

Lesson 7 How to use the Ultrasonic Module

In this lesson, we will learn how to use the Ultrasonic Module.

7.1 Components used in this course

Components	Quantity	Picture
Adeept Robot COntrl Board	1	
Type-C USB Cable	1	
Ultrasonic module	1	

7.2 The introduction of the Ultrasonic Module

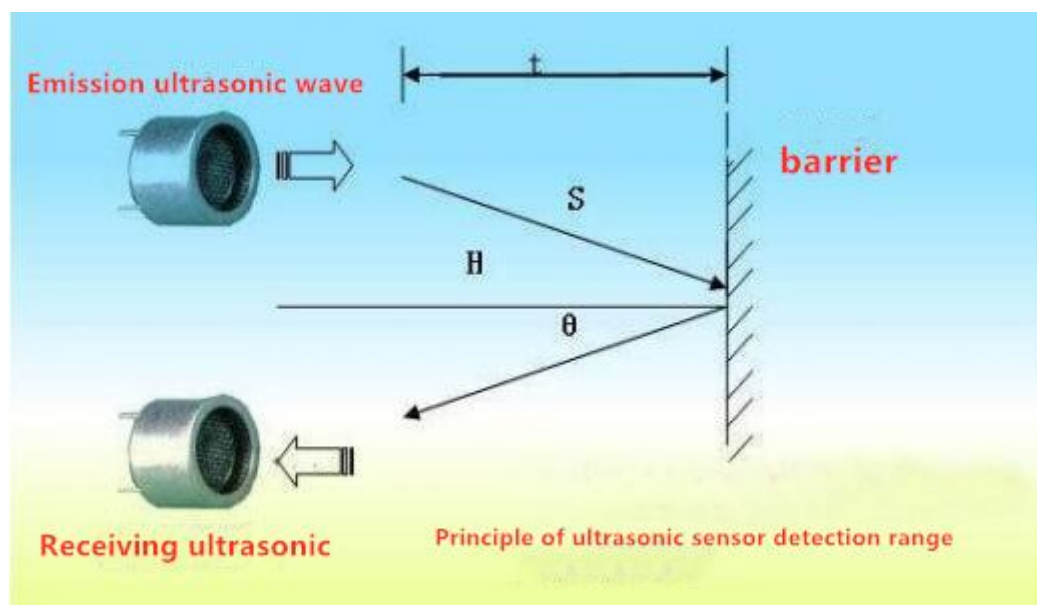
The ultrasonic module has four pins, namely VCC, GND, Echo and Trig. The HC-SR04 can provide a non-contact distance sensing function of 2cm-200cm, and the ranging accuracy can reach 3mm; The module includes an ultrasonic transmitter, receiver and control circuit. The basic working principle is as follows:

