

Basic course ---8.Tracking

!Note:

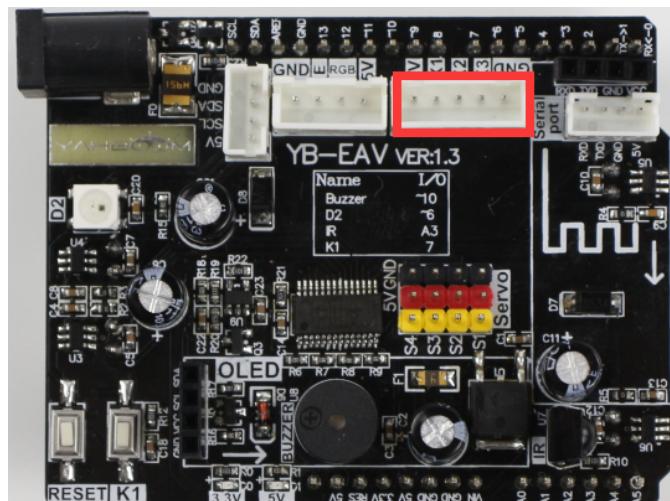
1. In order to avoid the interference of sunlight on infrared sensor, we need to carry out this experiment indoors.
2. In this experiment, you may need to modify the parameters in the program, please read the fifth part carefully.
3. The width of the black track is 1.6cm.

1. Learning goal

In this lesson, we will learn how to use tracking module.

2. Preparation

2.1 The position of the tracking sensor port on the expansion board. As shown below.



2.2 The pin of UNO board is connected the pins on the expansion board .

3.Principle of experimental

The basic principle of the tracking sensor is to use the reflective nature of the object.

Our experiment is to tracking the black line. When the infrared light is emitted to the black line, it will be absorbed by the black line. When the infrared light is emitted to the other color line, it will reflected to the infrared receiver tube.

From the hardware interface manual, we can know that ultrasonic module are driven by Pin A0,A1,A2.

Classification	Function	The number of Drive chip PCA9685	Drive Method	Connection with CPU	Uno board
Left Motor	Left front motor forward	LINB(13)	PCA9685	I2C_SDA/I2C_SCL	A4/A5
	Left front motor reverse	LINA(12)			
	Left rear motor forward	RINB(15)			
	Left rear motor reverse	RINA(14)			
Right Motor	Right front motor forward	LED10			
	Right front motor reverse	LED11			
	Right rear motor forward	LED8			
	Right rear motor reverse	LED9			
Servo	Control S1	LED0	Uno board drive directly	A0 A1 A2 12	11 7 A3 0 1 6 10
	Control S2	LED1			
	Control S3	LED2			
	Control S4	S1 (3)			
LOGO light	Control bluelight	LED7			
Tracking sensor	Left tracking sensor		Uno board drive directly	A0 A1 A2	11 7 A3 0 1 6 10
	Middle tracking sensor				
	Right tracking sensor				
Ultrasonic sensor	ultrasonic Echo Ultrasonic RGB light		Uno board drive directly	A0 A1 A2 12 11 7 A3 0 1 6 10	11 7 A3 0 1 6 10
Key	K1				
IR	IR control				
Bluetooth interface	RX TX				
On board RGB Light	RGB Light on expansion board				
Buzzer	Control buzzer				

3. About code

We need provided three codes for this experiment.

TrackingSensorTest.ino Tracking_test.ino Tracking_PID_v1.ino

4. Compiling and downloading code

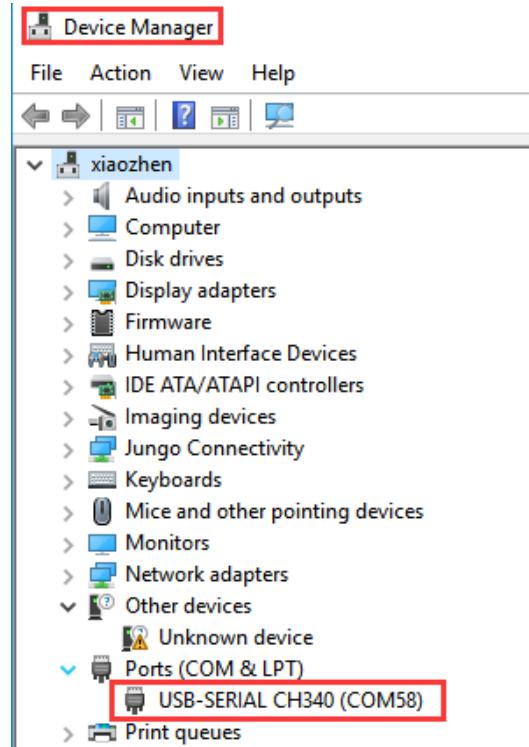
4.1 We need to open the code file by Arduino IDE software. Then click "√" under the menu bar to compile the code, and wait for the word "**Done compiling**" in the lower left corner, as shown in the figure below.

