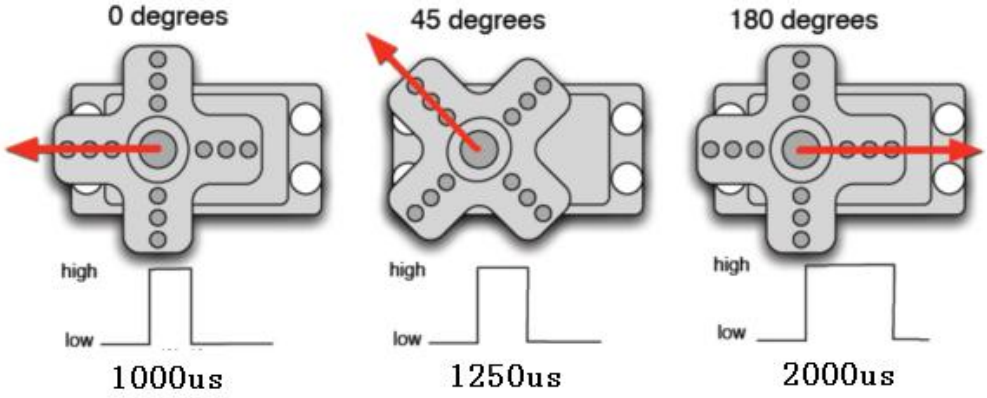
Course 18 ---servo control

The purpose of the experiment:

Based on Arduino UNO, a code is written to rotate the servo to the angle corresponding to the user's input number, and the angle print is displayed on the serial monitor of the Arduino IDE.

About the servo:

The actual object is shown below. Servo rotation angle is by adjusting the duty ratios of PWM (pulse width modulation) signal. The standard PWM (pulse width modulation) signal has a fixed period of 20ms (50Hz). Theoretically, pulse width distribution should be between 1 ms to 2 ms, but in fact between pulse width can be 0.5 ms and 2.5 ms. Pulse width and the servo rotation angle 0°～180° corresponds, as shown in the figure below.



Servo have many specifications, but all of the servo possess external three lines, with brown, red, orange, three kinds of color to distinguish. Due to brand is different, color is different, brown for the grounding line, red for positive line, orange for signal lines.

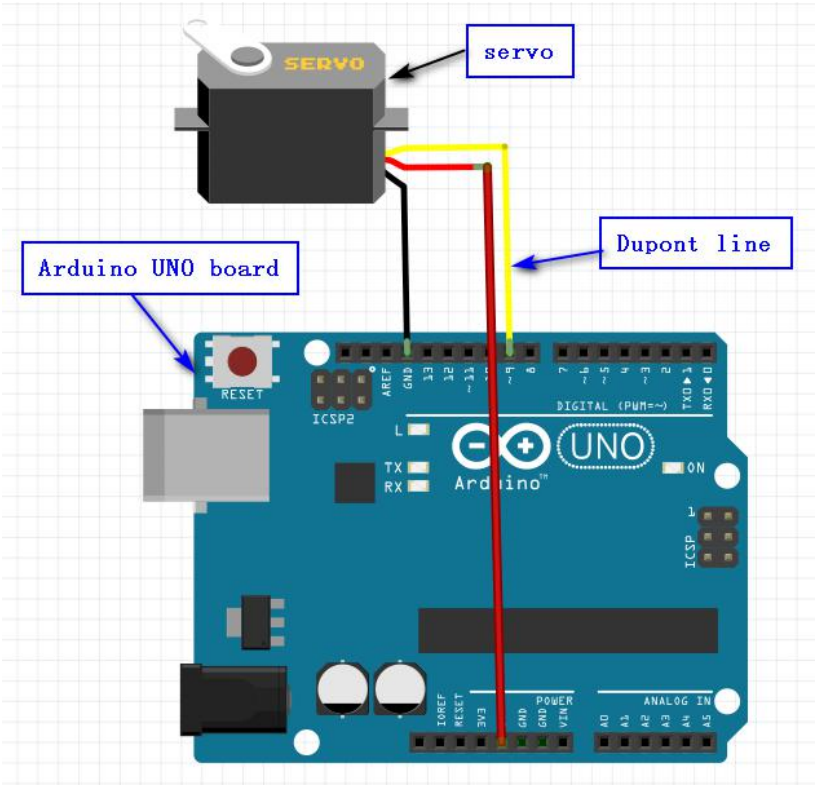
Note: Due to brand is different, for the same signal, different brands of servo rotation angle will be different.