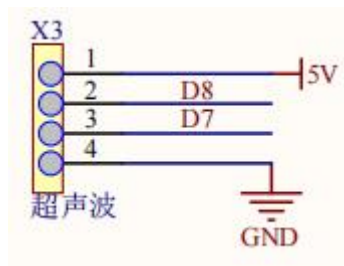
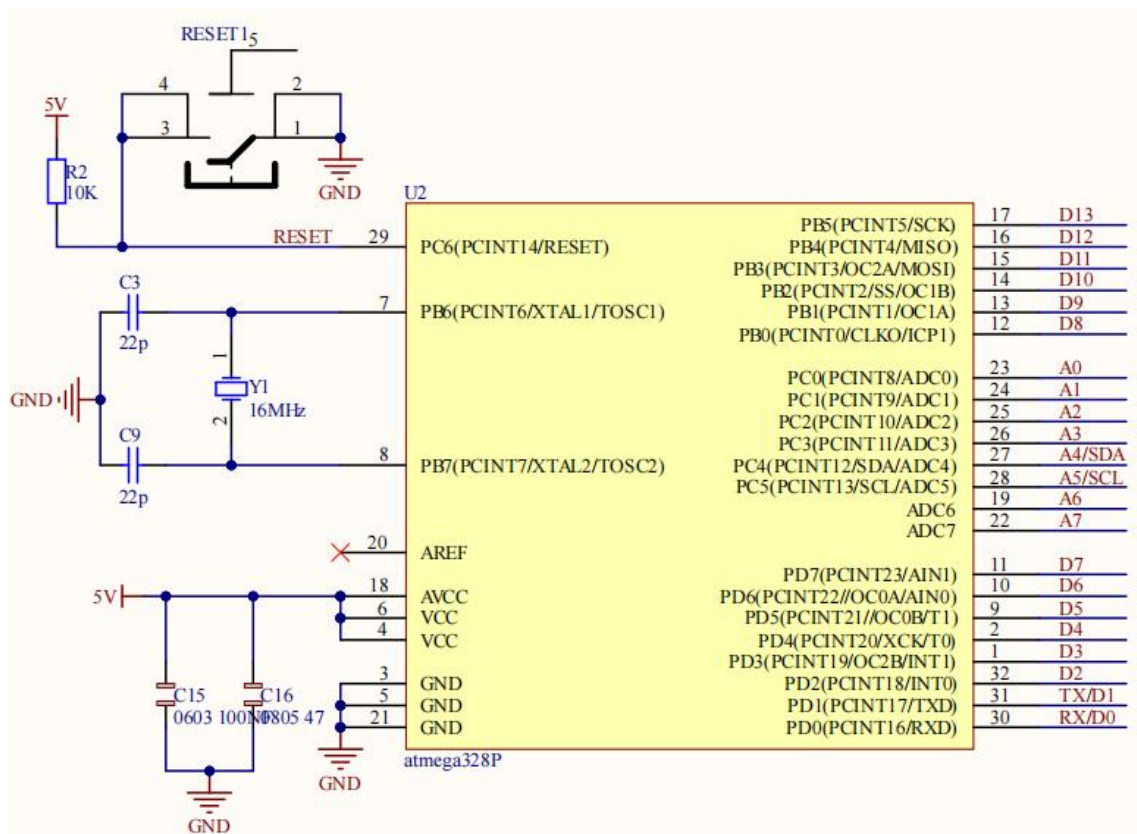
Use IO port TRIG to trigger distance measurement, and give a high level signal of at least 10us.

The module automatically sends eight 40khz square waves, and automatically detects whether there is a signal return.

There is a signal return, and a high level is output with the IO port ECHO. The duration of the high level is the time from emission to return of the ultrasonic wave.

The principle of distance detection by ultrasonic ranging sensor: the method of detecting distance by ultrasonic is called echo detection method, that is, the ultrasonic transmitter emits ultrasonic waves in a certain direction, and the timer starts timing at the same time as the launch time. The ultrasonic waves propagate in the air and encounter obstacles on the way. When the object surface (object) is blocked, it will be reflected back immediately, and the ultrasonic receiver will immediately stop timing when the reflected ultrasonic wave is received. The propagation speed of ultrasonic waves in the air is 340m/s. According to the time t recorded by the timer, the distance s from the launch point to the obstacle surface can be calculated, namely: $s=340t/2$. Using this principle of ultrasound, the ultrasonic ranging module is widely used in practical applications, such as car reversing radar, unmanned aerial vehicle, and smart car.