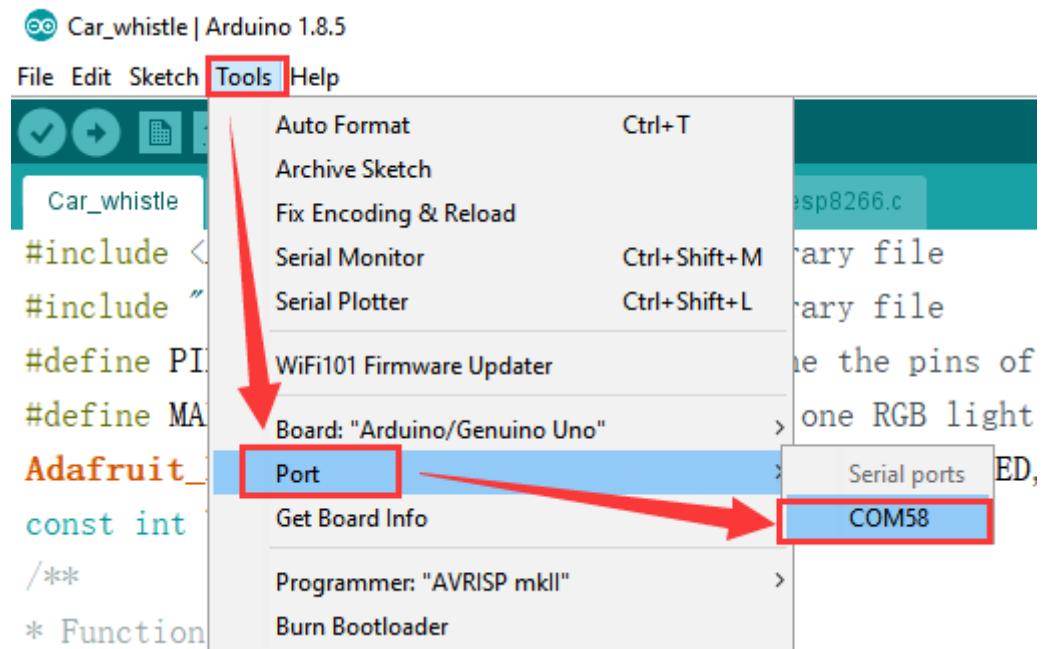
The screenshot shows the Arduino IDE interface. The title bar reads "Ultrasonic_Ranging | Arduino 1.8.5". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with various icons. The main area displays the code for "Ultrasonic_Ranging". A red box highlights the first icon in the toolbar (refresh) and the status bar at the bottom which says "Done compiling.".

```
#include <Adafruit_PWMServoDriver.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x40);
const int SingPin = 13;
float distance;

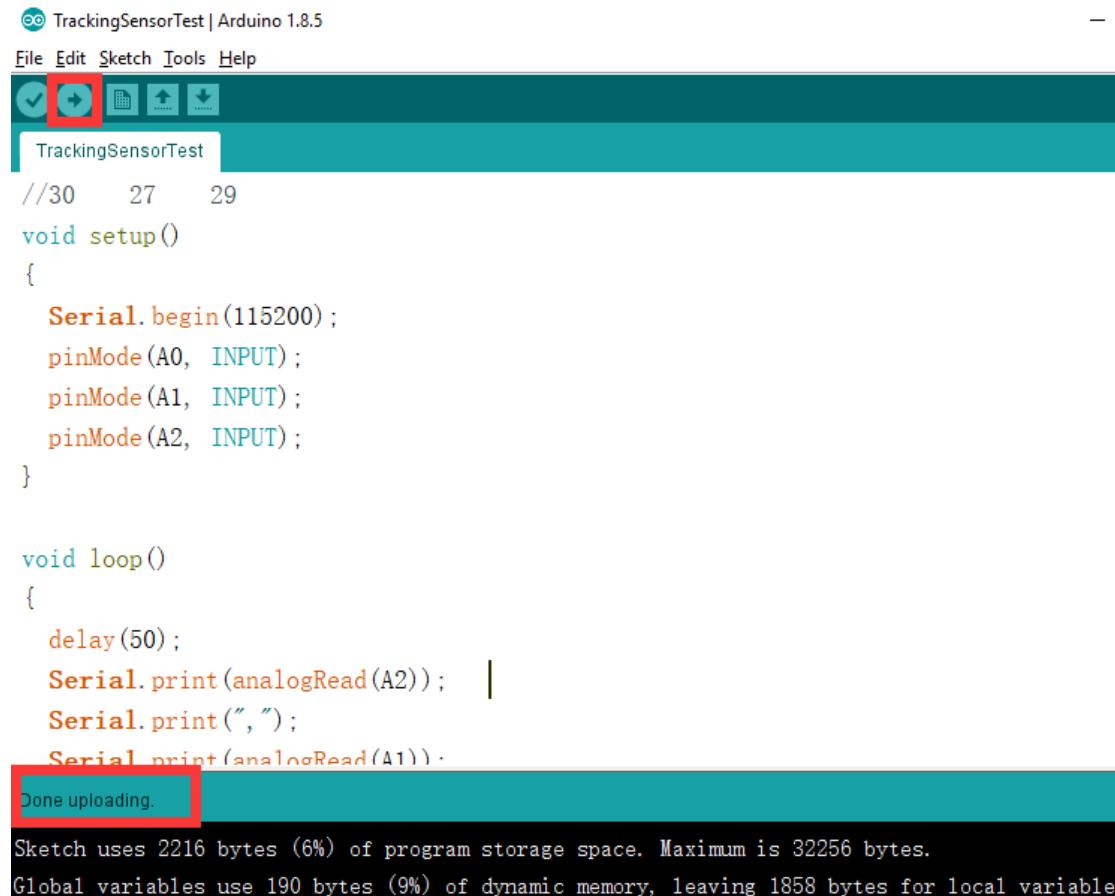
/**
 * Function      setup
 * @author        liusen
 * @date          2017. 07. 25
 * @brief         Initial configuration
 * @param[in]     void
```

