Automatic rotation experiment of steering gear from 0 to 180 degrees

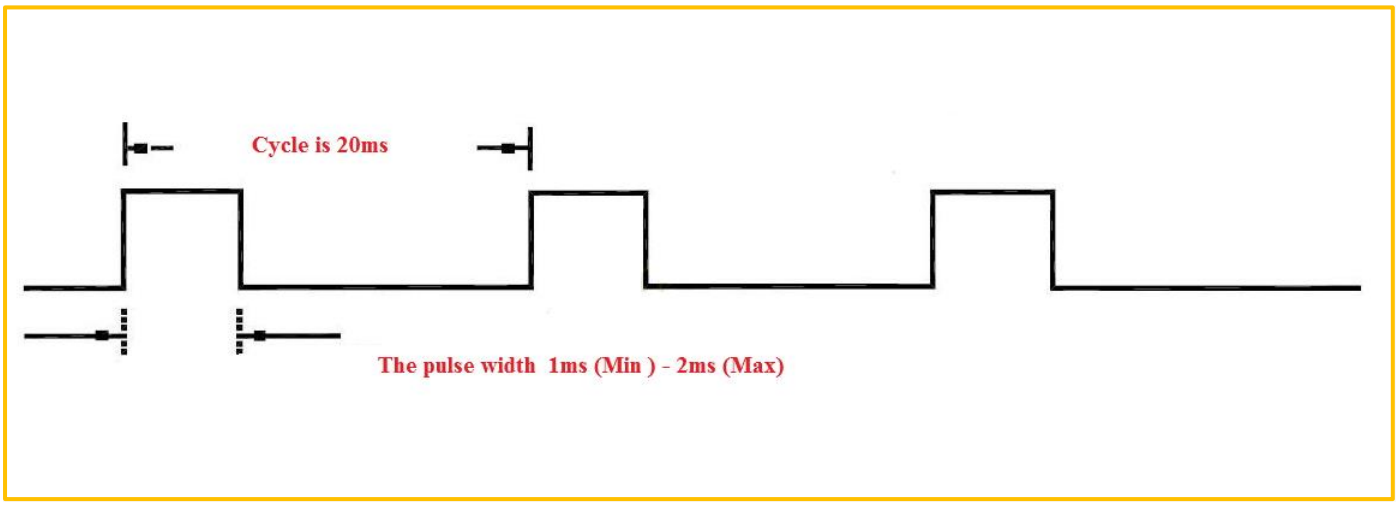
Introduction to Servo Motor

I guess you must have seen robots or high-tech products in American science-fiction films, at least heard of the noise of some moving automatic mechanical arms and audiences' scream. The noise comes from the rotation of the steering gear.

The steering gear is a kind of position (angle) servo driver, it can be rotated to any angle between 0 and 180 degrees, then precisely stop at your command, so it is suitable for those control systems which require angle changing and keeping . At present, it has been widely used in high-grade remote control toys, such as model aircraft, including the model plane, submarine model and remote control robot. Steering gear is an unprofessional name, in fact it is a kind of servo motor, a set of automatic control device which consists of DC motor, reduction gear group, sensor and control circuit. What is the automatic control? The so-called automatic control — continuously adjusting the output deviation by using a closed-loop feedback control circuit — makes the system output constant.

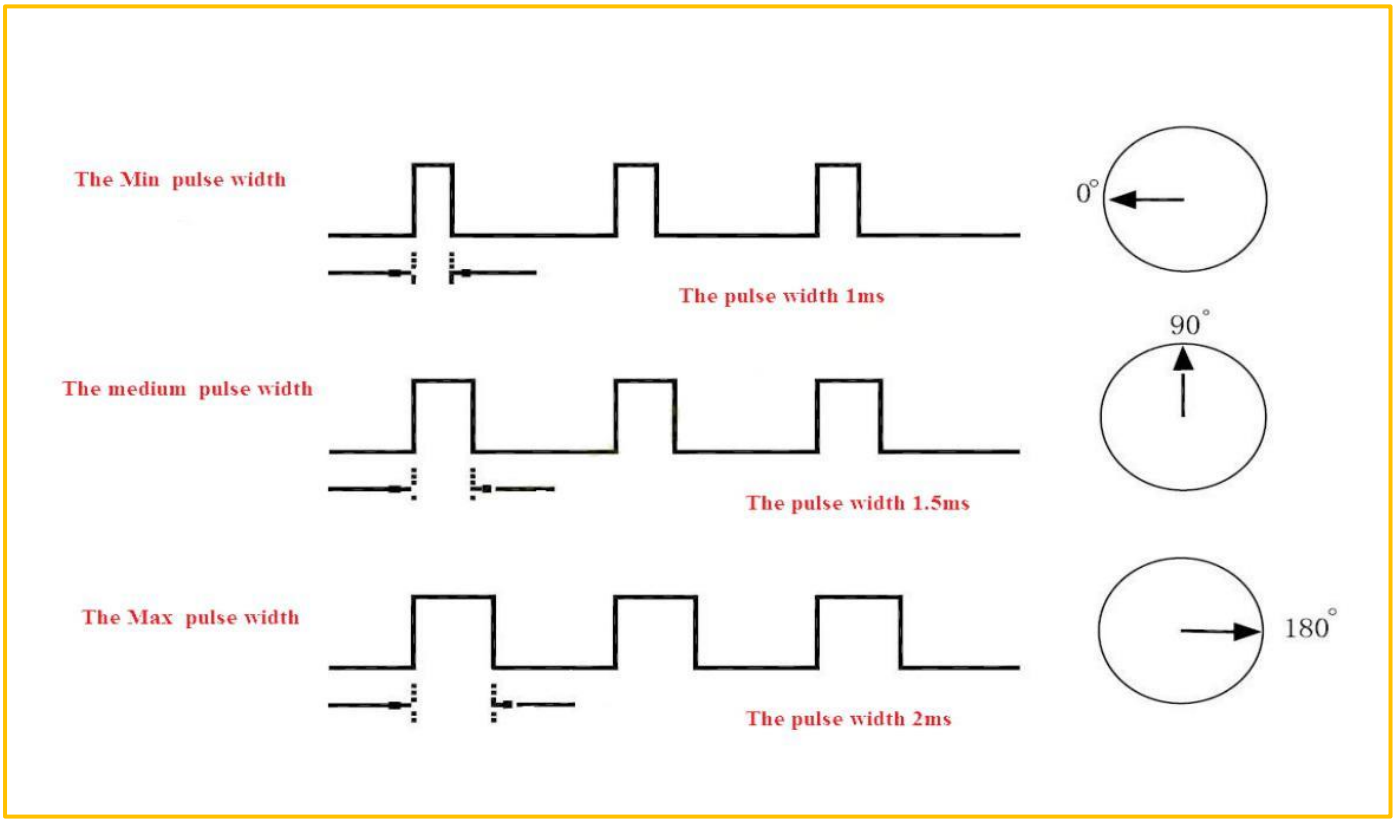
Experiment Principle

The steering gear servo system can be controlled by variable bandwidth pulse, the control line is used to transmit pulse. The parameters of the pulse consist minimum value, maximum value and frequency. In general, the cycle of the reference signal of the steering gear is 20ms, the bandwidth is 1.5ms. The reference signal is from the middle position. The steering gear has the maximum rotation angle, the middle position refers to the volumes from this position to the minimum angle and the maximum angle are exactly identical. The most important part, the maximum rotation angle varies with different steering gears, but of which the bandwidth of the middle position is certain, that is 1.5 ms.