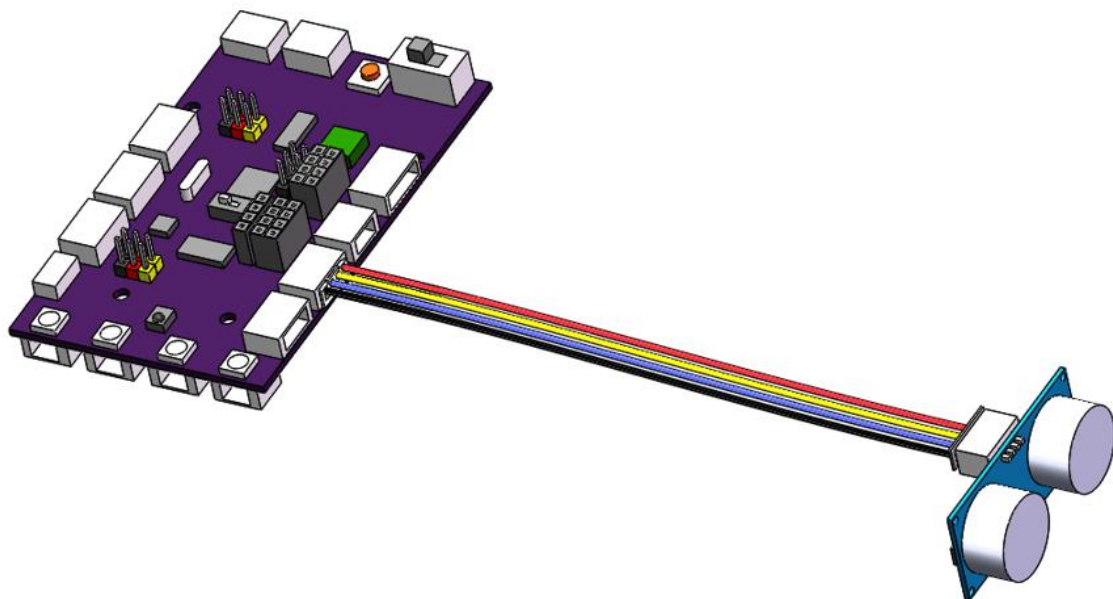
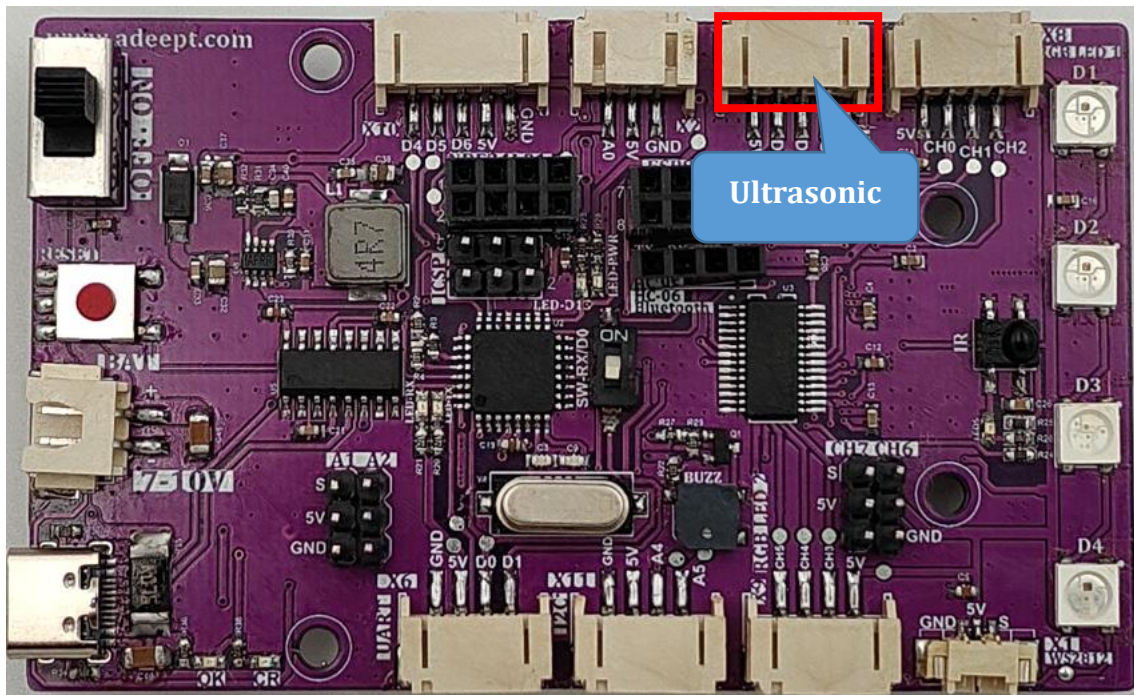


7.3 Wiring diagram

Ultrasonic Module	Arduino(X3)
VCC	5V
Trig	D7
Echo	D8
GND	GND

Connect the ultrasound to the red box in the picture.

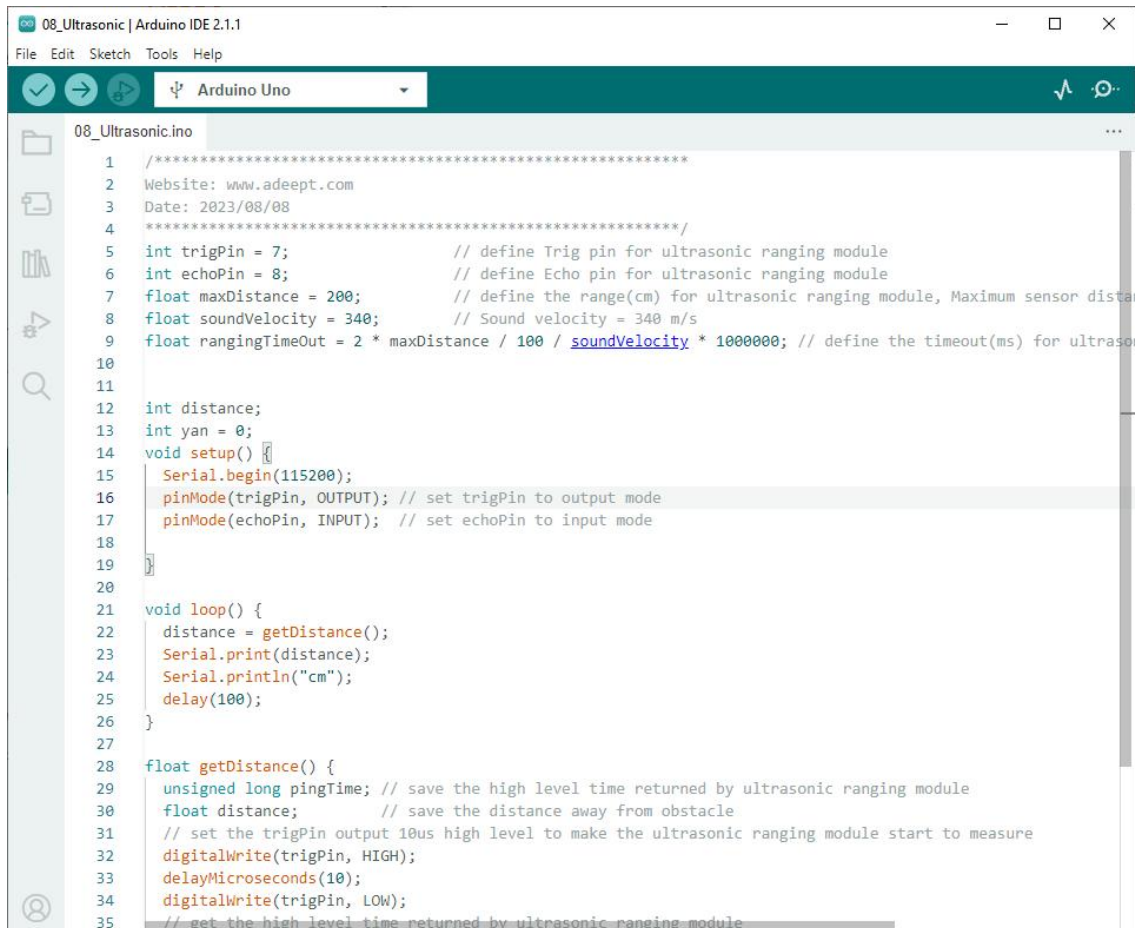
Figure as below:



The ultrasonic module uses a 4pin cable, the color is as shown in the picture, and the length is 15CM.

7.4 How to control Ultrasonic module

1. Connect your computer and Adeept Robot Control Board with a USB cable.
2. Open "07_Ultrasonic" folder in `/Code`, double-click `"07_Ultrasonic.ino"`.