4.2 In the menu bar of Arduino IDE, we need to select 【Tools】---【Port】--- selecting the port that the serial number displayed by the device manager just now, as shown in the figure below.





4.3 After the selection is completed, you need to click "→" under the menu bar to upload the code to the UNO board. When the word "**Done uploading**" appears in the lower left corner, the code has been successfully uploaded to the UNO board, as shown in the figure below.



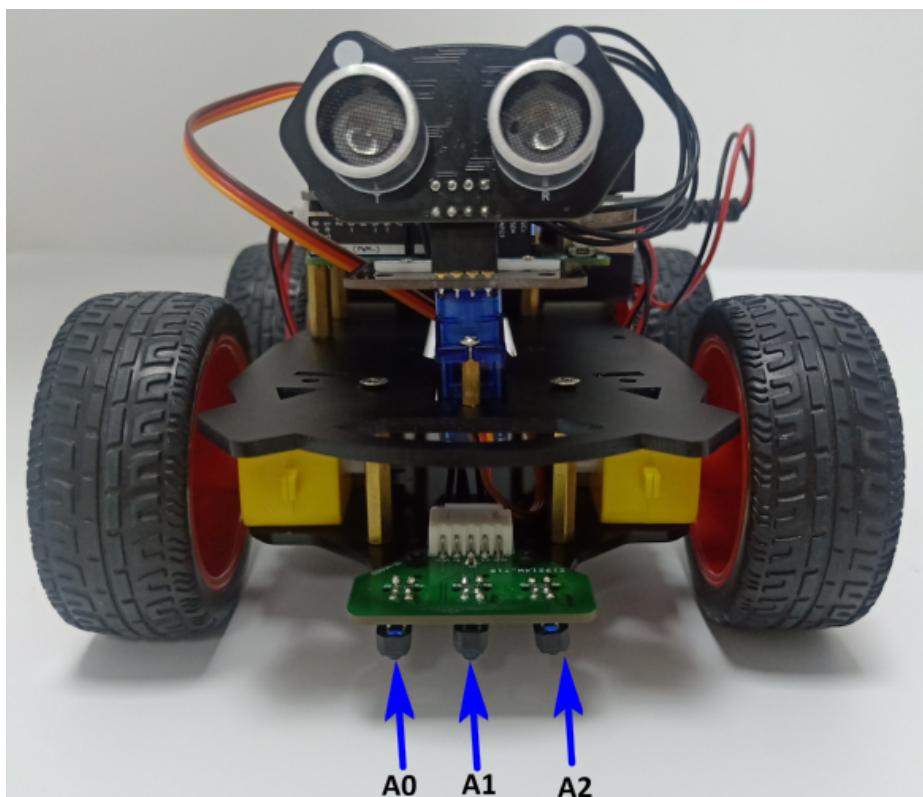