List of components required for the experiment:

Arduino UNO board *1
USB cable *1
Servo *1
Dupont line *1 bunch

Actual object connection diagram：

We need to connect the circuit as shown in the figure below.



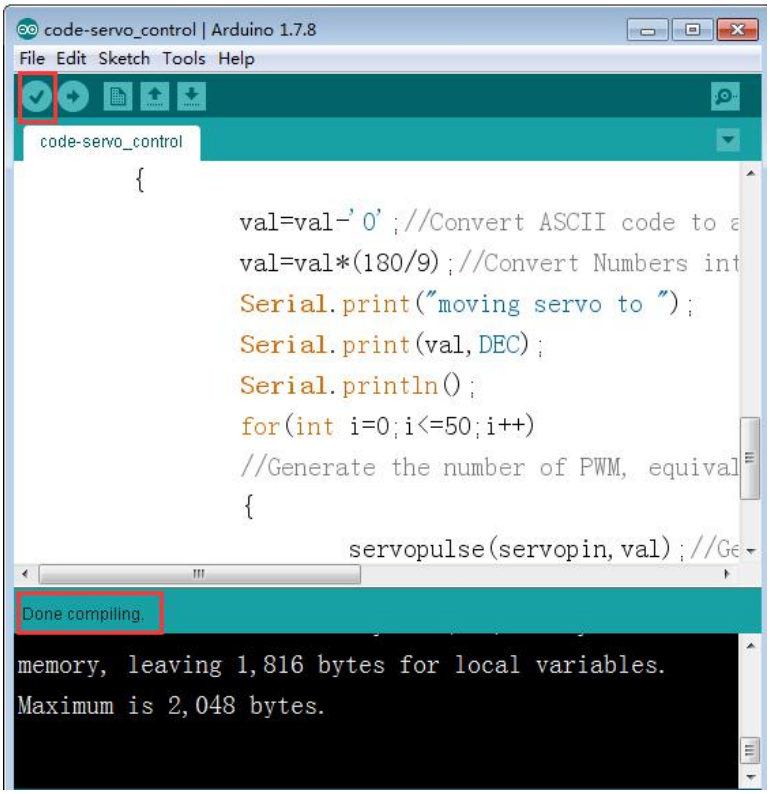
Experimental code analysis:

```
//UART send 1~9==>20~180 degree
int servopin=9;//Defining the port 9 for the servopin
int myangle;//Define Angle variable
int pulsewidth;//Define the pulse width variable
int val;
void servopulse(int servopin,int myangle)
/*A pulse function is defined to generate PWM values by simulation */
{
  pulsewidth=(myangle*11)+500;//Convert the Angle to 500-2480 pulse width
  digitalWrite(servopin,HIGH);//Giving a high level to the servo interface
  delayMicroseconds(pulsewidth);//The number of microseconds of delay pulse width
  digitalWrite(servopin,LOW);//Giving a low level to the servo interface
  delay(20-pulsewidth/1000);//The remaining time in the delay period
}
void setup()
{
  pinMode(servopin,OUTPUT);//Defining the servopin port for the output port
  Serial.begin(9600);//The baud rate is 9600
  Serial.println("servo=o_seral_simple ready" ) ;
```

