

Lesson 22 Avoid Obstacles Car And Follow Car function

22.1 Components used in this course

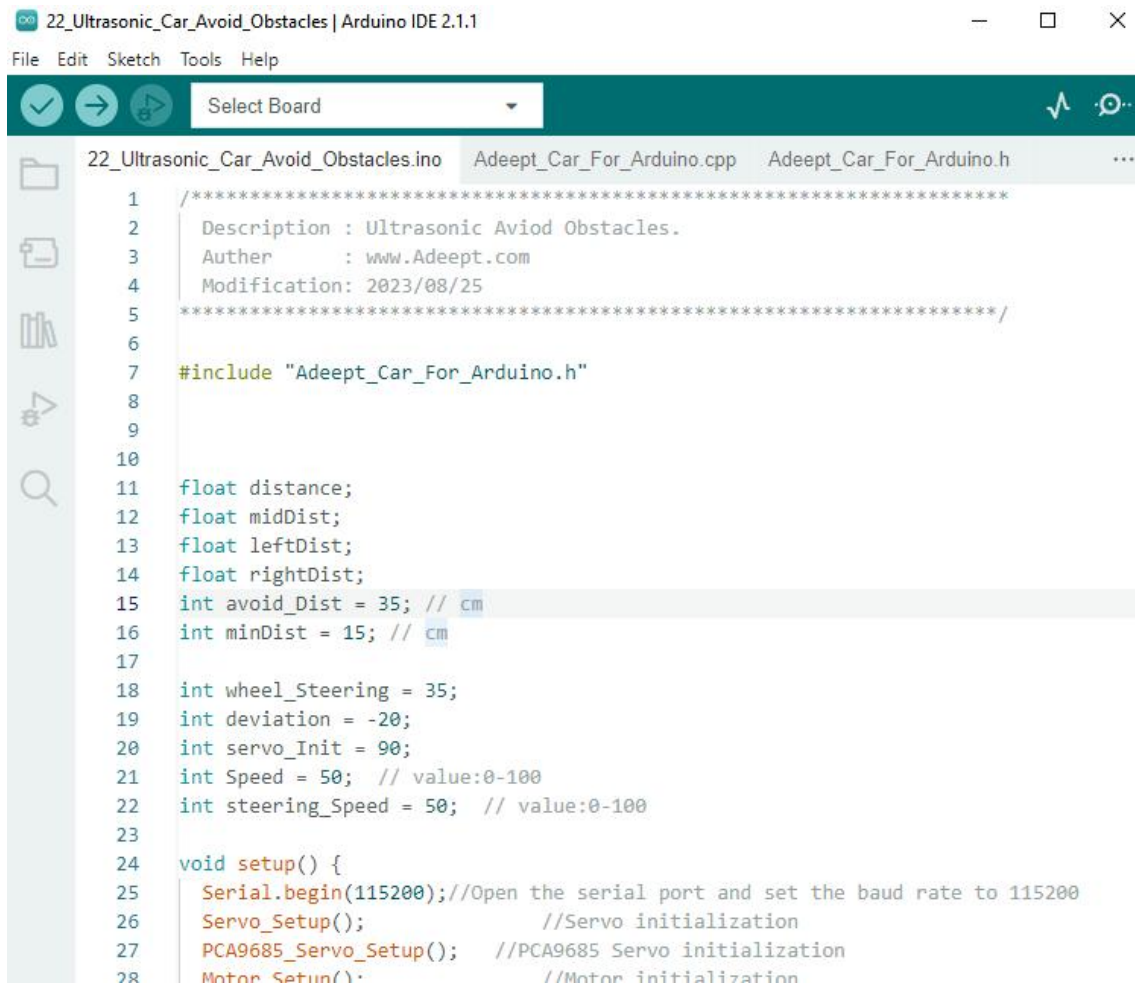
The assembled UnoCar-B Car.

22.2 Introduction the Avoid Obstacles Function

The obstacle avoidance function is realized through the ultrasonic module. When the ultrasonic module of the car detects that it is close to the obstacle, the car moves to the left and right sides.