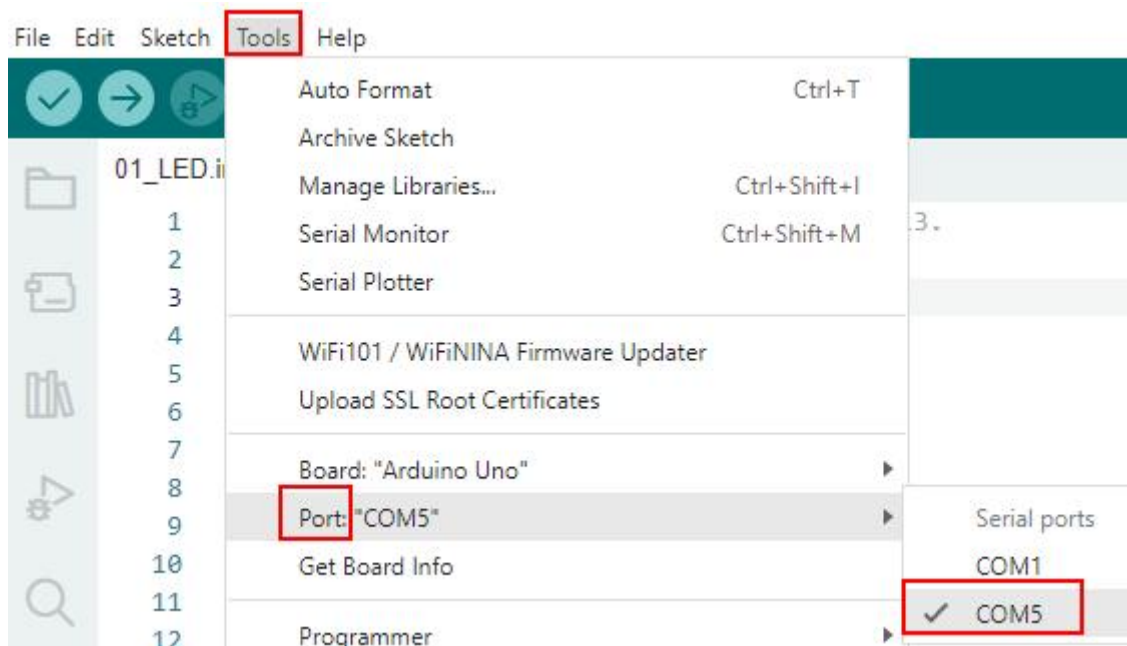
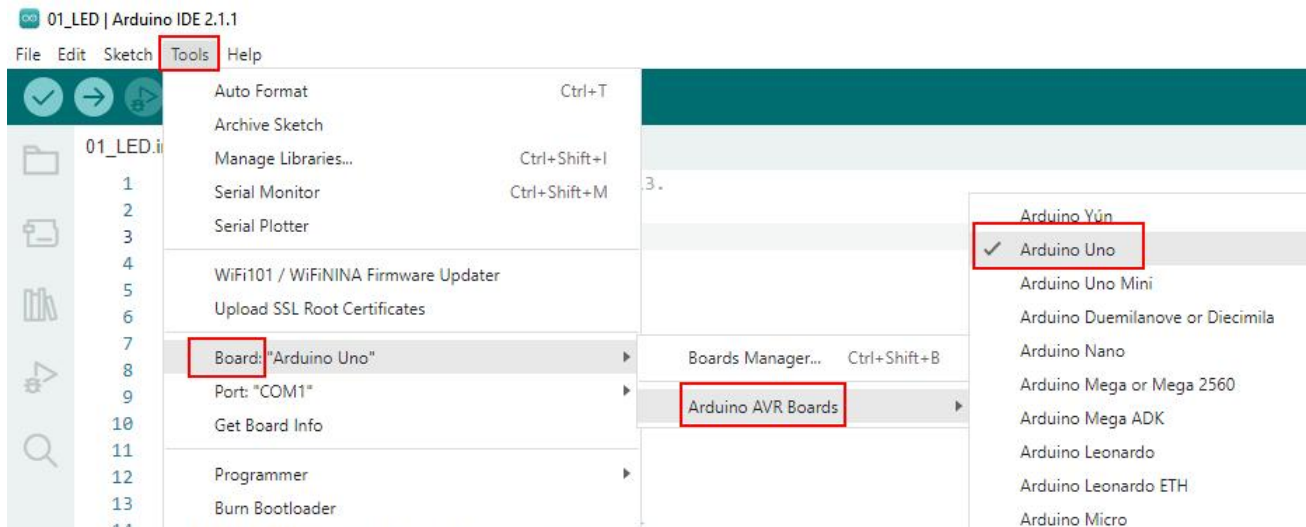
```
08_Ultrasonic.ino
1  /*****
2  Website: www.adeept.com
3  Date: 2023/08/08
4  *****/
5  int trigPin = 7;           // define Trig pin for ultrasonic ranging module
6  int echoPin = 8;           // define Echo pin for ultrasonic ranging module
7  float maxDistance = 200;   // define the range(cm) for ultrasonic ranging module, Maximum sensor distance
8  float soundVelocity = 340; // Sound velocity = 340 m/s
9  float rangingTimeOut = 2 * maxDistance / 100 / soundVelocity * 1000000; // define the timeout(ms) for ultrasonic ranging module
10
11
12  int distance;
13  int yan = 0;
14  void setup() {
15    Serial.begin(115200);
16    pinMode(trigPin, OUTPUT); // set trigPin to output mode
17    pinMode(echoPin, INPUT);  // set echoPin to input mode
18  }
19
20
21  void loop() {
22    distance = getDistance();
23    Serial.print(distance);
24    Serial.println("cm");
25    delay(100);
26  }
27
28  float getDistance() {
29    unsigned long pingTime; // save the high level time returned by ultrasonic ranging module
30    float distance;         // save the distance away from obstacle
31    // set the trigPin output 10us high level to make the ultrasonic ranging module start to measure
32    digitalWrite(trigPin, HIGH);
33    delayMicroseconds(10);
34    digitalWrite(trigPin, LOW);
35    // get the high level time returned by ultrasonic ranging module
```


