3124 in block DC_Motor (silent)
attributes=0
7 bunchid:0 @@/ 0: avatar transition ID=3124 / top level operator
  value:Empty transition
  nexts= 0:idle/ ID=3119 in block DC_Motor (silent)
attributes=0
8 bunchid:0 @@/ 0: idle ID=3119 / top level operator
  nexts= 0:avatar transition/ ID=3125 in block DC_Motor (silent)
attributes=0
9 bunchid:0 @@/ 0: avatar transition ID=3133 / top level operator
  value:Empty transition
  nexts= 0:Ultrasonic_Init/ ID=3130 in block Ultrasonic_Sensor (silent)
attributes=0
10 bunchid:0 @@/ 0: Ultrasonic_Init ID=3130 / top level operator
  nexts= 0:avatar transition/ ID=3134 in block Ultrasonic_Sensor (silent)
attributes=0
```

---

## C implementation

### State.h

```
1  /*
2   * State.h
3   *
4   * Created on: Nov 19, 2022
5   * Author: Ebram Habib
6   */
7
8  #ifndef STATE_H_
9  #define STATE_H_
10
11  #include "stdio.h"
12  #include "stdlib.h"
13
14  //Automatic State Function Generation
15  #define STATE_define(_stateFunc_) void ST_##_stateFunc_()
16  #define STATE(_stateFunc_) ST_##_stateFunc_
17
18  //State Connection
19  void US_Set_distance( int d);
20  void DC_motor (int s);
21
22  #endif /* STATE_H_ */
23
```

## Ultrasonic\_Sensor.h

```
1  /*
2   * Ultrasonic_Sensor.h
3   *
4   * Created on: Nov 19, 2022
5   * Author: Ebram Habib
6   */
7
8  #ifndef ULTRASONIC_SENSOR_H_
9  #define ULTRASONIC_SENSOR_H_
10
11  #include "State.h"
12
13  enum {
14      US_busy
15  }US_state_id;
16
17  //Prototypes
18  void US_init();
19  STATE_define(US_busy);
20
21  //Global Pointer to a Function
22  extern void (*US_state)();
23
24  #endif /* ULTRASONIC_SENSOR_H_ */
25
```



## Ultrasonic\_Sensor.c

```
1  /*
2   * Ultrasonic_Sensor.c
3   *
4   * Created on: Nov 19, 2022
5   * Author: Ebram Habib
6   */
7
8  #include "Ultrasonic_Sensor.h"
9
10 //Global Pointer to Function
11 void (*US_state)();
12
13 //Module Variables
14 int US_distance = 0;
15
16 int generate_random_num(int l, int r, int count)
17 {
18     //This will generate a random number in range between l and r
19
20     int i;
21     for(i = 0; i < count; i++)
22     {
23         int rand_num = (rand() % (r - l + 1)) + l;
24         return rand_num;
25     }
26 }
27
28 void US_init ()
29 {
30     //Initialize the Ultrasonic Sensor
31     //Call The Sensor's Driver or Functions
32     printf("Ultrasonic Sensor Has been Initialized \n");
33 }
34
35 STATE_define (US_busy)
36 {
37     //State Action
38     US_state_id = US_busy;
39     //Read from The Ultrasonic Sensor
40     US_distance = generate_random_num(45,55,1);
41
42     printf(" US_busy State >> Distance = %d \n", US_distance);
43
44     US_Set_distance(US_distance);
45     US_state = STATE(US_busy);
46 }
```

## Collision\_Avoidance.h

```
1  /*
2   * Collision_Avoidance.h
3   *
4   * Created on: Nov 19, 2022
5   * Author: Ebram Habib
6   */
7
8  #ifndef COLLISION_AVOIDANCE_H_
9  #define COLLISION_AVOIDANCE_H_
10
11  #include "State.h"
12
13  //Defining the States
14  enum {
15      CA_waiting,
16      CA_driving
17  }CA_state_id;
18
19  //Declare STATE Functions for CA
20  STATE_define(CA_waiting);
21  STATE_define(CA_driving);
22
23  //Global Pointer to Function
24  extern void (*CA_state)();
25
26  #endif /* COLLISION_AVOIDANCE_H_ */
27
```

## Collision\_Avoidance.c

```
1  /*
2   * Collision_Avoidance.c
3   *
4   * Created on: Nov 19, 2022
5   * Author: Ebrahim Habib
6   */
7
8  #include "State.h"
9  #include "Collision_Avoidance.h"
10
11 //Variables
12 int CA_speed = 0;
13 int CA_distance = 0;
14 int CA_threshold = 50;
15
16 //State Pointer to Function
17 void (*CA_state) ();
18
19
20 void US_Set_distance(int d)
21 {
22     CA_distance = d;
23     (CA_distance <= CA_threshold)? (CA_state = STATE(CA_waiting)) : (CA_state = STATE(CA_driving));
24     printf("US >> Distance = %d >> CA ", CA_distance);
25 }
26
27 STATE_define (CA_waiting)
28 {
29
30     //State Name
31     CA_state_id = CA_waiting;
32     printf(" CA waiting state: Distance = %d Speed = %d \n", CA_distance, CA_speed);
33
34     //State Action
35     CA_speed = 0;
36     DC_motor(CA_speed);
37 }
38
39 STATE_define (CA_driving)
40 {
41     //State Name
42     CA_state_id = CA_driving;
43     printf(" CA driving state: Distance = %d Speed = %d \n", CA_distance, CA_speed);
44
45     //State Action
46     CA_speed = 30;
47     DC_motor(CA_speed);
48 }
49
```

## DC\_Motor.h

```
1  /*
2   * DC_Motor.h
3   *
4   * Created on: Nov 19, 2022
5   * Author: Ebram Habib
6   */
7
8  #ifndef DC_MOTOR_H_
9  #define DC_MOTOR_H_
10
11  #include "State.h"
12
13  //Defining DC Motor States
14  enum {
15      DC_idle,
16      DC_busy
17  }DC_state_id;
18
19  //Prototypes
20  void DC_init();
21  STATE_define(DC_idle);
22  STATE_define(DC_busy);
23
24  //Global Pointer to Function
25  extern void (*DC_state)();
26
27  #endif /* DC_MOTOR_H_ */
28
```

## DC\_Motor.c

```
1  /*
2   * DC_Motor.c
3   *
4   * Created on: Nov 19, 2022
5   * Author: Ebrahim Habib
6   */
7
8  #include "stdio.h"
9  #include "stdlib.h"
10 #include "DC_Motor.h"
11
12 //Variables
13 unsigned int DC_speed = 0;
14
15 //Pointer to Function
16 void (*DC_state)();
17
18 void DC_init()
19 {
20     //Initialize the PWM
21     printf(" DC Motor Has Been Initialized \n");
22 }
23
24 DC_motor(int s)
25 {
26     DC_speed = s;
27     DC_state = STATE(DC_busy);
28     printf(" Sending the speed from the CA to the DC Motor ");
29 }
30
31 STATE_define(DC_idle)
32 {
33     //State Name
34     DC_state_id = DC_idle;
35
36     printf(" DC Idle State >> Speed = %d \n", DC_speed);
37 }
38
39 STATE_define(DC_busy)
40 {
41     //State Name
42     DC_state_id = DC_busy;
43
44     //State Action
45
46
47     //Calling the DC Driver
48     DC_state = STATE(DC_idle);
49
50     printf(" DC Busy State >> Speed = %d \n", DC_speed);
51 }
52
```

## Main.c

```
1  /*
2   * main.c
3   *
4   * Created on: Nov 19, 2022
5   * Author: Ebrahim Habib
6   */
7
8  #include "Collision_Avoidance.h"
9  #include "Ultrasonic_Sensor.h"
10 #include "DC_Motor.h"
11
12 void setup ()
13 {
14     //Initialize all the drivers
15     //Initialize IRQ ...
16     //Initialize HAL, US_Driver, DC_Driver
17     //Initialize BLOCK
18     US_init();
19     DC_init();
20
21     //Set States Pointers for each Function
22     CA_state = STATE(CA_waiting);
23     US_state = STATE(US_busy);
24     DC_state = STATE(DC_idle);
25 }
26
27
28 void main ()
29 {
30     volatile int d;
31
32     setup();
33
34     while(1)
35     {
36         //Call state for each block
37         US_state ();
38         CA_state ();
39         DC_state ();
40
41         //Delay
42         for(d = 0; d <= 1000; d++);
43     }
44 }
```