5. Experimental phenomena

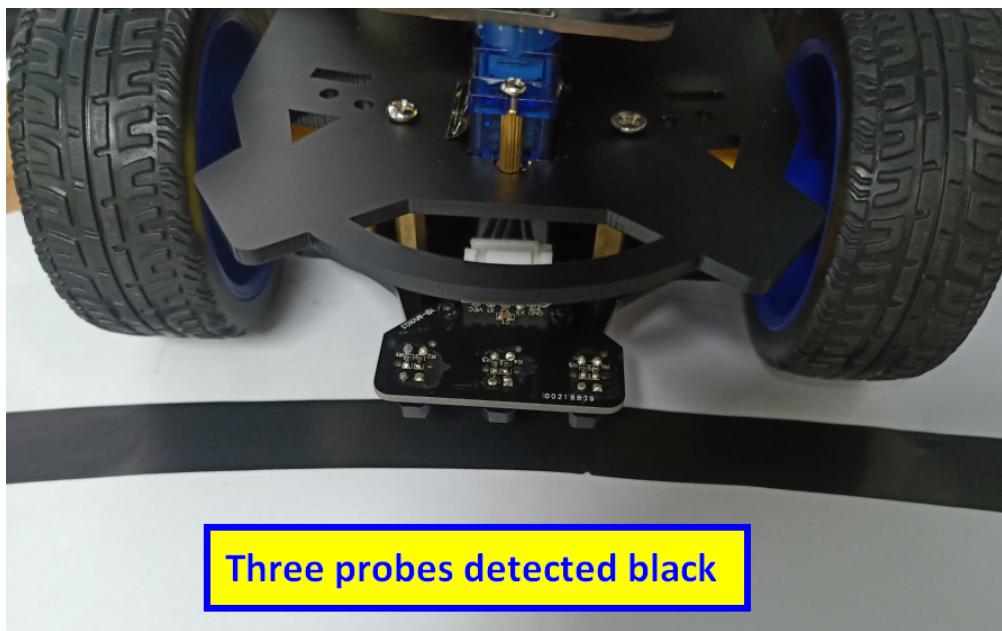
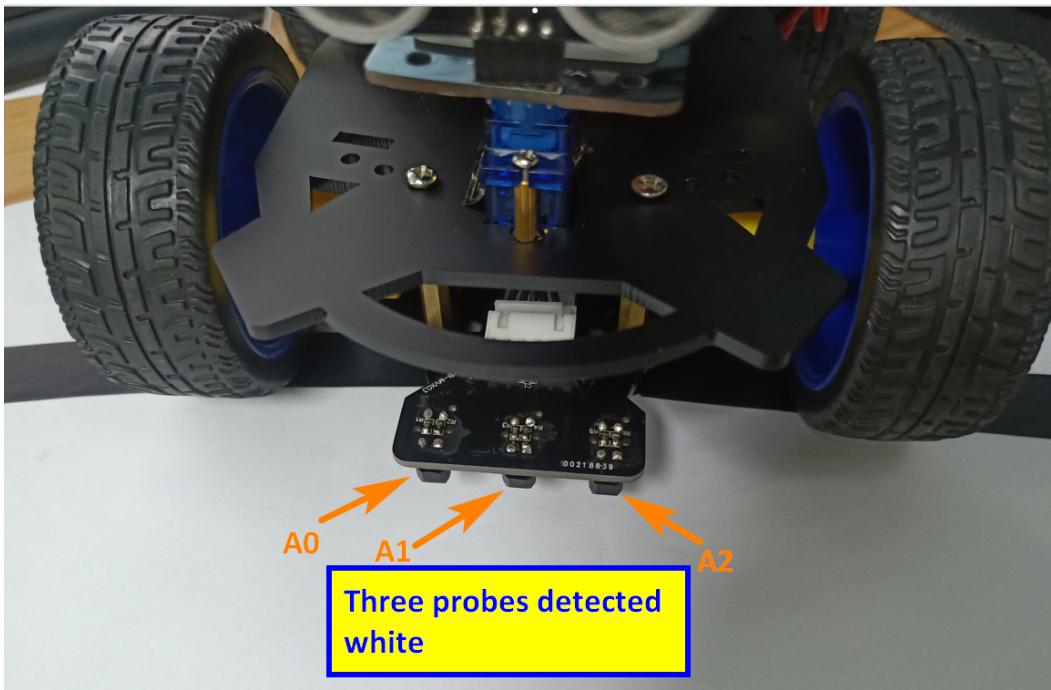
For first code: Tracking_test.ino

5.1.1 First, You need to open **TrackingSensorTest.ino** in the **TrackingSensorTest** folder. And connect the robot car with your computer by cable.

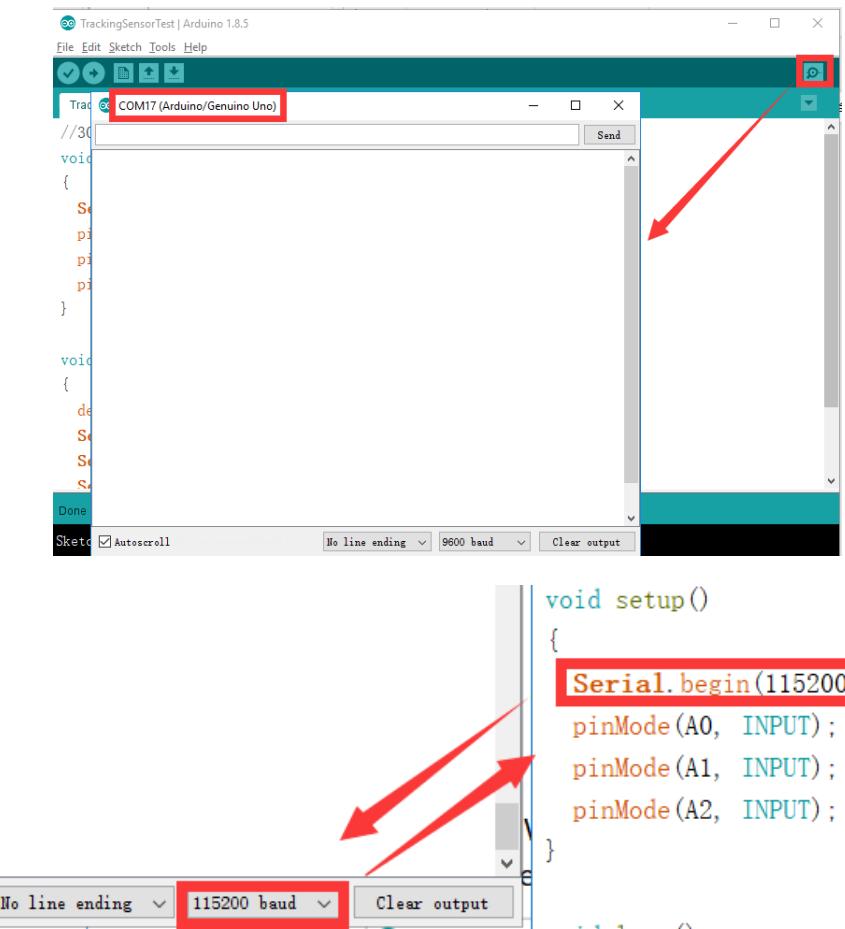
5.1.2 Then, you need to place three tracking probes on white (**must be the track where you will be conducting a tracking test**). As shown below.