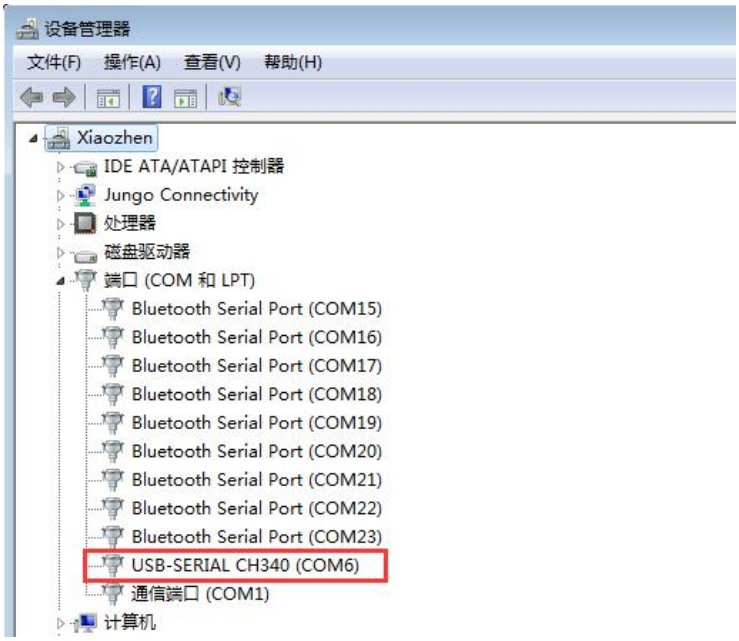
```
}
void loop()
{
    val=Serial.read();//Reading the data received by the serial port
    if(val>'0'&&val<='9')//Determineing whether the received data values conform to the range
    {
        val=val-'0';//Convert ASCII code to a value，例'9'-'0'=0x39-0x30=9
        val=val*(180/9);//Convert Numbers into angles，例 9*（180/9）=180
        Serial.print("moving servo to ");
        Serial.print(val,DEC);
        Serial.println();
        for(int i=0;i<=50;i++)
                //Generate the number of PWM, equivalent delay to ensure that the response Angle can be turned
        {
            servopulse(servopin,val);//Generate PWM values by simulation
        }
    }
}
```

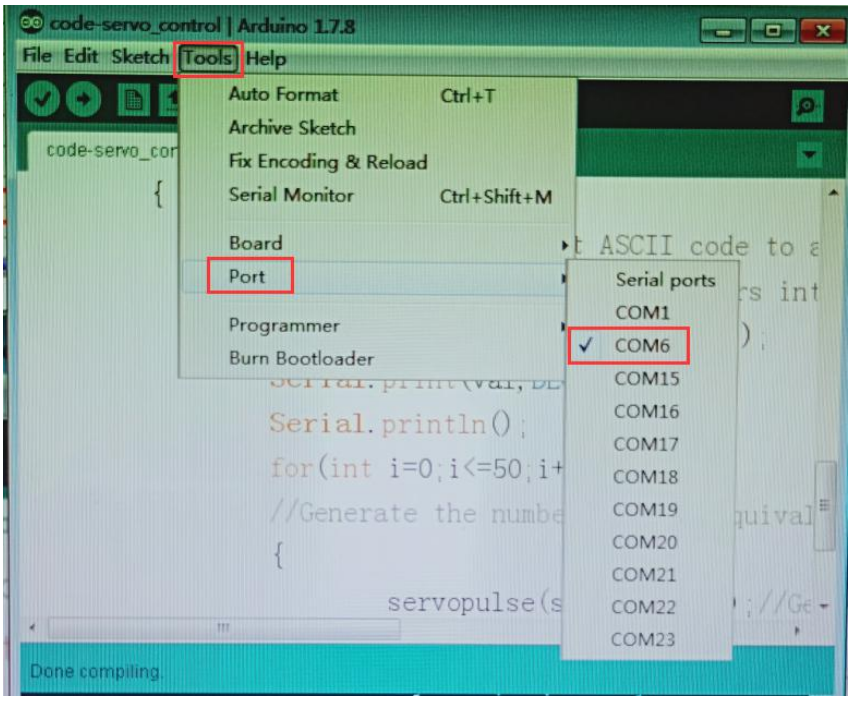
Experimental steps:

1.We need to open the code for this experiment: **code-servo_control.ino**, click "√"under the menu bar, compile the code, and wait for the words of **Done compiling** in the lower left corner, as shown in the following figure.

