Serial port adjustment steering gear angle experiment

Introduction of the steering gear

In the robot electromechanical control system, the steering gear control effect is an important factor influencing the performance. The steering gear can be used as the basic output actuator in mems and model aircraft. Its simple control and output make the single-chip microcomputer system very easy to interface with it.

A steering gear is a position (Angle) servo driver for control systems that require constant change in Angle and can be maintained. Currently in high grade remote control toys, such as aircraft models, including aircraft models, submarine models; Remote-controlled robots are already widely used. It can be rotated to any Angle between 0 and 180 degrees and then stopped exactly as you command, thus suitable for control systems that require Angle change and retention. Steering gear is an unprofessional name. It is actually a servomotor, a set of automatic controls, made up of dc motors, reduction gear sets, sensors and control circuits. What is automatic control? So-called automatic control - by using a closed-loop feedback control circuit to constantly adjust the output deviation - keeps the system output constant.

Experiment Purpose

Keywish Arduino Uno R3 motherboard was used to realize the experiment of adjusting the steering gear Angle.

Component List

◆   Keywish Arduino Uno R3 motherboard
◆   USB cable
◆   SG90 steering gear
◆   Several jumper wires

Working Principle

The steering gear control signal enters the signal modulation chip from the channel of the receiver to obtain the dc bias voltage. It has an internal reference circuit that generates a reference signal with a period of 20ms and a width of 1.5ms. The obtained dc offset voltage is compared with the voltage of the potentiometer to obtain the voltage difference output. Finally, the positive and negative output of the voltage difference to the motor drive chip determines the positive and negative rotation of the motor. When the motor speed is constant, the potentiometer is rotated by the cascade reduction gear, so that the voltage difference is 0 and the motor stops rotating. The steering gear has the maximum rotation Angle, and the middle position refers to the volume from that position to the minimum Angle, and the maximum Angle is exactly the same. The most

important part, the maximum rotation Angle varies with the steering gear, but the bandwidth in the middle position is fixed, 1.5 ms.

Steering gear control

The control of the steering gear generally requires a time base pulse of about 20ms, and the high level part of the pulse is generally the Angle control pulse part within the range of 0.5ms~2.5ms. Take 180-degree servo as an example, then the corresponding control relationship is as follows:

- 0.5ms--------------0degree；
- 1.0ms------------45degree；
- 1.5ms------------90degree；
- 2.0ms-----------135degree；
- 2.5ms-----------180degree；
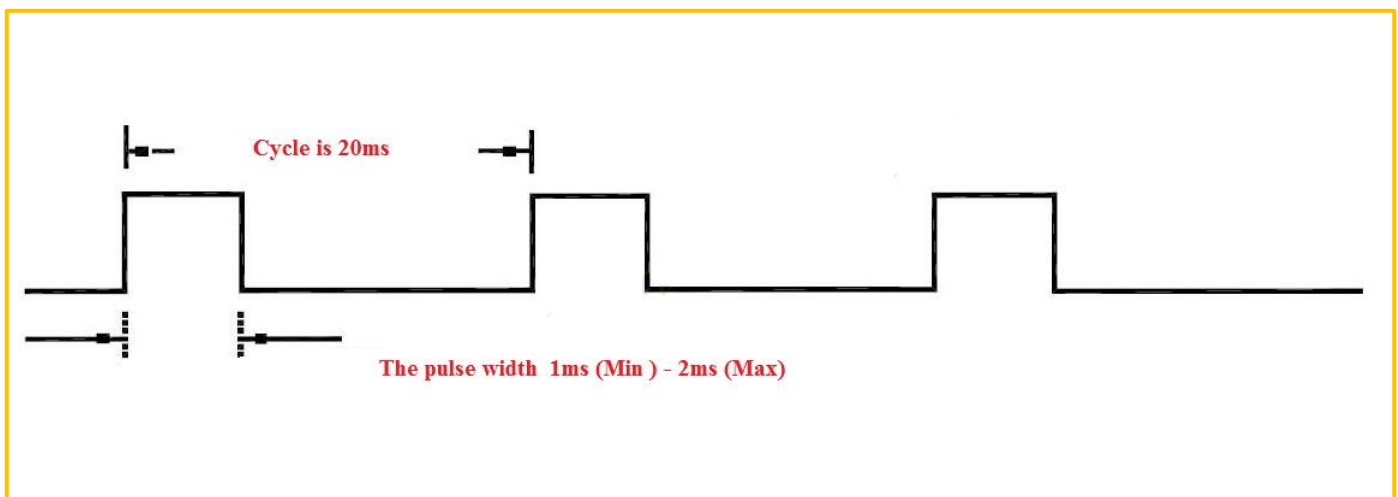
The rotation angle is produced by the continuous pulse from control line. This method is called pulse modulation. The length of pulse decides the rotation angle of steering gear. For example: the steering gear rotates to the middle position by 1.5 millisecond pulse(for 180 °steering gear, the middle position is 90 °). When the control system issues commands to move the steering gear to a particular position and make it keep a certain angle, then the influence of the external force won't change the angle, but the ceiling is its biggest torsion. Unless the control system continuously issues pulse to stable the steering angle, the angle will not always stay the same.