22.3 How to use Avoid Obstacles Car

1. Before uploading the program, please lift the trolley to avoid damage to items or the trolley caused by the movement of the trolley after the upload program is completed.
2. Connect your computer and Adeept Robot Control Board with a USB cable.
3. Open "22_Ultrasonic_Car_Avoid_Obstacles" folder in "[Adeept_UnoCar-B/Code](#)", double-click "[22_Ultrasonic_Car_Avoid_Obstacles](#)".



```
22_Ultrasonic_Car_Avoid_Obstacles.ino | Arduino IDE 2.1.1
File Edit Sketch Tools Help
Select Board

22_Ultrasonic_Car_Avoid_Obstacles.ino Adeept_Car_For_Arduino.cpp Adeept_Car_For_Arduino.h ...

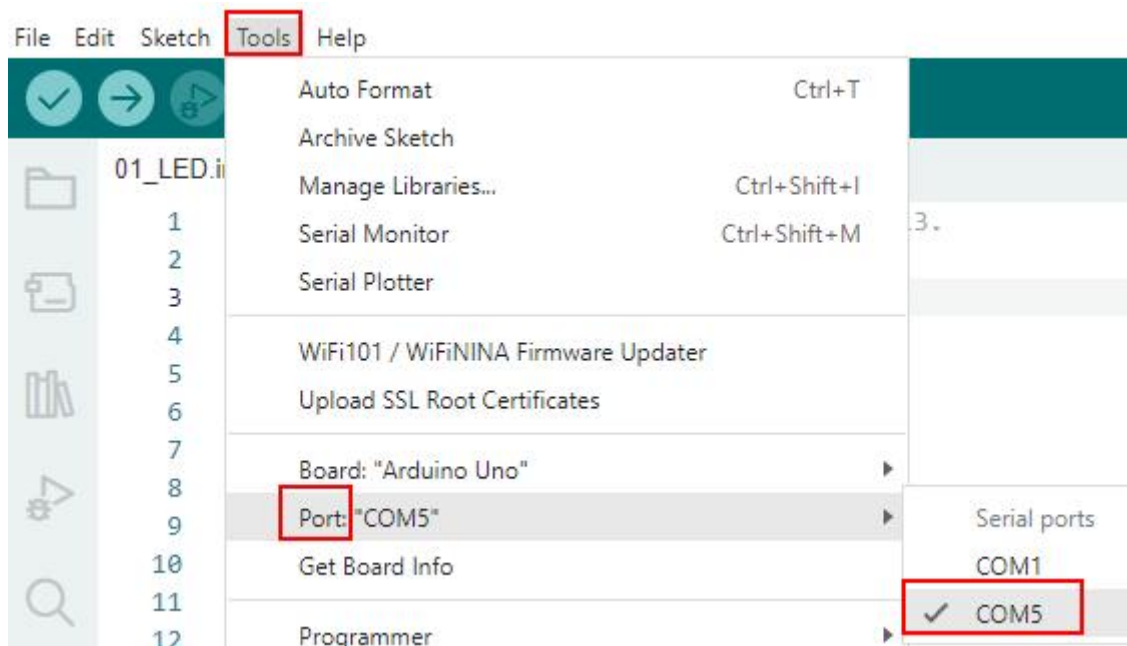
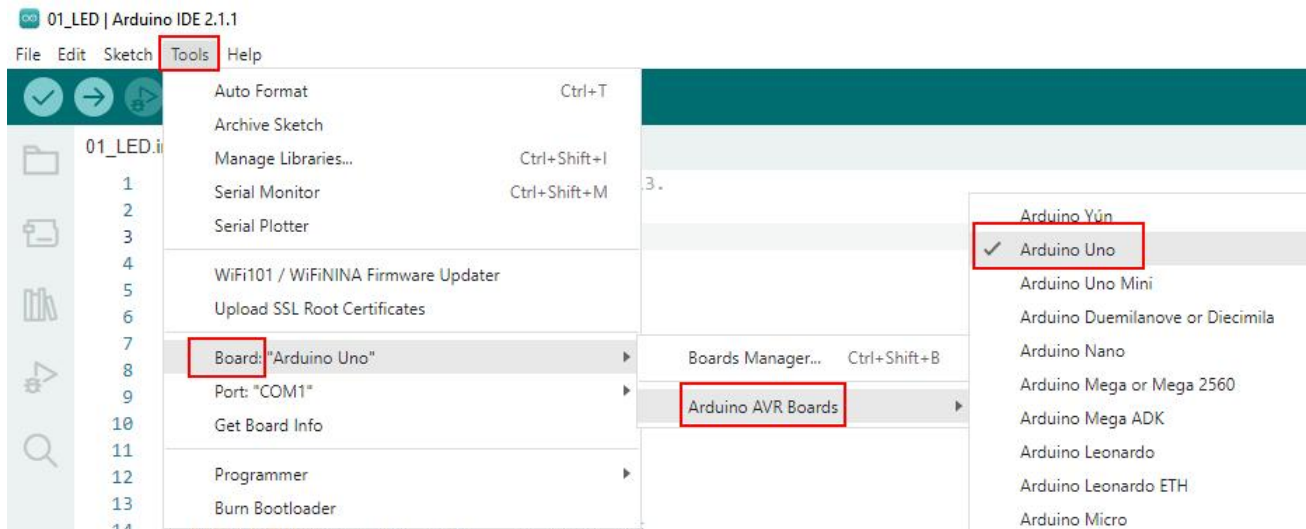
1  /*****
2   Description : Ultrasonic Avidod Obstacles.
3   Author      : www.Adeept.com
4   Modification: 2023/08/25
5   *****/
6
7  #include "Adeept_Car_For_Arduino.h"
8
9
10
11 float distance;
12 float midDist;
13 float leftDist;
14 float rightDist;
15 int avoid_Dist = 35; // cm
16 int minDist = 15; // cm
17
18 int wheel_Steering = 35;
19 int deviation = -20;
20 int servo_Init = 90;
21 int Speed = 50; // value:0-100
22 int steering_Speed = 50; // value:0-100
23
24 void setup() {
25   Serial.begin(115200); //Open the serial port and set the baud rate to 115200
26   Servo_Setup();        //Servo initialization
27   PCA9685_Servo_Setup(); //PCA9685 Servo initialization
28   Motor_Setup();        //Motor initialization
```