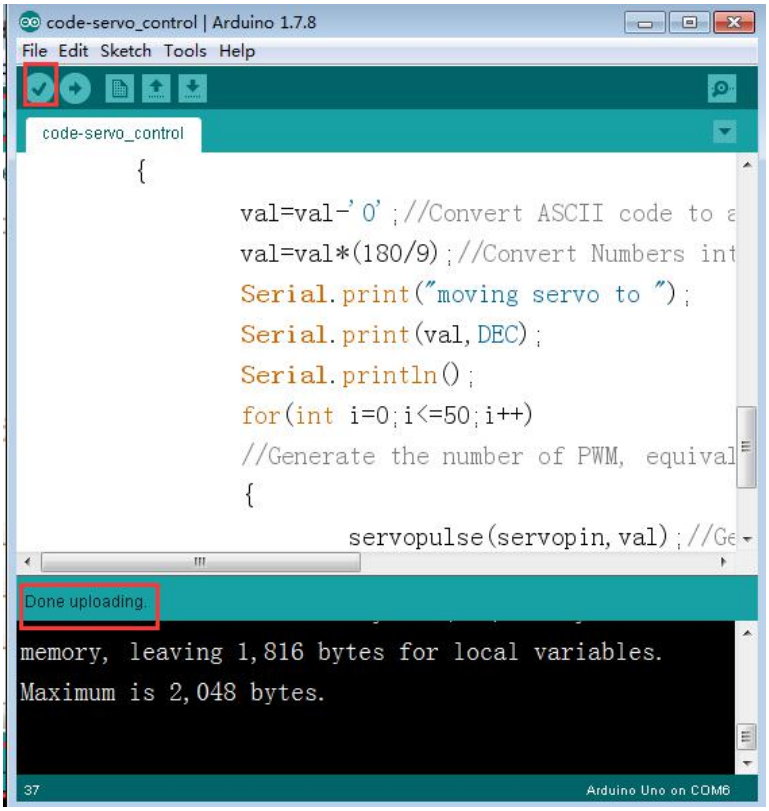


2. In the menu bar of Arduino IDE，you need to select the【Tools】---【Port】--- select the port that the serial number displayed by the device manager just now.for example:COM6,as shown in the following figure.