5.1.3 Then, you need to place three tracking probes on black line (**must be the track where you will be conducting a tracking test**). As shown below.





5.1.4 Next, We need to open the **Serial Monitor** in the upper right corner of the Arduino IDE interface. And we need to select the same baud rate as set in the program. As shown below.



5.1.5 We can see that when the three probes of the tracking sensor detect white and detect black, the currently output analog value is printed out. As shown below.

(!!!Note: This data just for my experimental environment , you must record the value you printed, wait until the data is stable)

```
A0 A1 A2
1 158
2 
3 45, 42, 43
4 46, 42, 43
5 46, 42, 44
6 59, 48, 49
7 169, 73, 63
8 185, 91, 69
9 216, 130, 98
10 266, 185, 149
11 336, 271, 228
12 593, 572, 523
13 873, 842, 851
14 889, 828, 881
15 863, 788, 864
16 897, 855, 886
17 895, 857, 883
18 }
```

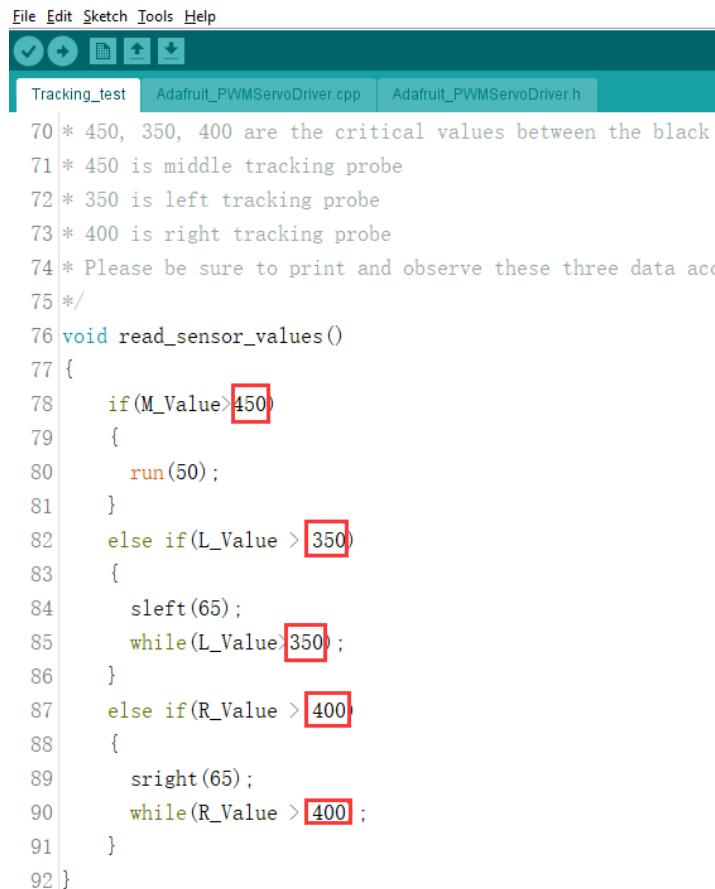
Three probes detected white

Excessive phase data

Three probes detected black

If the data shows a little fluctuation, that is normal.

5.1.6 We need to open the **Tracking_test.ino** file in the **Tracking_test** folder, and according to the value printed in the previous step, take an optimal threshold and modify the data in the program.



```

File Edit Sketch Tools Help
[File] [Edit] [Sketch] [Tools] [Help]
[Tracking_test] [Adafruit_PWMServoDriver.cpp] [Adafruit_PWMServoDriver.h]

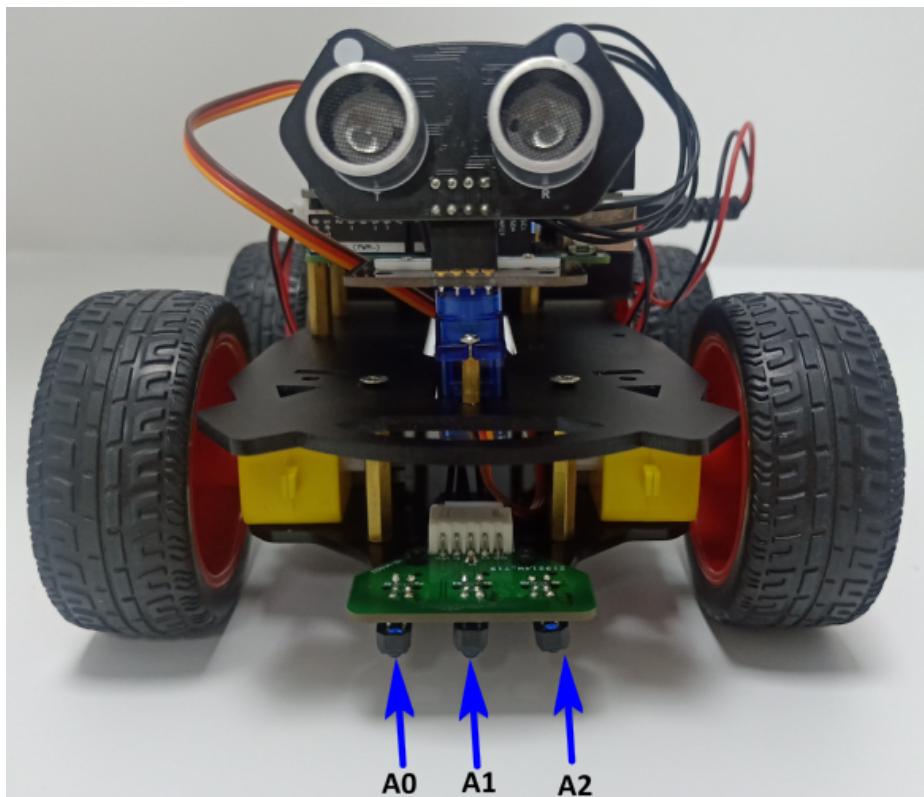
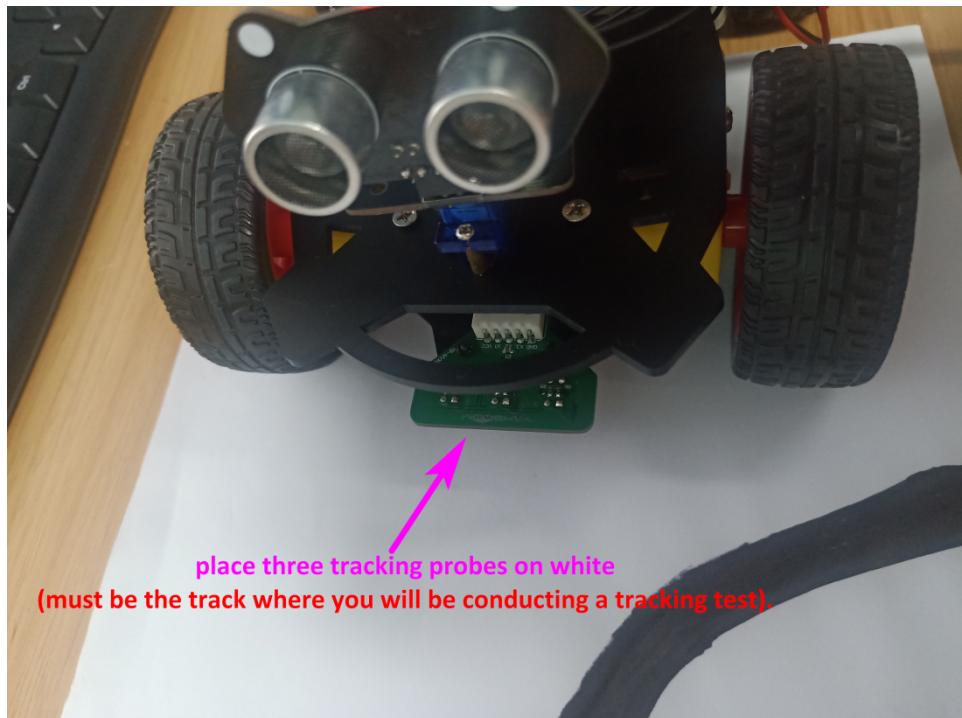
70 * 450, 350, 400 are the critical values between the black
71 * 450 is middle tracking probe
72 * 350 is left tracking probe
73 * 400 is right tracking probe
74 * Please be sure to print and observe these three data acc
75 */
76 void read_sensor_values()
77 {
78     if(M_Value>450)
79     {
80         run(50);
81     }
82     else if(L_Value > 350)
83     {
84         sleft(65);
85         while(L_Value>350);
86     }
87     else if(R_Value > 400)
88     {
89         sright(65);
90         while(R_Value > 400);
91     }
92 }
```

5.1.7 After the modification is complete, save the program and download the program to the robot car.

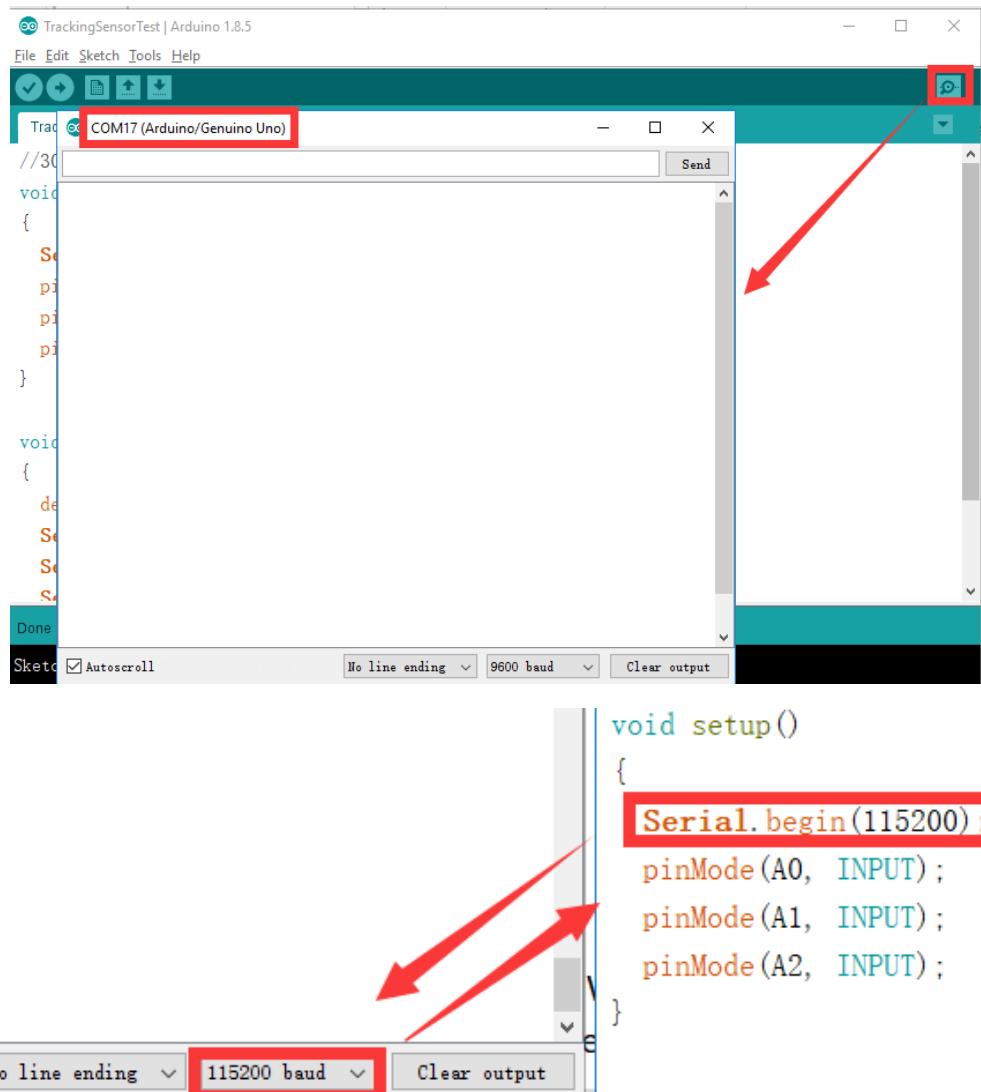
For second code: **Tracking_PID.ino**

5.2.1 First, You need to open **TrackingSensorTest.ino** in the **TrackingSensorTest** folder. And connect the robot car with your computer by cable.

5.2.2 Then, you need to place three tracking probes on white (**must be the track where you will be conducting a tracking test**). As shown below.

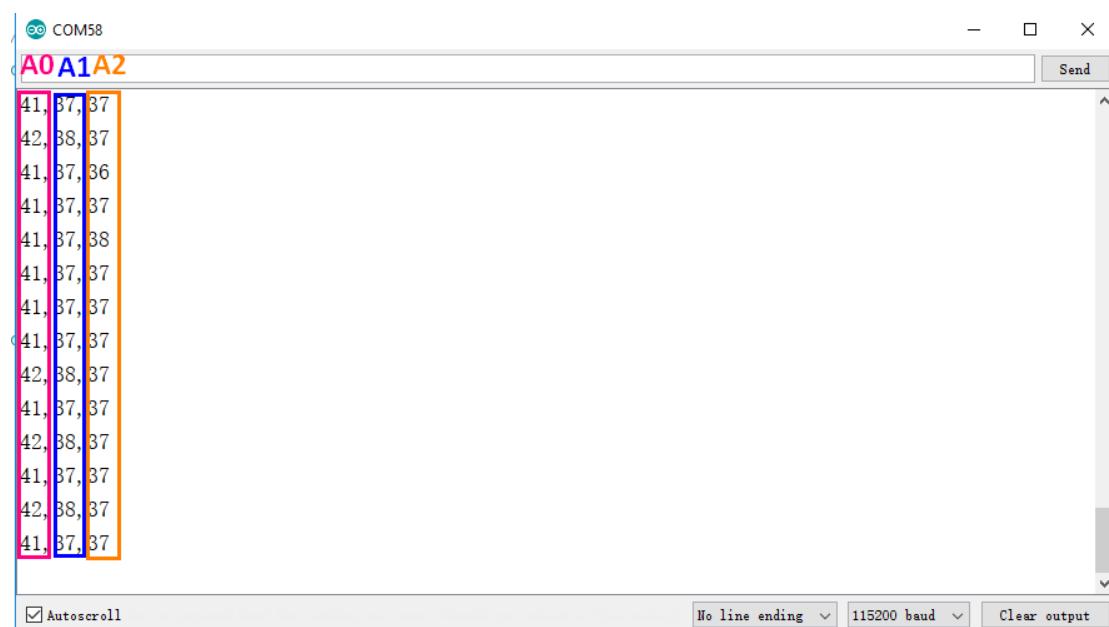


5.2.3 Next, We need to open the **Serial Monitor** in the upper right corner of the Arduino IDE interface. And we need to select the same baud rate as set in the program. As shown below.