4. Select development board and serial port.

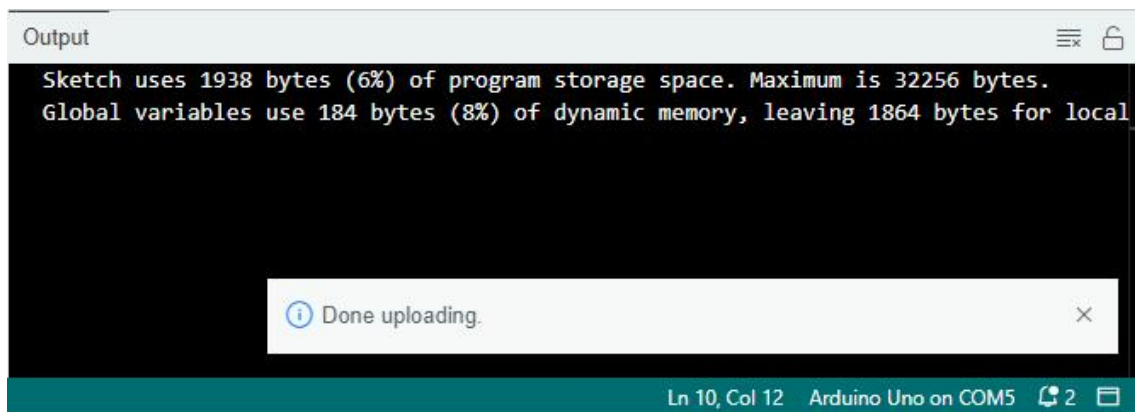
Board: Tools--->Board--->Arduino AVR Boards--->Arduino Uno

Port: Tools --->Port--->COMx

Note: The port number will be different in different computers.