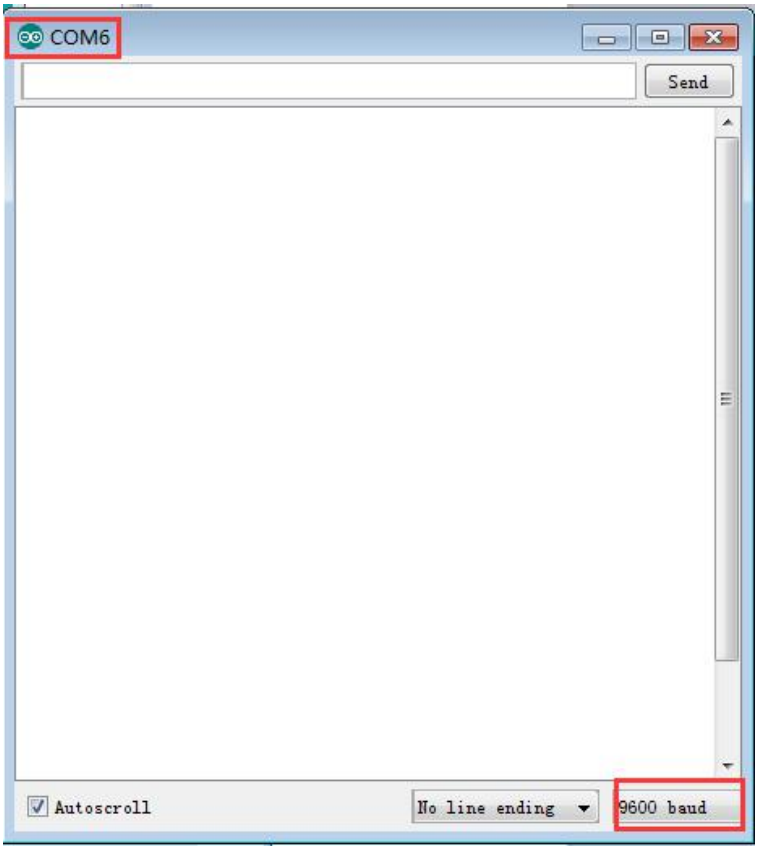




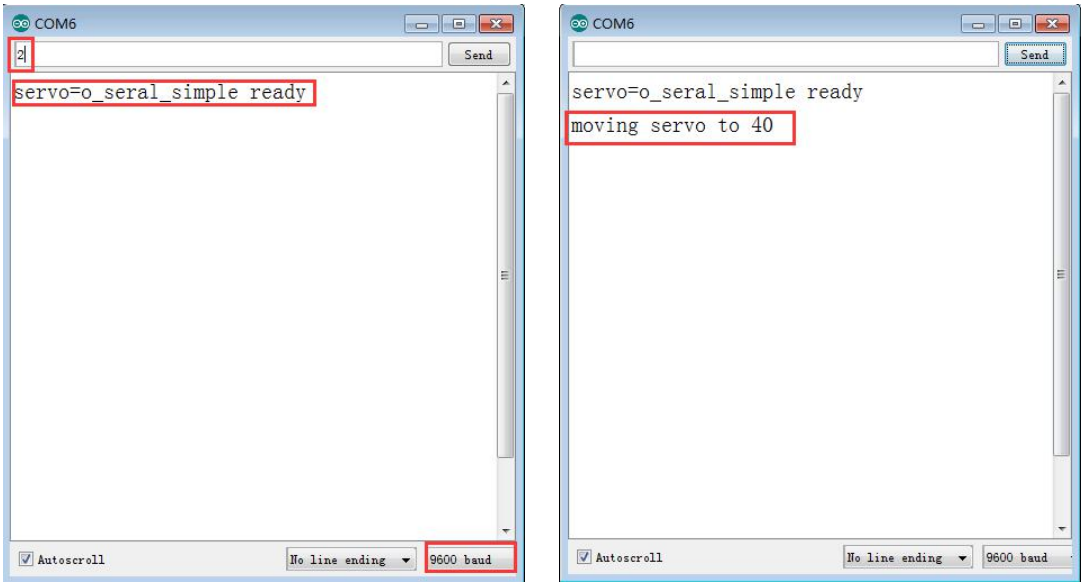
3. After the selection is completed, you need to click "→"under the menu bar,and upload the code to the Arduino UNO board, when appears to **Done uploading** on the lower left corner , that means that the code has been successfully uploaded to the Arduino UNO board, as shown in the following figure.

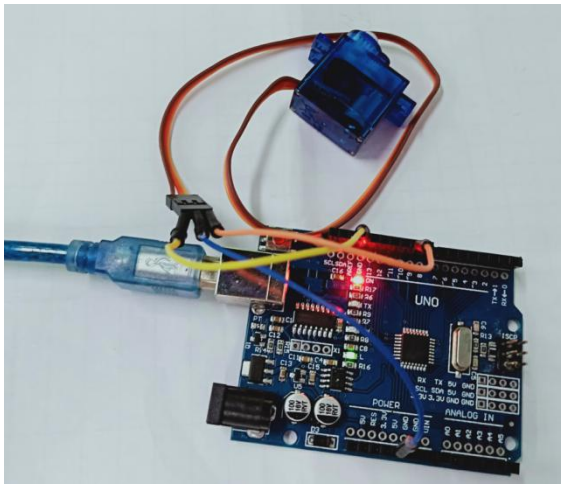


4. You can open the serial port monitor on the top right corner of Arduino IDE, A serial port of Arduino port will appear, and the baud rate is set to 9600 on the lower right corner, as shown in the following figure.



5.After the code is uploaded, we open the serial port monitor of Arduino IDE , you can see the words "servo=o_seral_simple ready" written in the program. And then input a number between 1 ~ 9 randomly in the send box, servo will turn the corresponding angle. Moreover, the serial port monitor will print out the corresponding angle, a comment in the program: "UART send 1~9= >20~180 degree" as shown in the figure below (for example only).





The code of the experiment: