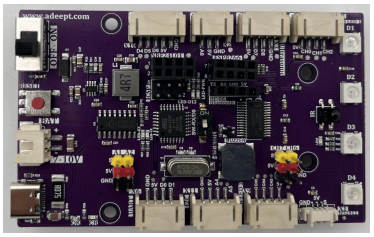
Lesson 10 Measure the Distance with an Ultrasonic Sensor

10.1 Overview

This course mainly introduces how to use ultrasonic sensors to measure distance, and combines Adeept Robot Control Board and Arduino development environment to explain in detail the required components, working principles, circuit connections, code writing and testing process, helping learners master the application of ultrasonic ranging technology in practical projects.

10.2 Required Components

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

10.3 Principle Introduction

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

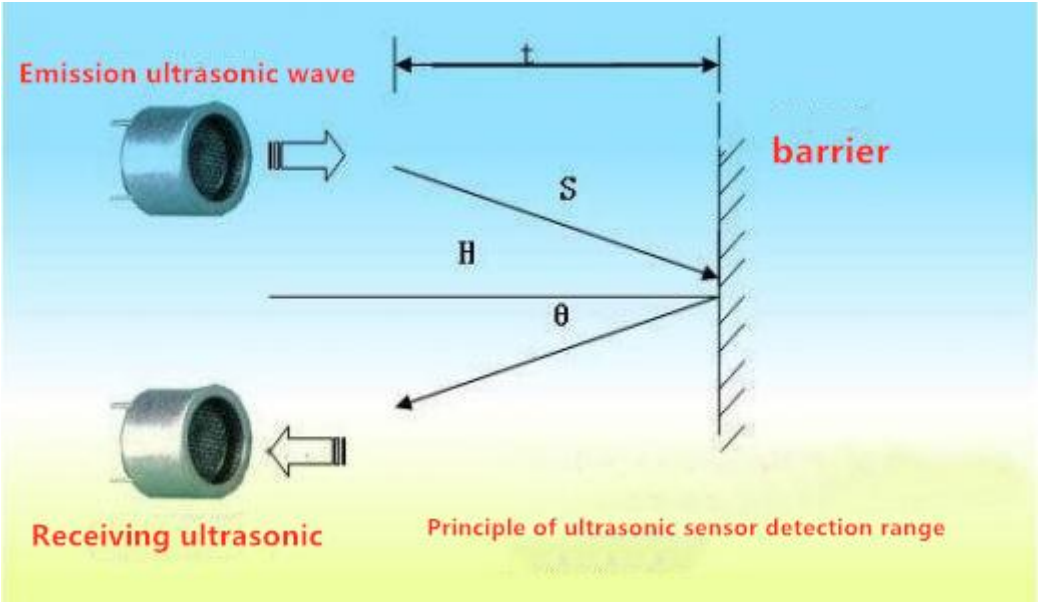
3mm; The module includes an ultrasonic transmitter, receiver and control circuit. The basic working principle is as follows:

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.

