

Lesson 20 Line Tracking Car Function

Reflective Optical Sensor (including Line Tracking Sensor) should be avoided using in environment with infrared interference, like sunlight. Sunlight contains a lot of invisible light such as infrared and ultraviolet.

Under environment with intense sunlight, Reflective Optical Sensor cannot work normally.

20.1 Components used in this course

The assembled UnoCar-B car.

20.2 Introduction the Line Tracking Function

The table below shows the values for all cases when the three tracking sensors detect objects of different colors. A detection of a black object or no object represents 1, and a detection of a white object represents 0. Value represents the value composed of three signals, which is used to judge the moving direction of the car. The program we provide uses the decimal value as the judgment value.

Left	Middle	Right	Value(Binary)	Value(Decimal)
0	0	0	000	0
0	0	1	001	1
0	1	0	010	2
0	1	1	011	3

1	0	0	100	4
1	0	1	101	5
1	1	0	110	6
1	1	1	111	7

20.3 Wiring diagram

Make a simple video line inspection environment for testing the Line Tracking function.

Example: Paste a piece of black tape on A4 paper. We used tape about 1.5cm wide.

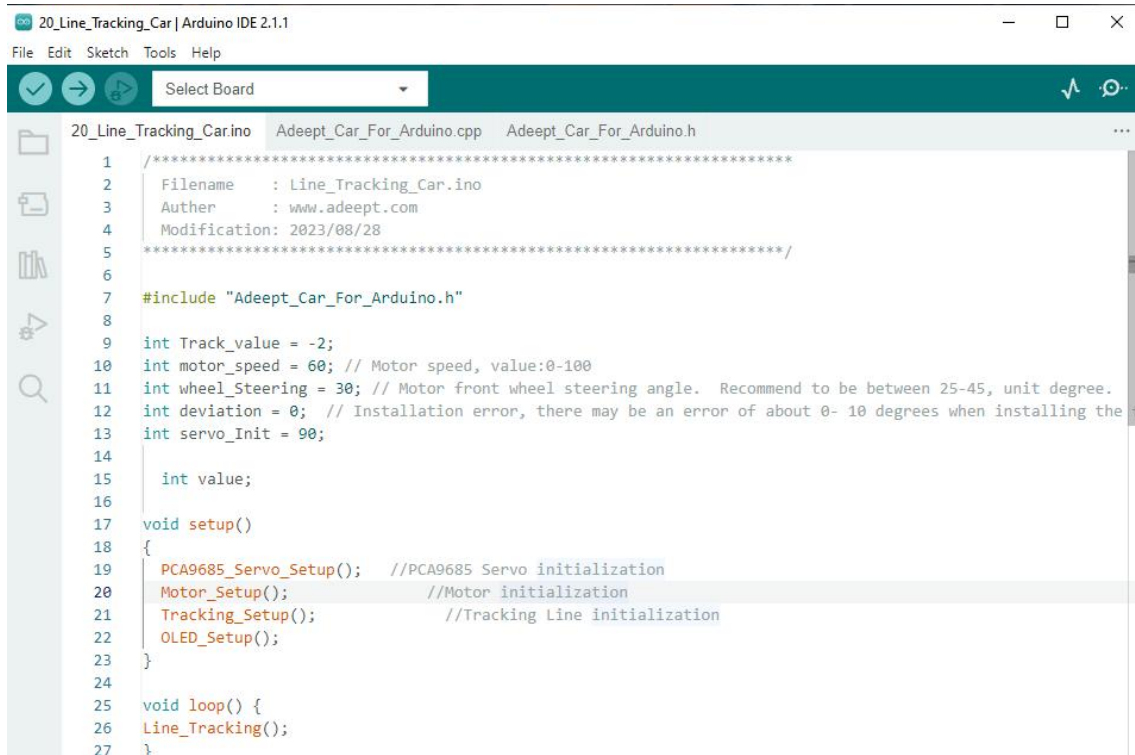




Adeept

20.4 How to control Line Tracking module

1. Connect your computer and Adeept Robot Control Board with a USB cable.
2. Open “20_Line_Tracking_Car” folder in “[Adeept_UnoCar-B/Code](#)”, double-click “[20_Line_Tracking_Car.ino](#)”.



```

1  /*****
2  Filename   : Line_Tracking_Car.ino
3  Author    : www.adeept.com
4  Modification: 2023/08/28
5  *****/
6
7  #include "Adeept_Car_For_Arduino.h"
8
9  int Track_value = -2;
10 int motor_speed = 60; // Motor speed, value:0-100
11 int wheel_Steering = 30; // Motor front wheel steering angle. Recommend to be between 25-45, unit degree.
12 int deviation = 0; // Installation error, there may be an error of about 0- 10 degrees when installing the f
13 int servo_Init = 90;
14
15 int value;
16
17 void setup()
18 {
19   PCA9685_Servo_Setup(); //PCA9685 Servo initialization
20   Motor_Setup();         //Motor initialization
21   Tracking_Setup();       //Tracking Line initialization
22   OLED_Setup();
23 }
24
25 void loop() {
26   Line_Tracking();
27 }

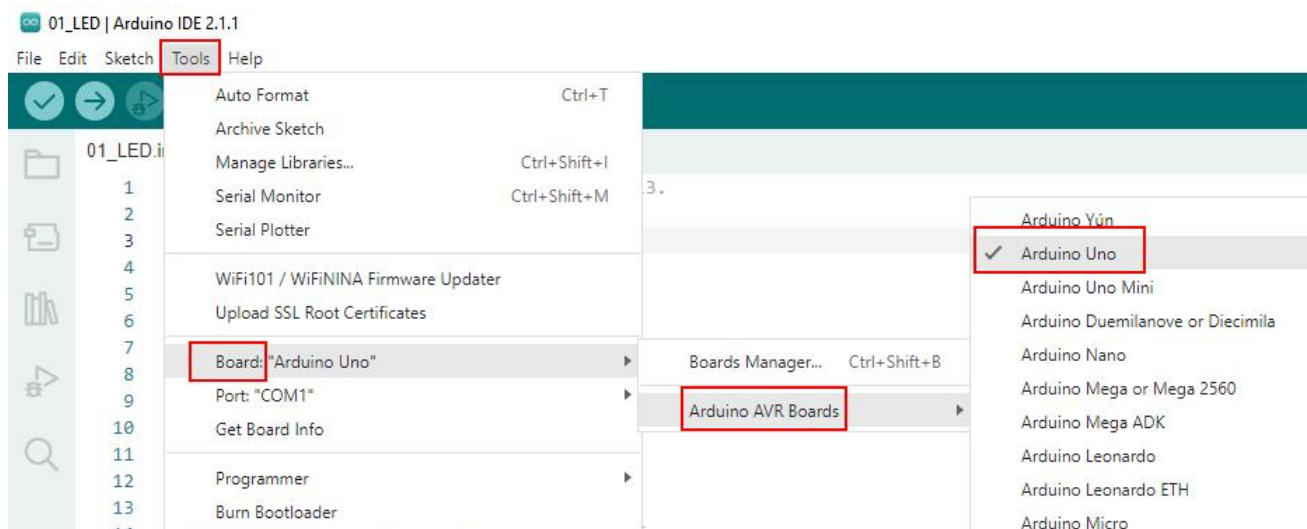
```