When the steering gear receives a pulse less than 1.5ms , the output shaft will take the middle position as standard and rotate a certain angle counterclockwise; when the received pulse is greater than 1.5ms, then the output shaft rotates clockwise. Different brands of steering gears, and even the same brand of different steering gears, the maximum and minimum value could be different.
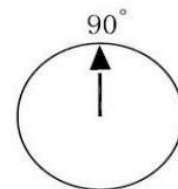
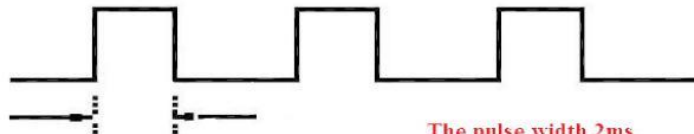The Min pulse width

The pulse width 1ms
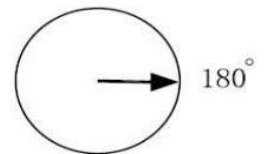
0°
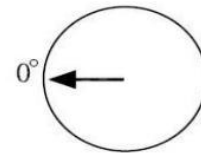
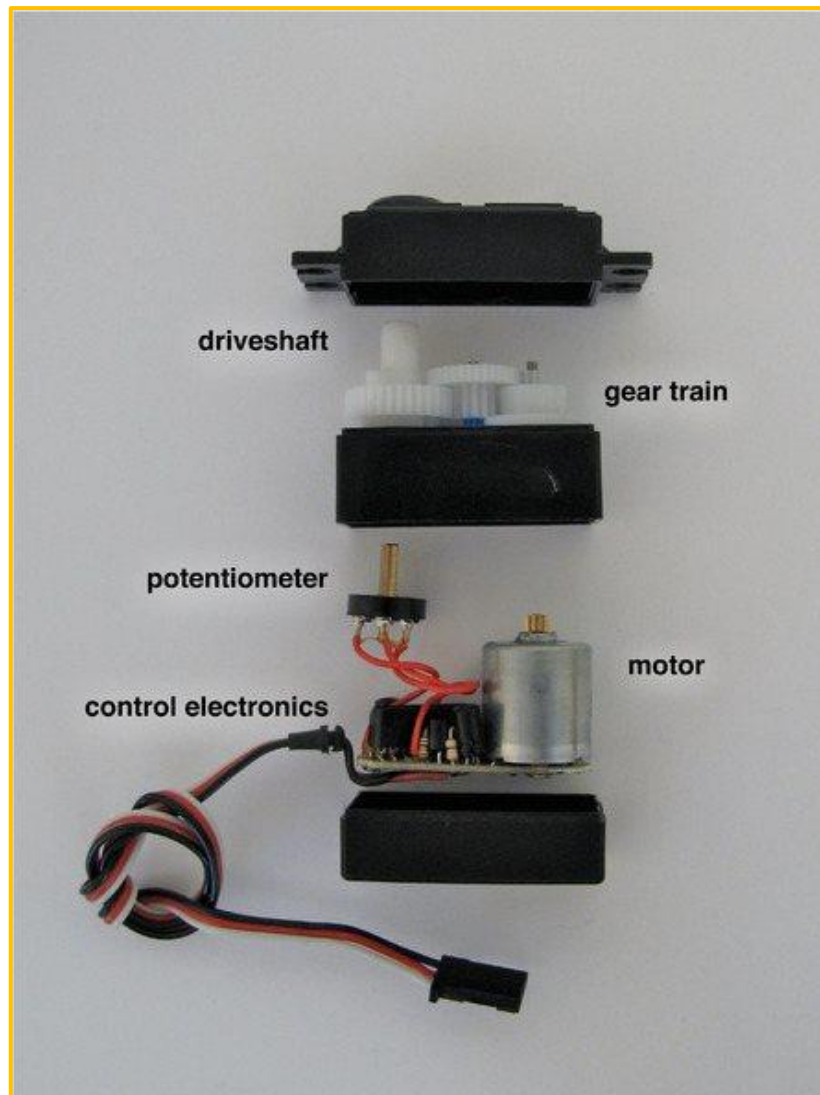The medium pulse width