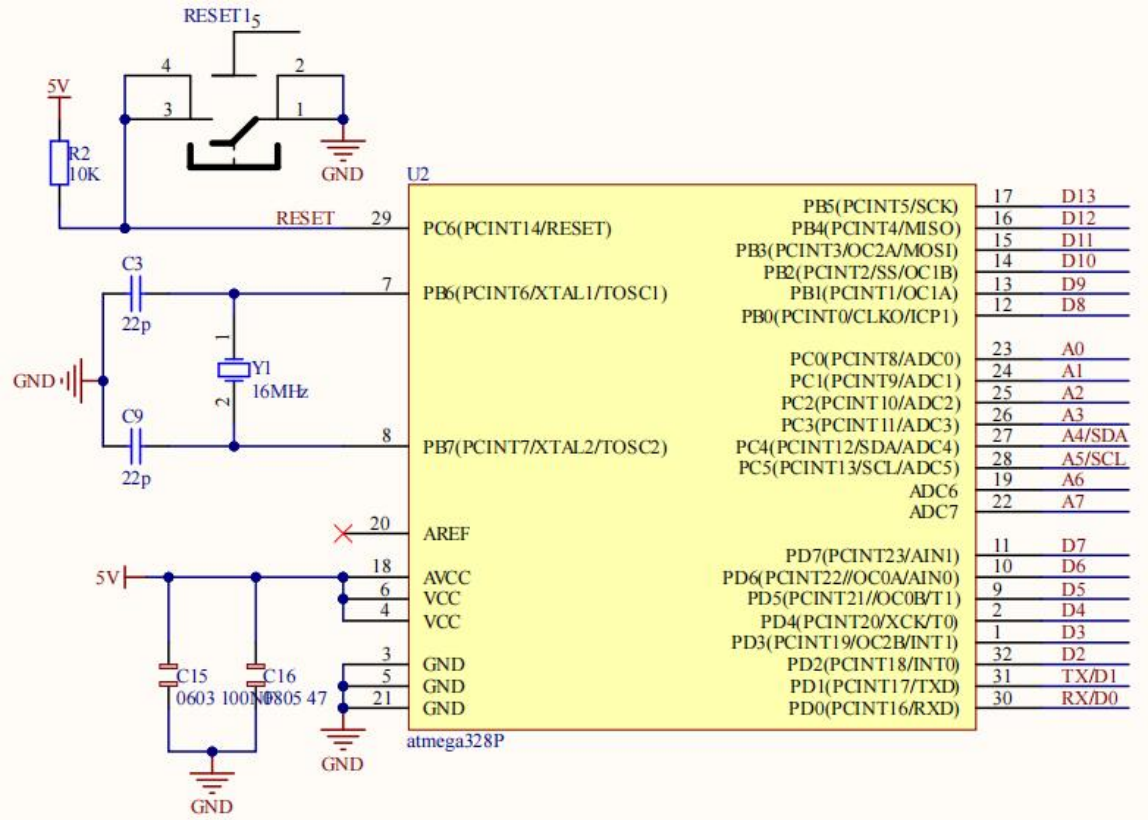
The principle of distance detection by ultrasonic ranging sensor: the method of detecting distance by ultrasonic is called echo detection method, that is, the ultrasonic transmitter emits ultrasonic waves in a certain direction, and the timer starts timing at the same time as the launch time. The ultrasonic 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.

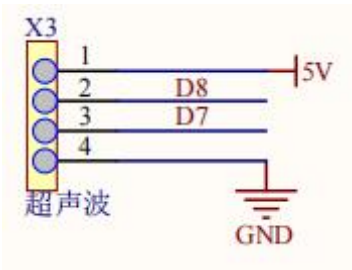


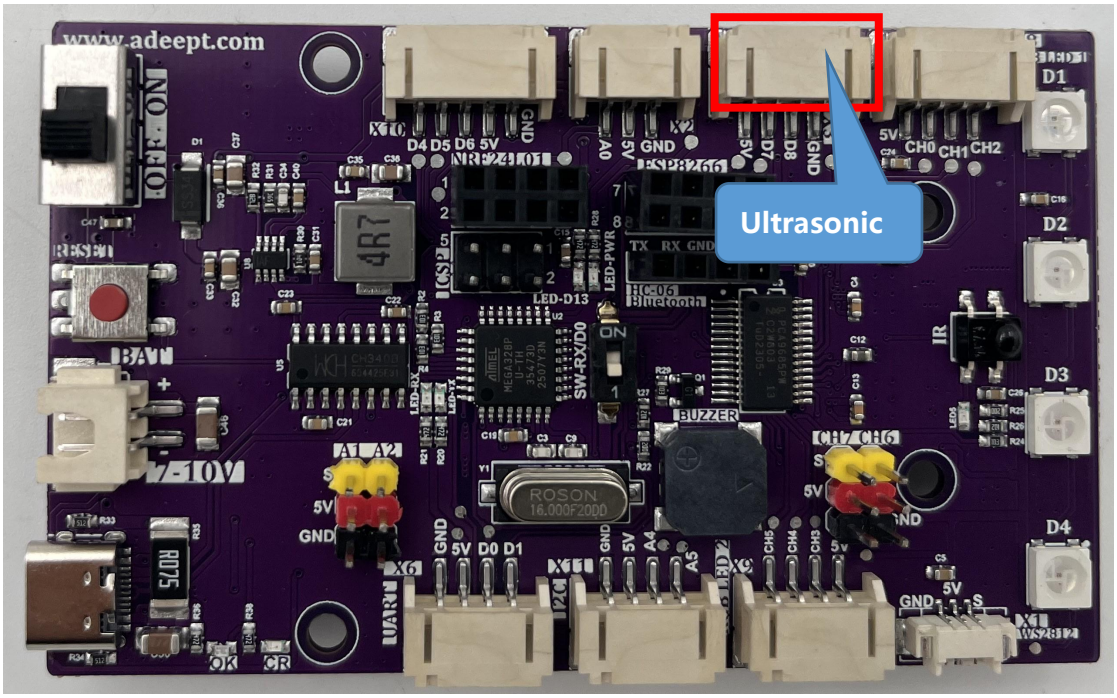
PINS of Adeept Robot Control Board	Arduino(X3)
VCC	5V