5. After opening, click  to upload the code program to the Arduino. If there is no error warning in the console below, it means that the Upload is successful.

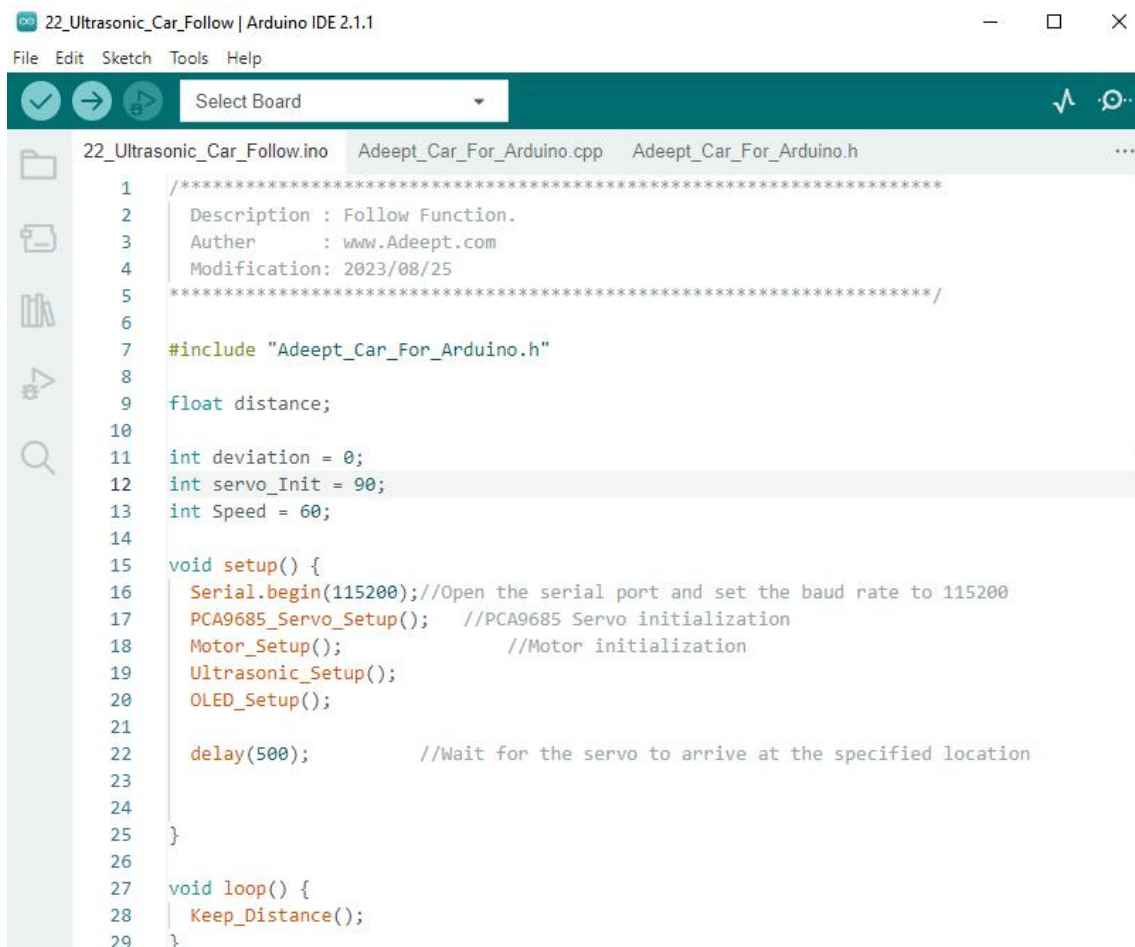


6. Now the UnoCar-B car starts to implement the ultrasonic obstacle avoidance function. Please disconnect the USB cable, use the 18650 battery to supply power, and then test the obstacle avoidance function of the car in a suitable place.

22.4 Follow Car Function

The car realizes the linear tracking function. The car will keep a distance of 30-40cm from the objects in front.

1. Open “22_Ultrasonic_Car_Follow” folder in “[Adeept_UnoCar-B/Code](#)”, double-click “22_Ultrasonic_Car_Follow”.



```
22_Ultrasonic_Car_Follow | Arduino IDE 2.1.1
File Edit Sketch Tools Help
Select Board

22_Ultrasonic_Car_Follow.ino Adeept_Car_For_Arduino.cpp Adeept_Car_For_Arduino.h

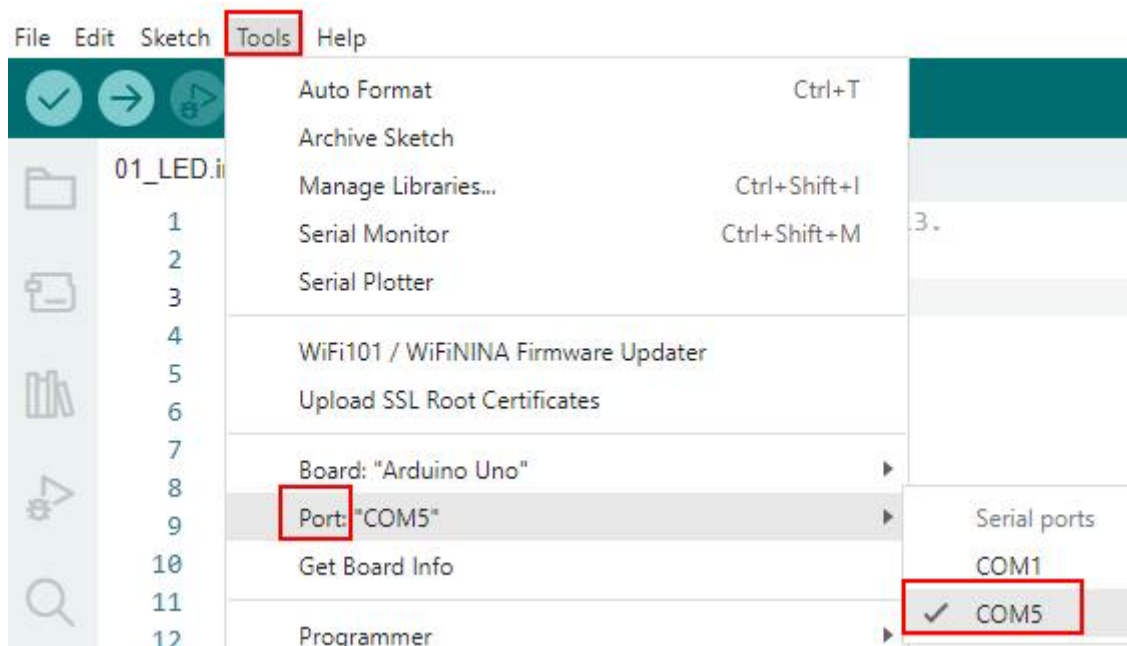
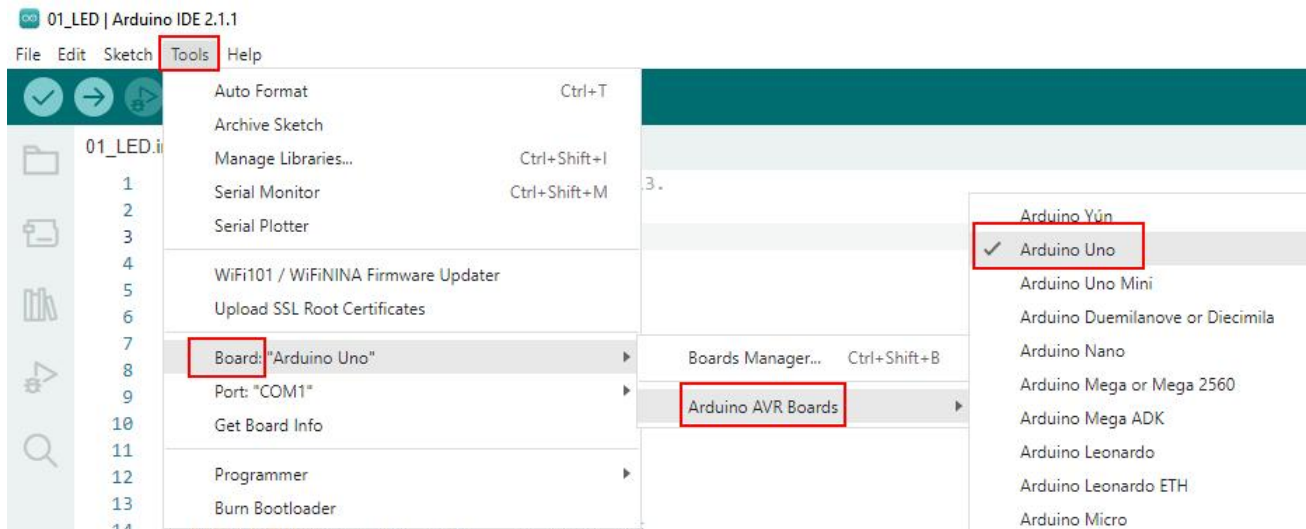
1  /*****
2   Description : Follow Function.
3   Auther      : www.Adeept.com
4   Modification: 2023/08/25
5   *****/
6
7  #include "Adeept_Car_For_Arduino.h"
8
9  float distance;
10
11 int deviation = 0;
12 int servo_Init = 90;
13 int Speed = 60;
14
15 void setup() {
16   Serial.begin(115200); //Open the serial port and set the baud rate to 115200
17   PCA9685_Servo_Setup(); //PCA9685 Servo initialization
18   Motor_Setup();         //Motor initialization
19   Ultrasonic_Setup();
20   OLED_Setup();
21
22   delay(500);             //Wait for the servo to arrive at the specified location
23
24 }
25
26
27 void loop() {
28   Keep_Distance();
29 }
```