The pulse width 1.5ms

90°

The Max pulse width

The pulse width 2ms

180°

Internal Structure of Steering Gear

The experimental principle

The signal line of the steering gear is connected to the Keywish Arduino Uno R3 main board. After burning the program, input the Angle of the steering gear to be adjusted in the serial port monitor, and make the steering gear turn to the position of the corresponding Angle
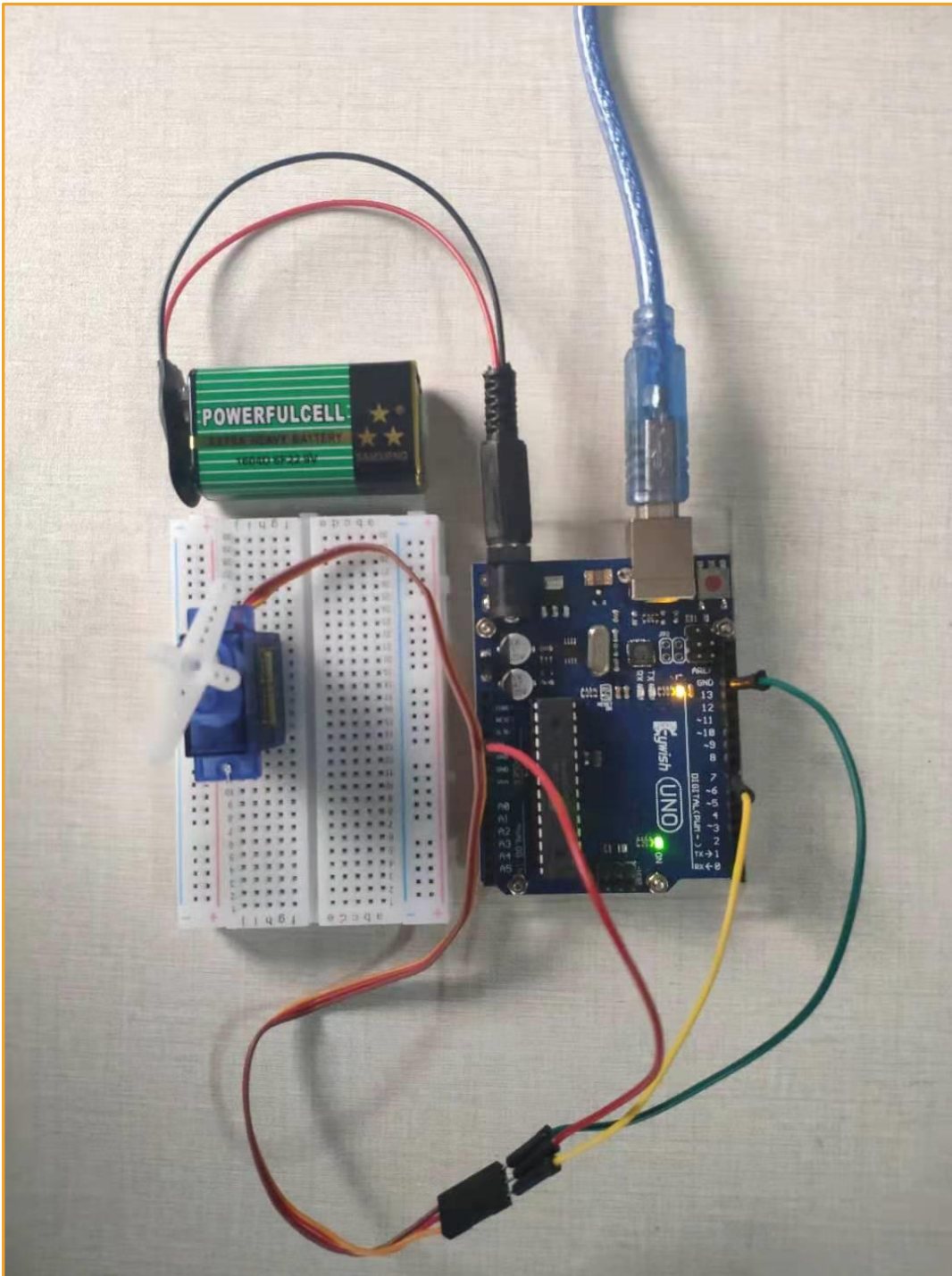
Wiring of Circuit

Arduino UNO	Steering Gear

| GND	Black line
VCC	Red line
13	Yellow line
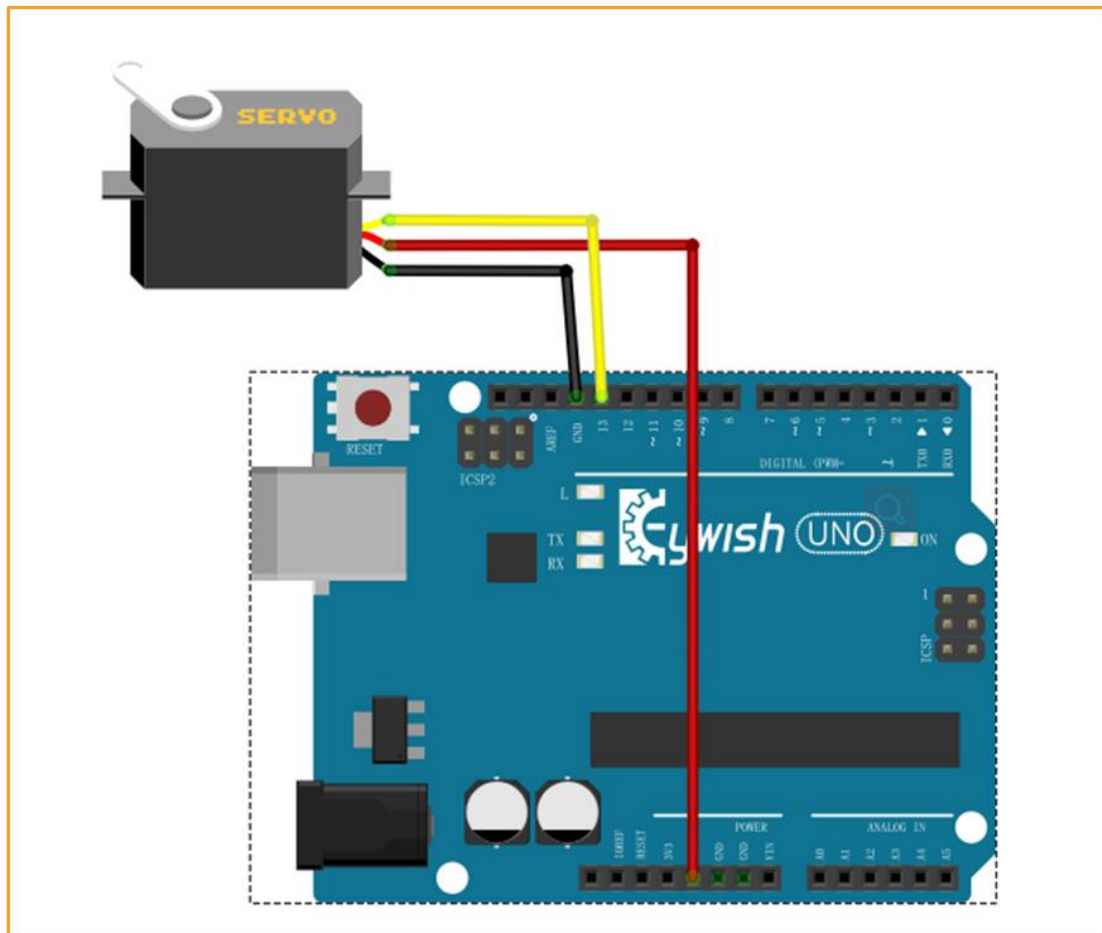
Experiment Result

Code

```cpp
#include <Servo.h>
Servo myservo;
int ServoPin = 13;
char inByte = 0;
int angle = 0;
String temp = "";//Temporary character variables, or use it for the cache
void setup()
{
  Serial.begin(9600);  //Set the baud rate
  pinMode(ServoPin, OUTPUT);
  myservo.attach(ServoPin);
}
void loop()
{
  while (Serial.available() > 0) //Determine whether the serial data
  {
    inByte = Serial.read();//Read data, the serial port can only read 1 character
    temp += inByte;// The characters read into temporary variables inside the cache, Continue
to determine the serial port there is no data, know all the data read out
  }

  if (temp != "")// Determine whether the temporary variable is empty
  {
    angle = temp.toInt();    //Convert variable string type to integer
    Serial.print("Servo degree: ");
    Serial.println(angle);
    //Control the servo to rotate the corresponding angle.
  }
   temp = "";// Please see temporary variables
  myservo.write(angle);
  delay(100);//Delayed 100 milliseconds

}
```
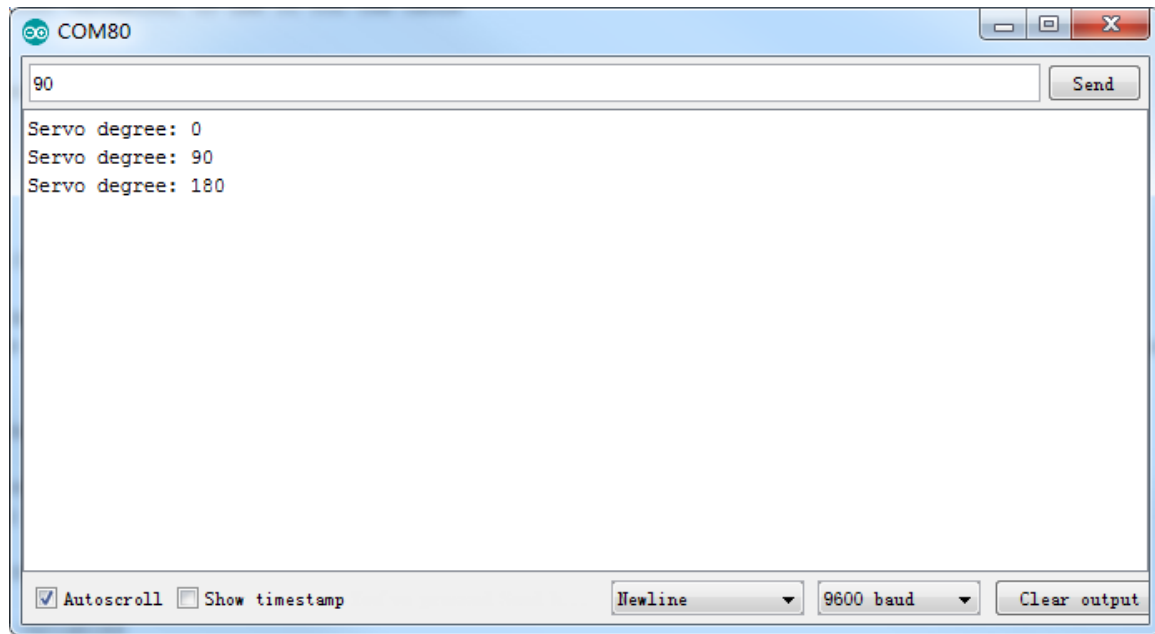
After burning the program to the UNO board, do not unplug the USB cable, plug the battery into the UNO board to power it, and then turn on the Arduino IED's serial monitor;
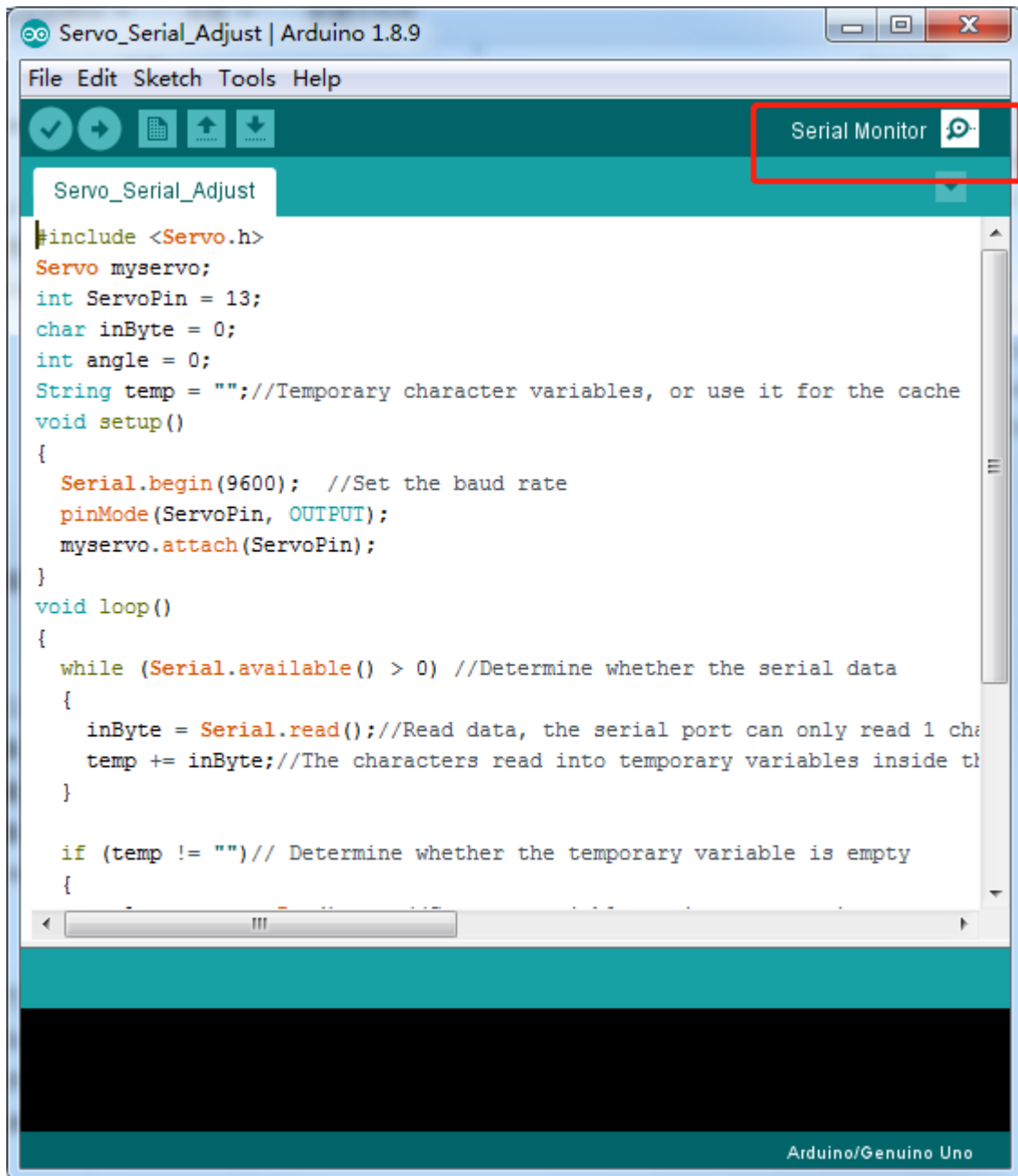
Open the serial monitor diagram

After opening the serial monitor, the serial monitor successively receives 0,90,180,90, such steering gear Angle values.

Serial input steering gear Angle value

After the serial monitor input the above steering gear Angle values, it will be found that the steering gear will turn.

Mixly graphical programming

Declare `inByte` as `char` ▾ value

Declare `angle` as `int` ▾ value

Declare `Temp` as `string` ▾ value

```
setup
    Serial ▾ baud rate | 9600

⚙ if | Serial ▾ isAvailable?
do   inByte | Serial ▾ read ▾
     temp | inByte + ▾ temp

⚙ if | temp ≠ ▾ " "
do   angle | int ▾ temp
     Serial ▾ print | " Servo degree: "
     Serial ▾ println | angle

temp | " "
Servo Pin | 13 ▾
Degree (0~180) | angle
      Delay(ms) | 100
```