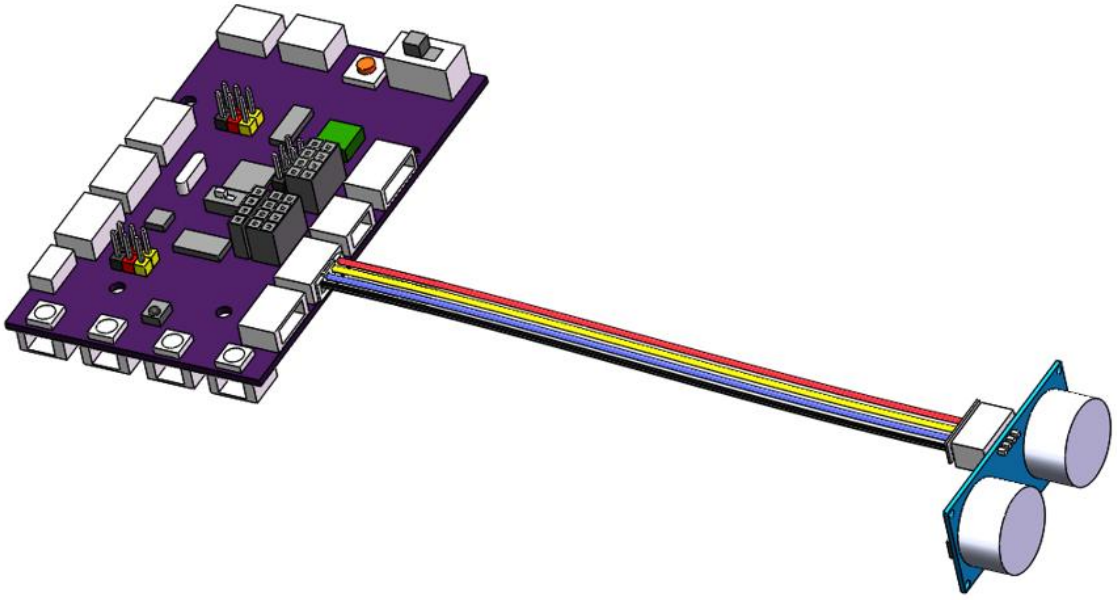
Trig	D7
Echo	D8
GND	GND

10.4 Wiring Diagram





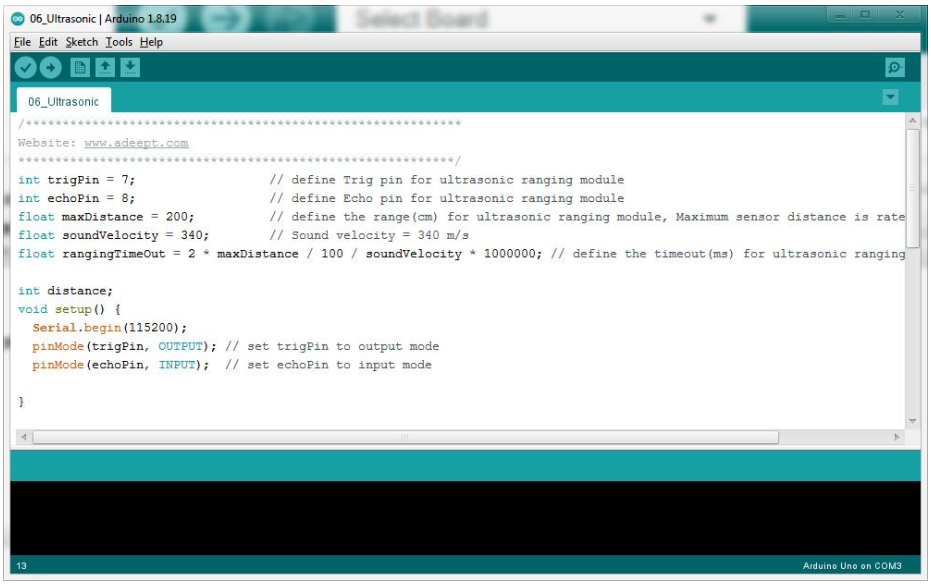




10.5 Demonstration

1. Connect your computer and Adeept Robot Control Board (Arduino Board) with a USB cable.

2. Open "**06_Ultrasonic**" folder in "/Code", double-click "06_Ultrasonic.ino".

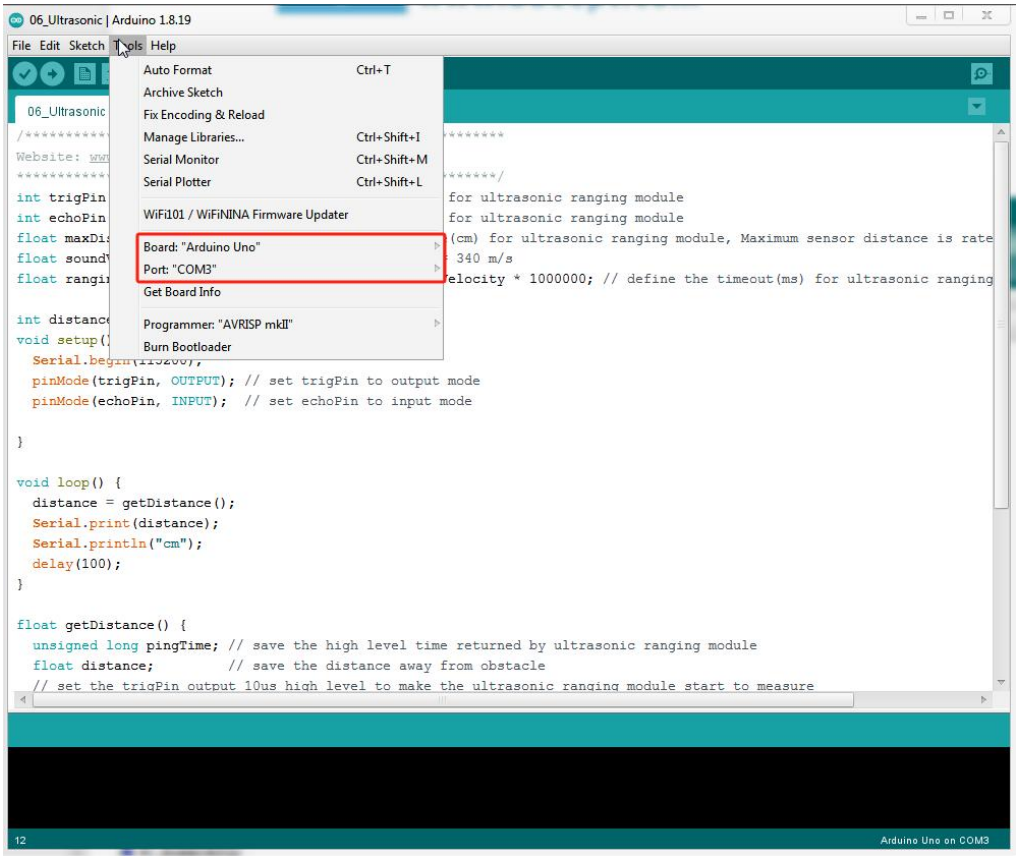


3. Select development board and serial port.

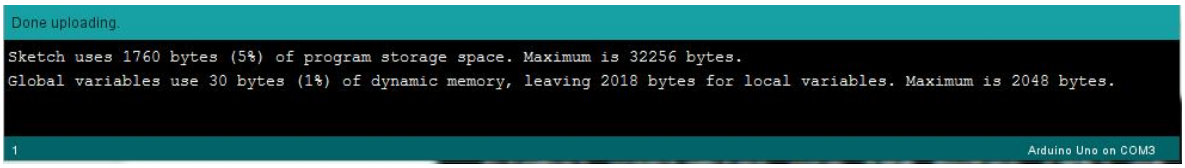
Board: Tools--->Board--->Arduino AVR Boards--->Arduino Uno

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

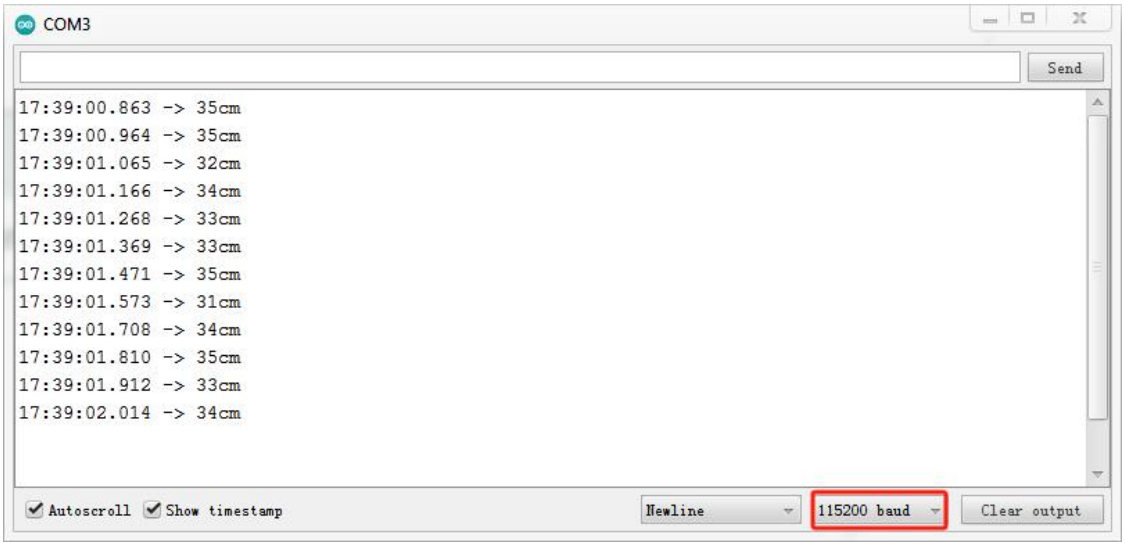
Note: The port number will be different in different computers.



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. After successfully running the program, the ultrasonic sensor will continuously measure the distance and display the measurement results in real-time on the serial monitor, in centimeters.

10.6 Code

Complete code refer to 06_Ultrasonic.ino

```
/*************************************************************
Website: www.adeept.com
*************************************************************/
int trigPin = 7;                    // define Trig pin for ultrasonic ranging module
int echoPin = 8;                    // define Echo pin for ultrasonic ranging module
float maxDistance = 200;            // define the range(cm) for ultrasonic ranging module, Maximum sensor
```

```
07  distance is rated at 400-500cm.
08  float soundVelocity = 340;        // Sound velocity = 340 m/s
09  float rangingTimeOut = 2 * maxDistance / 100 / soundVelocity * 1000000; // define the timeout(ms) for
10  ultrasonic ranging module
11
12  int distance;
13  void setup() {
14    Serial.begin(115200);
15    pinMode(trigPin, OUTPUT); // set trigPin to output mode
16    pinMode(echoPin, INPUT);  // set echoPin to input mode
17
18  }
19
20  void loop() {
21    distance = getDistance();
22    Serial.print(distance);
23    Serial.println("cm");
24    delay(100);
25  }
26
27  float getDistance() {
28    unsigned long pingTime; // save the high level time returned by ultrasonic ranging module
29    float distance;         // save the distance away from obstacle
30    // set the trigPin output 10us high level to make the ultrasonic ranging module start to measure
31    digitalWrite(trigPin, HIGH);
32    delayMicroseconds(10);
33    digitalWrite(trigPin, LOW);
34    // get the high level time returned by ultrasonic ranging module
35    pingTime = pulseIn(echoPin, HIGH, rangingTimeOut);
36    if (pingTime != 0) {  // if the measure is not overtime
37      distance = pingTime * soundVelocity / 2 / 10000;  // calculate the obstacle distance(cm) according
38  to the time of high level returned
39      return distance;    // return distance(cm)
40    }
41    else                  // if the measure is overtime
42      return maxDistance; // returns the maximum distance(cm)
43  }
```

Code explanation

Initialization Stage:

In the setup function, initialize serial communication and set the baud rate to 115200 for easy viewing of measurement results in the serial monitor; Set the trigger pin to output mode and the echo pin to input mode.

In the loop function, call the getDistance() to obtain the distance data.

If the distance data obtained is normal (greater than or equal to 0), print the distance value in cm on the serial monitor; If the distance retrieval timeout occurs (return value is -1), print "Measurement error" to indicate a measurement error.

Delay by 100ms after each measurement to avoid frequent measurements and ensure the stability of measurement results.