5.2.4 We can see that when the three probes of the tracking sensor detect white, the currently output analog value is printed out. As shown below.

(!!!Note: This data just for my experimental environment , you must record the value you printed, wait until the data is stable)



If the data shows a little fluctuation, that is normal, you can take the average of the five data.

5.2.5 You need to open **Tracking_PID.ino** in the **Tracking_PID** folder. And modify the three parameters in the program.

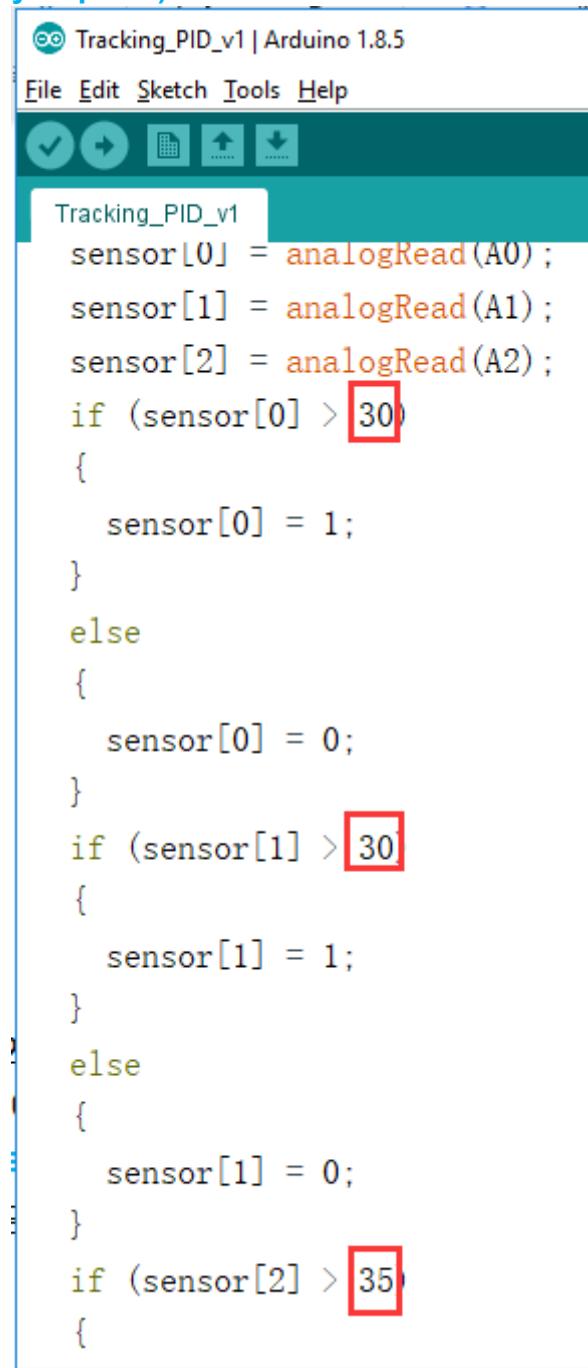
```

79 void read_sensor_values()
80 {
81   sensor[0] = analogRead(A0);
82   sensor[1] = analogRead(A1);
83   sensor[2] = analogRead(A2);
84   if (sensor[0] > 100)
85   {
86     sensor[0] = 1;
87   }
88   else
89   {
90     sensor[0] = 0;
91   }
92   if (sensor[1] > 100)
93   {
94     sensor[1] = 1;
95   }
96   else
97   {
98     sensor[1] = 0;
99   }
100  if (sensor[2] > 100)
101  {
102    sensor[2] = 1;
103  }

```

According to 5.2.4, you can see the data I printed is 41,37, 37. So the three parameters in the red coil in my program need to be changed to 100, 100, 100.

(In other words, the parameters you modify are larger than the values you print.)



```

Tracking_PID_v1 | Arduino 1.8.5
File Edit Sketch Tools Help

Tracking_PID_v1
sensor[0] = analogRead(A0);
sensor[1] = analogRead(A1);
sensor[2] = analogRead(A2);
if (sensor[0] > 30)
{
    sensor[0] = 1;
}
else
{
    sensor[0] = 0;
}
if (sensor[1] > 30)
{
    sensor[1] = 1;
}
else
{
    sensor[1] = 0;
}
if (sensor[2] > 35)
{
}

```

5.2.6 For this program, you need to modify K_p, K_i, K_d according to actual condition.

```
#include <Adafruit_PWMServoDriver.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x40);

float max = 3.85;
float s = 100;
float Kp = 37, Ki = 4, Kd = 60; |
float error = 0, P = 0, I = 0, D = 0, PID_value = 0;
```

5.2.7 After the modification is complete, save the program and download the program to the robot car.