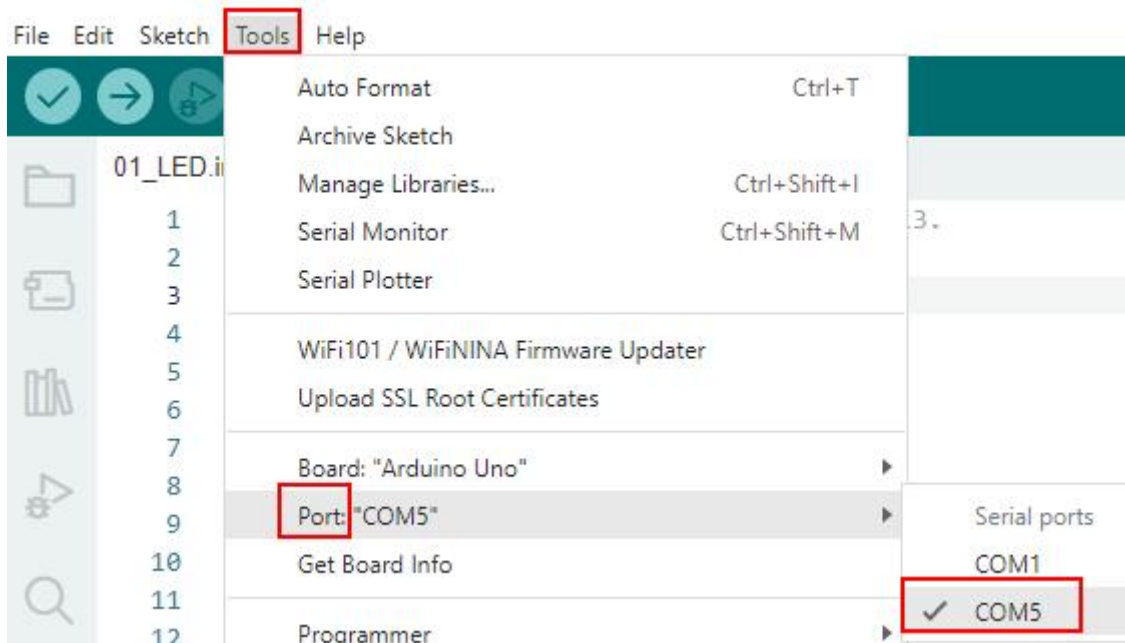
3. Select development board and serial port.


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

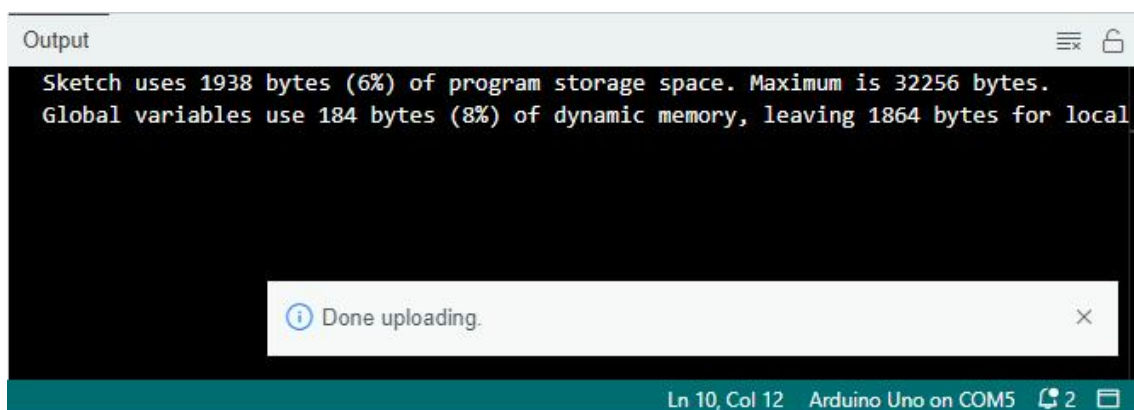
Port: Tools --->Port--->COMx

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





4. 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.



5. Place the trolley on the made test line track, and lift the rear wheel of the trolley slightly to avoid the trolley from moving. Move the front wheel of the trolley left and right to observe whether the steering of the front wheel of the trolley is consistent with the position of the test track.

If the value detected by the tracking module is abnormal, you can use tools such as a screwdriver to try to adjust the blue potentiometer of the tracking module.

After the test is completed, you can make a car tracking line. It may be necessary to adjust the car motor speed, the steering angle of the front wheel of the car and other parameters according to the actual track.