3. Select development board and serial port.

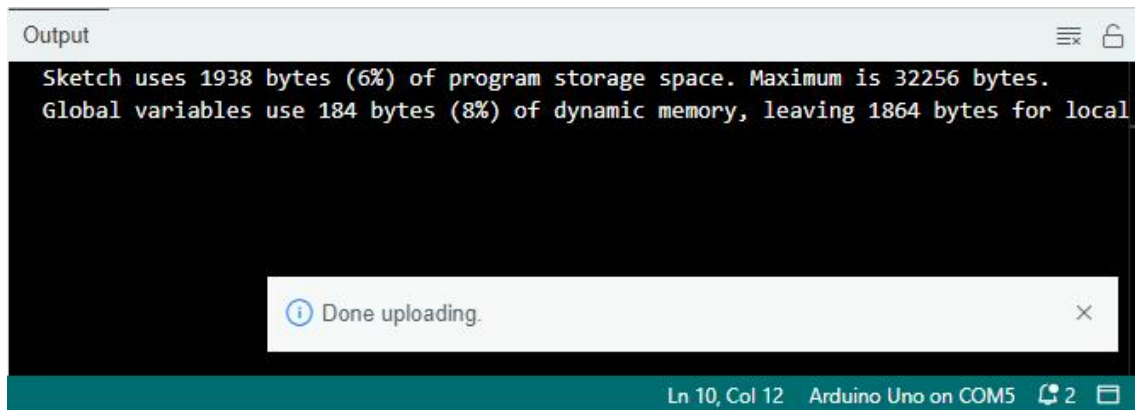
Board: Tools--->Board--->Arduino AVR Boards--->Arduino Uno