7. Select development board and serial port.

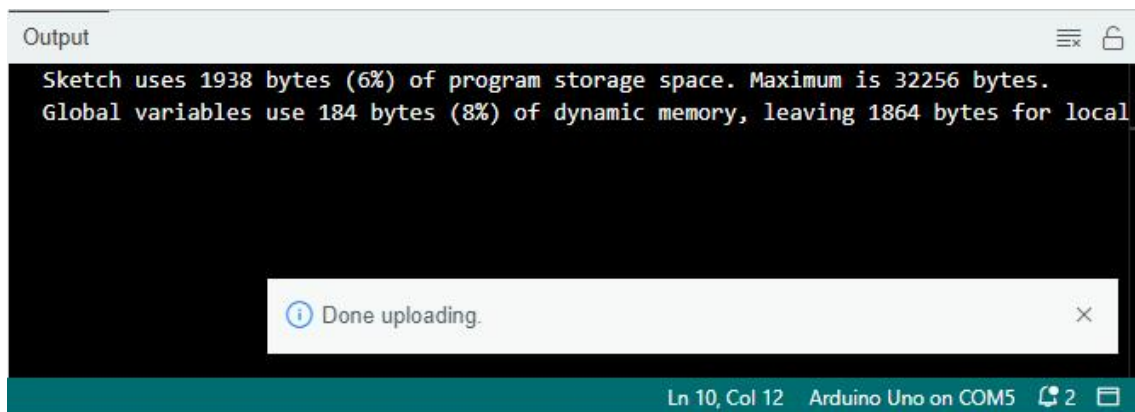
Board: Tools--->Board--->Arduino AVR Boards--->Arduino Uno

Port: Tools --->Port--->COMx

Note: The port number will be different in different computers.



8. After opening, click  to upload the code program to the Arduino. If there is no error warning in the console below, it means that the Upload is successful.