20.5 Code

```
1. #include "Adeept_Car_For_Arduino.h"
2.
3. int Track_value = -2;
4. int motor_speed = 60; // Motor speed, value:0-100
5. int wheel_Steering = 30; // Motor front wheel steering angle. Recommend to be between 25-45, unit degree.
6. int deviation = 0; // Installation error, there may be an error of about 0- 10 degrees when installing the front wheel servo of the trolley.
7. int servo_Init = 90;
8.
9. int value;
10.
11. void setup()
12. {
13.   PCA9685_Servo_Setup(); //PCA9685 Servo initialization
14.   Motor_Setup(); //Motor initialization
15.   Tracking_Setup(); //Tracking Line initialization
16.   OLED_Setup();
17. }
18.
19. void loop() {
20.   Line_Tracking();
21. }
22.
23. void Line_Tracking(){
24.   value = Track_Read(); //Read the value of the tracking module.
25.   // Serial.println(value);
```

```
26.  switch (value)
27.  {
28.      case 0:    //000 stop
29.          // Servo_1_Angle(servo_Init + deviation); // stop
30.          Servo_Angle(1, servo_Init + deviation);    // mid
31.          Motor(1, 1, 0);
32.          Motor(2, 1, 0);
33.          if (Track_value != 0){
34.              OLED_clear();
35.          }
36.          Track_value = 0;
37.          break;
38.
39.      case 1:    //010 forward
40.          // Servo_1_Angle(servo_Init+ deviation);
41.          Servo_Angle(1, servo_Init + deviation);    // mid
42.          Motor(1, 1, motor_speed); //Motor1 forward
43.          Motor(2, 1, motor_speed); //Motor2 forward
44.          if (Track_value != 2){
45.          }
46.          Track_value = 2;
47.          break;
48.
49.      case 2:    //100 Left
50.          // Servo_1_Angle(servo_Init + wheel_Steering + deviation); // left
51.          Servo_Angle(1, servo_Init + deviation + wheel_Steering);    // left
52.          Motor(1, 1, motor_speed);
53.          Motor(2, 1, motor_speed);
54.          if (Track_value != 4){
55.          }
56.          Track_value = 4;
57.          break;
58.
59.      case 3:    //110 Left
60.          // Servo_1_Angle(servo_Init + wheel_Steering + deviation); // left
61.          Servo_Angle(1, servo_Init + deviation + wheel_Steering);    // left
62.          Motor(1, 1, motor_speed);
63.          Motor(2, 1, motor_speed);
64.          if (Track_value != 6){
65.          }
66.          Track_value = 6;
67.          break;
```

```
68.
69.     case 4:    //001 right
70.         // Servo_1_Angle(servo_Init - wheel_Steering + deviation); // right
71.         Servo_Angle(1, servo_Init + deviation - wheel_Steering);    // right
72.         Motor(1, 1, motor_speed);
73.         Motor(2, 1, motor_speed);
74.         if (Track_value != 1){
75.             }
76.         Track_value = 1;
77.         break;
78.
79.     case 5:    //011 right
80.         // Servo_1_Angle(servo_Init - wheel_Steering + deviation); // right
81.         Servo_Angle(1, servo_Init + deviation - wheel_Steering);    // right
82.         Motor(1, 1, motor_speed);
83.         Motor(2, 1, motor_speed);
84.         if (Track_value != 3){
85.             }
86.         Track_value = 3;
87.         break;
88.
89.     case 6:    //111 stop
90.         // Servo_1_Angle(servo_Init + deviation); // stop
91.         Servo_Angle(1, servo_Init + deviation);    // mid
92.         Motor(1, 1, 0);
93.         Motor(2, 1, 0);
94.
95.         if (Track_value != 7){
96.             }
97.         Track_value = 7;
98.         break;
99.     case 7:    //101 forward
100.        // Servo_1_Angle(servo_Init+ deviation);
101.        Servo_Angle(1, servo_Init + deviation);    // mid
102.        Motor(1, 1, motor_speed); //Motor1 forward
103.        Motor(2, 1, motor_speed); //Motor2 forward
104.        if (Track_value != 5){
105.            }
106.        Track_value = 5;
107.        break;
108.    default:
109.        break;
```



```
110.     }  
111.     // delay(100);  
112. }
```