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

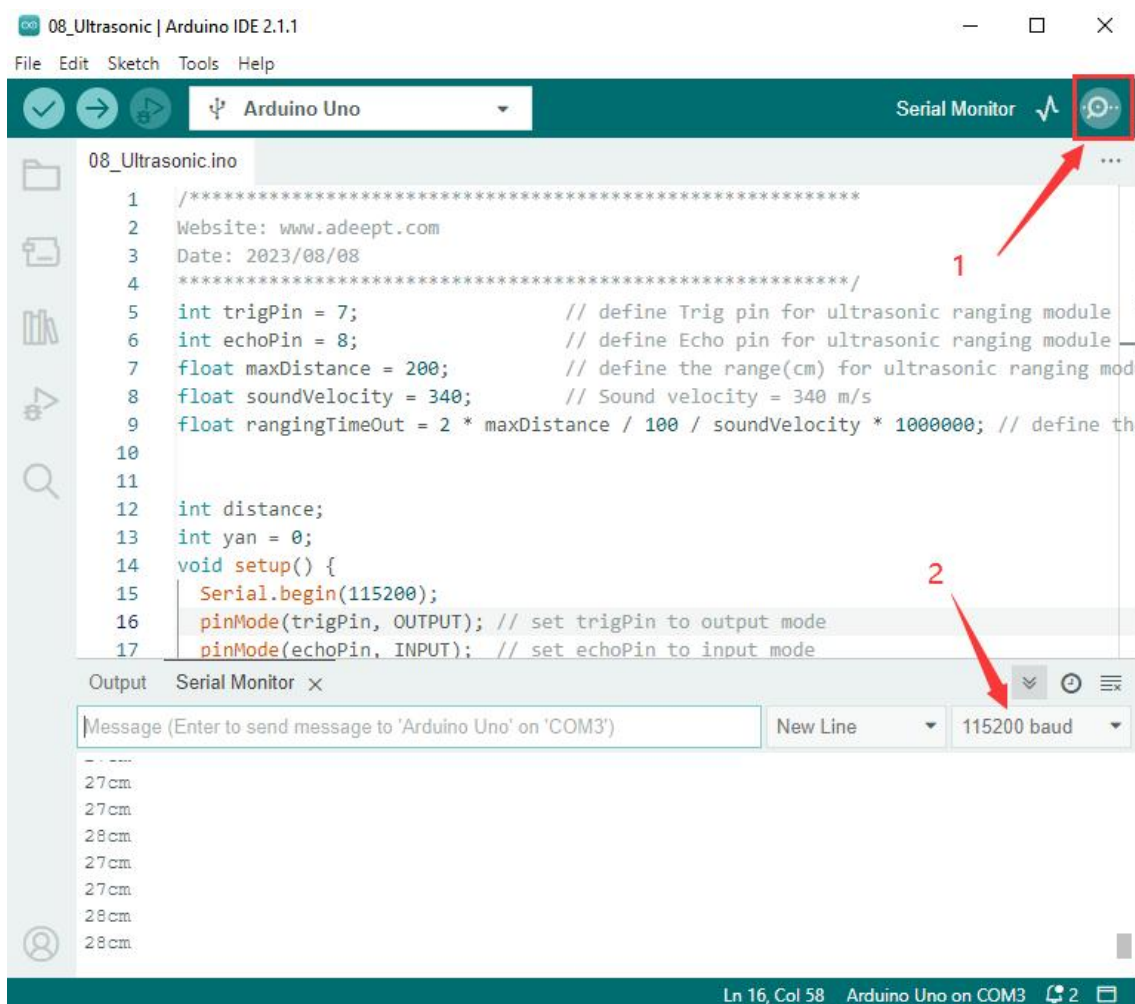
Note: The port number will be different in different computers.



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. Click Serial Monitor, Set the baud rate as 115200.



6. You will be able to see the distance value detected by the ultrasonic module displayed on the screen.

7.5 Code

```
1. int trigPin = 7;           // define Trig pin for ultrasonic ranging module
2. int echoPin = 8;          // define Echo pin for ultrasonic ranging module
3. float maxDistance = 200;   // define the range(cm) for ultrasonic ranging
    module, Maximum sensor distance is rated at 400-500cm.
4. float soundVelocity = 340; // Sound velocity = 340 m/s
5. float rangingTimeOut = 2 * maxDistance / 100 / soundVelocity * 1000000; // def
    ine the timeout(ms) for ultrasonic ranging module
6.
7. int distance;
8. void setup() {
9.     Serial.begin(115200);
10.    pinMode(trigPin, OUTPUT); // set trigPin to output mode
11.    pinMode(echoPin, INPUT);  // set echoPin to input mode
12. }
13.
14. void loop() {
15.     distance = getDistance();
16.     Serial.print(distance);
17.     Serial.println("cm");
18.     delay(100);
19. }
20.
21. float getDistance() {
22.     unsigned long pingTime; // save the high level time returned by ultrasonic r
        anging module
23.     float distance;         // save the distance away from obstacle
24.     // set the trigPin output 10us high level to make the ultrasonic ranging mod
        ule start to measure
25.     digitalWrite(trigPin, HIGH);
26.     delayMicroseconds(10);
27.     digitalWrite(trigPin, LOW);
28.     // get the high level time returned by ultrasonic ranging module
29.     pingTime = pulseIn(echoPin, HIGH, rangingTimeOut);
30.     if (pingTime != 0) { // if the measure is not overtime
31.         distance = pingTime * soundVelocity / 2 / 10000; // calculate the obstacl
            e distance(cm) according to the time of high level returned
```



```
32.     return distance;    // return distance(cm)
33. }
34. else                // if the measure is overtime
35.     return maxDistance; // returns the maximum distance(cm)
36. }
```