22.5 Code

```
1. #include "Adeept_Car_For_Arduino.h"
2.
3. float distance;
4. float midDist;
5. float leftDist;
6. float rightDist;
7. int avoid_Dist = 35; // cm
8. int minDist = 15; // cm
9.
10. int wheel_Steering = 35;
11. int deviation = -20;
12. int servo_Init = 90;
13. int Speed = 50; // value:0-100
14. int steering_Speed = 50; // value:0-100
15.
16. void setup() {
17.   Serial.begin(115200); //Open the serial port and set the baud rate to 115200
18.   Servo_Setup(); //Servo initialization
19.   PCA9685_Servo_Setup(); //PCA9685 Servo initialization
20.   Motor_Setup(); //Motor initialization
21.   // WS2812 OFF
22.   delay(500); //Wait for the servo to arrive at the specified location
23. }
24.
25. void loop() {
```

```
26.   Avoid_Obstacles();
27.
28. }
29.
30. void Avoid_Obstacles(){
31.   Servo_Angle(2, servo_Init + deviation);
32.   delay(80);
33.   int a = GetDistance();
34.   int b = GetDistance();
35.   int c = GetDistance();
36.   midDist = (a+b+c)/3;
37.   Serial.print("Mid:");
38.   Serial.println(midDist);
39.   // Servo_1_Angle(servo_Init); // front wheel
40.   Motor(1,1,0); //Stop the car
41.   Motor(2,1,0);
42.
43.   if (midDist > avoid_Dist){
44.     // Servo_1_Angle(servo_Init+ deviation); // front wheel
45.     Servo_Angle(1, servo_Init + deviation); // front wheel
46.     Motor(1,1,Speed); //forward
47.     Motor(2,1,Speed);
48.   }
49.   else if (midDist <= avoid_Dist){
50.     // Servo_1_Angle(servo_Init + deviation); // front wheel
51.     Servo_Angle(1, servo_Init + deviation); // front wheel
52.     Motor(1,1,0); //Stop the car
53.     Motor(2,1,0);
54.     // Servo_2_Angle(servo_Init - 60); // left distance.
55.     Servo_Angle(2, servo_Init + deviation - 60); // left distance.
56.     delay(400);
57.     int a = GetDistance();
58.     int b = GetDistance();
59.     int c = GetDistance();
60.     leftDist = (a+b+c)/3;
61.     Serial.print("Left:");
62.     Serial.println(leftDist);
63.     // Servo_2_Angle(servo_Init + 60); // right distance.
64.     Servo_Angle(2, servo_Init + deviation + 60); // right distance.
65.     delay(400);
66.     a = GetDistance();
67.     b = GetDistance();
```



```

68.      c = GetDistance();
69.      rightDist = (a+b+c)/3;
70.      Serial.print("Right:");
71.      Serial.println(rightDist);
72.      // Servo_2_Angle(servo_Init); // back to mid.
73.      Servo_Angle(2, servo_Init + deviation);    // back to mid.
74.
75.      if ((leftDist < avoid_Dist)&&(rightDist < avoid_Dist)){ // Judgment left and right.
76.          if (leftDist >= rightDist){
77.              // There are obstacles on the right backward to the left.
78.              // Servo_1_Angle(servo_Init + wheel_Steering + deviation); //turn left backward
79.              Servo_Angle(1, servo_Init + deviation + wheel_Steering);    // turn left backwa
rd
80.              Motor(1,-1,Speed); //backward
81.              Motor(2,-1,Speed);
82.              delay(500);
83.          }
84.          else{ //There are obstacles on the left.
85.              // Servo_1_Angle(servo_Init - wheel_Steering + deviation); //turn right backwar
d
86.              Servo_Angle(1, servo_Init + deviation - wheel_Steering);    // turn right backw
ard
87.              Motor(1,-1,Speed); //backward
88.              Motor(2,-1,Speed);
89.              delay(500);
90.          }
91.      }
92.      else if ((leftDist > avoid_Dist)&&(rightDist <= avoid_Dist)){
93.          if (midDist < minDist){ // Obstacle ahead
94.              // Servo_1_Angle(servo_Init+ deviation); // backward
95.              Servo_Angle(1, servo_Init + deviation);    // backward
96.              Motor(1,-1,Speed);
97.              Motor(2,-1,Speed);
98.              delay(400);
99.          }
100.         // Servo_1_Angle(servo_Init + wheel_Steering + deviation); // turn left backwar
d
101.         Servo_Angle(1, servo_Init + deviation + wheel_Steering);    // turn left backwa
rd
102.         Motor(1,-1,Speed);
103.         Motor(2,-1,Speed);

```

```

104.         delay(500);
105.     }
106.     else if ((leftDist <= avoid_Dist) &&(rightDist > avoid_Dist)){ // There are obstacles on the left.
107.         if (midDist < minDist){ // Obstacle ahead
108.             // Servo_1_Angle(servo_Init + deviation); // backward
109.             Servo_Angle(1, servo_Init + deviation); // backward
110.             Motor(1,-1,Speed);
111.             Motor(2,-1,Speed);
112.             delay(500);
113.         }
114.         // Servo_1_Angle(servo_Init - wheel_Steering + deviation); //turn right backward
115.         Servo_Angle(1, servo_Init + deviation - wheel_Steering); // turn right backward
116.         Motor(1,-1,Speed); //backward
117.         Motor(2,-1,Speed);
118.         delay(400);
119.     }
120.     else if ((leftDist >= avoid_Dist) &&( rightDist >= avoid_Dist)){
121.         if (leftDist > rightDist){ // The distance to the right is greater than the left
122.             if (midDist < minDist){
123.                 // Servo_1_Angle(servo_Init+ deviation); // backward
124.                 Servo_Angle(1, servo_Init + deviation); // backward
125.                 Motor(1,-1,Speed);
126.                 Motor(2,-1,Speed);
127.                 delay(500);
128.             }
129.             // Servo_1_Angle(servo_Init + wheel_Steering + deviation); // turn left backward
130.             Servo_Angle(1, servo_Init + deviation + wheel_Steering); // turn left backward
131.             Motor(1,-1,Speed);
132.             Motor(2,-1,Speed);
133.             delay(400);
134.         }
135.     else{
136.         if (midDist < minDist){
137.             // Servo_1_Angle(servo_Init+ deviation); // backward
138.             Servo_Angle(1, servo_Init + deviation); // backward
139.             Motor(1,-1,Speed);

```

```
140.           Motor(2,-1,Speed);
141.           delay(500);
142.       }
143. // Servo_1_Angle(servo_Init + wheel_Steering + deviation); // turn left backward
144.           Servo_Angle(1, servo_Init + deviation + wheel_Steering); // turn left backward
145.           Motor(1,-1,Speed);
146.           Motor(2,-1,Speed);
147.           delay(400);
148.       }
149.   }
150. }
151. // delay(100);
152. }
```

Follow Function

```
1. #include "Adeept_Car_For_Arduino.h"
2.
3. float distance;
4. int deviation = 0;
5. int servo_Init = 90;
6. int Speed = 60;
7.
8. void setup() {
9.     Serial.begin(115200); //Open the serial port and set the baud rate to 115200
10.    PCA9685_Servo_Setup(); //PCA9685 Servo initialization
11.    Motor_Setup(); //Motor initialization
12.    Ultrasonic_Setup();
13.    OLED_Setup();
14.    delay(500); //Wait for the servo to arrive at the specified location
15. }
16.
17. void loop() {
18.    Keep_Distance();
19. }
20.
21. void Keep_Distance(){
22.     distance = GetDistance();
23.     if (distance < 30){
24.         Servo_Angle(1, servo_Init + deviation); // front wheel
```

```
25.      Motor(1,-1,Speed); //forward
26.      Motor(2,-1,Speed);
27.      }
28.      else if (distance > 40){
29.          Servo_Angle(1, servo_Init + deviation);    // front wheel
30.          Motor(1,1,Speed); //forward
31.          Motor(2,1,Speed);
32.      }
33.      else {
34.          Motor(1,1,0); // stop
35.          Motor(2,1,0);
36.      }
37.      delay(100);
38.  }
```