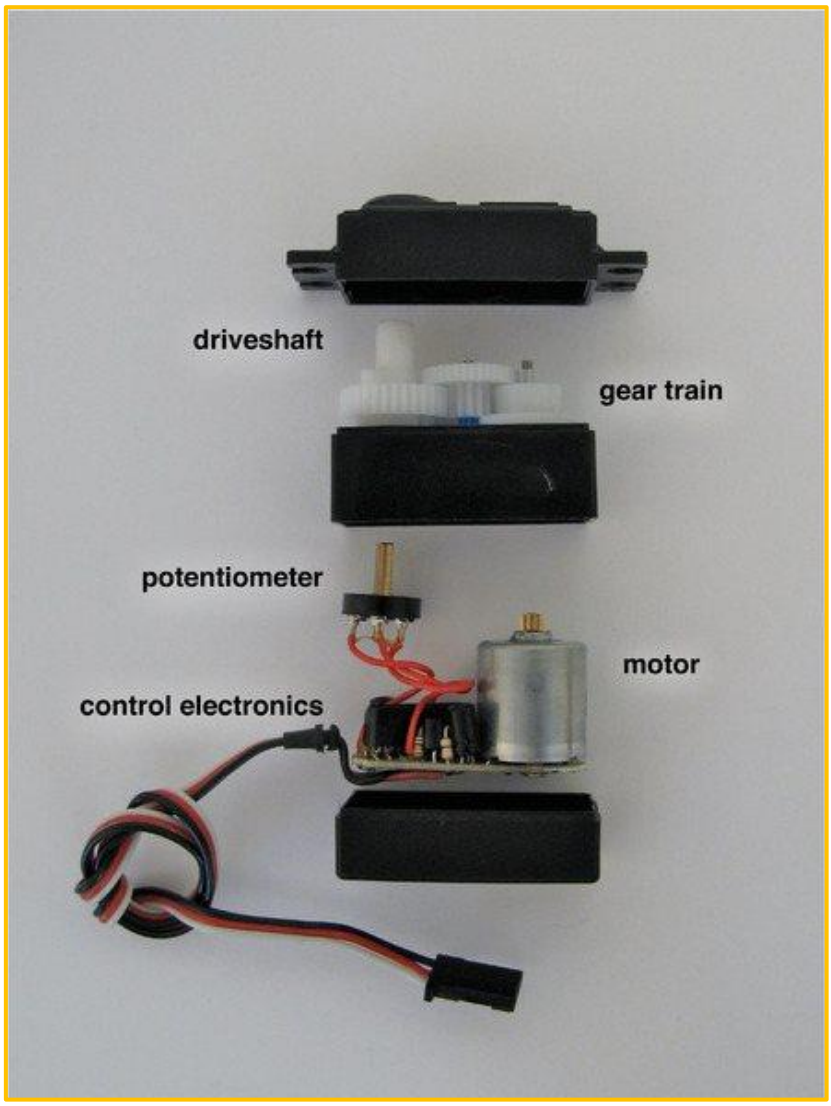


The rotation angle is produced by the continuous pulse from control line. This method is called pulse modulation. The length of pulse decides the rotation angle of steering gear. For example: the steering gear rotates to the middle position by 1.5 millisecond pulse(for 180 ° steering gear, the middle 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.



Internal Structure of Steering Gear



Experiment Purpose

- Understand the operating principle of steering gear
- The Keywish Arduino Uno R3 motherboard was used to control the automatic rotation of the steering engine from 0 °to 180 °.

Experimental purpose

- Through the bluetooth APP of mobile phone, the buzzer is controlled to emit the corresponding sound of piano keys

Experimental principle

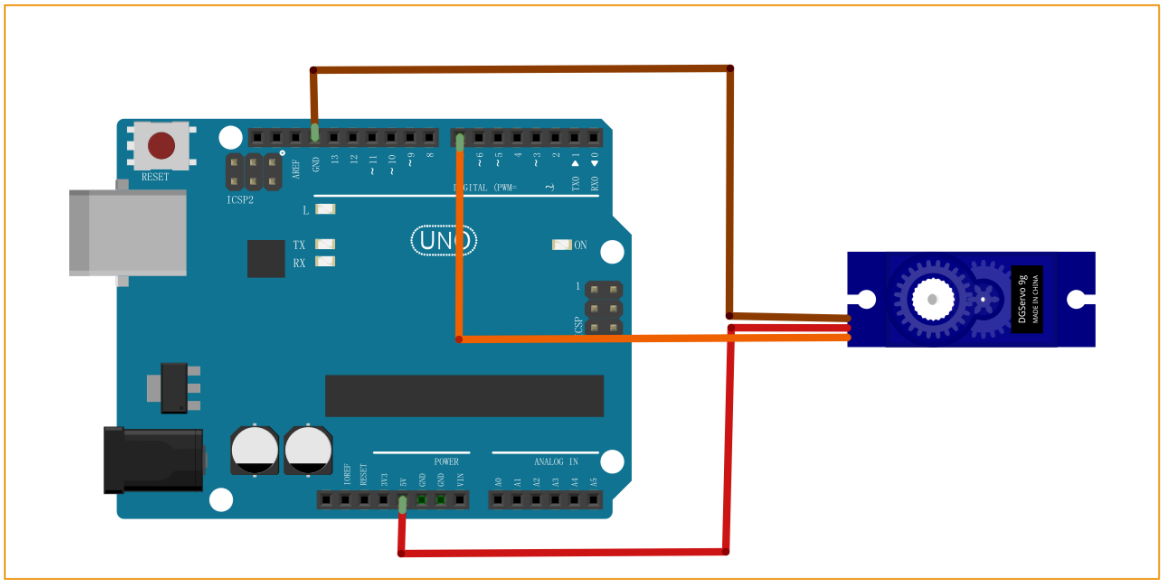
The signal wire of the steering gear is connected to the 7 feet of Keywish Arduino Uno R3 motherboard. After burning the program, input the Angle to adjust the steering gear in the serial port monitor to make the steering gear turn to the corresponding Angle

The component list

- Arduinos Uno motherboard
- Bread plate
- USB cable
- Servo Motor*1
- Jumpers

Wiring

Arduino UNO R3 board	Servo motor
5V	VCC(reed)
GND	GND（brown）
7	S（orange）



Arduino IDE programming program (using the steering engine library file)

```
#include <Servo.h>
  Servo myservo;
  int pos = 0;
  void setup()
  {
```

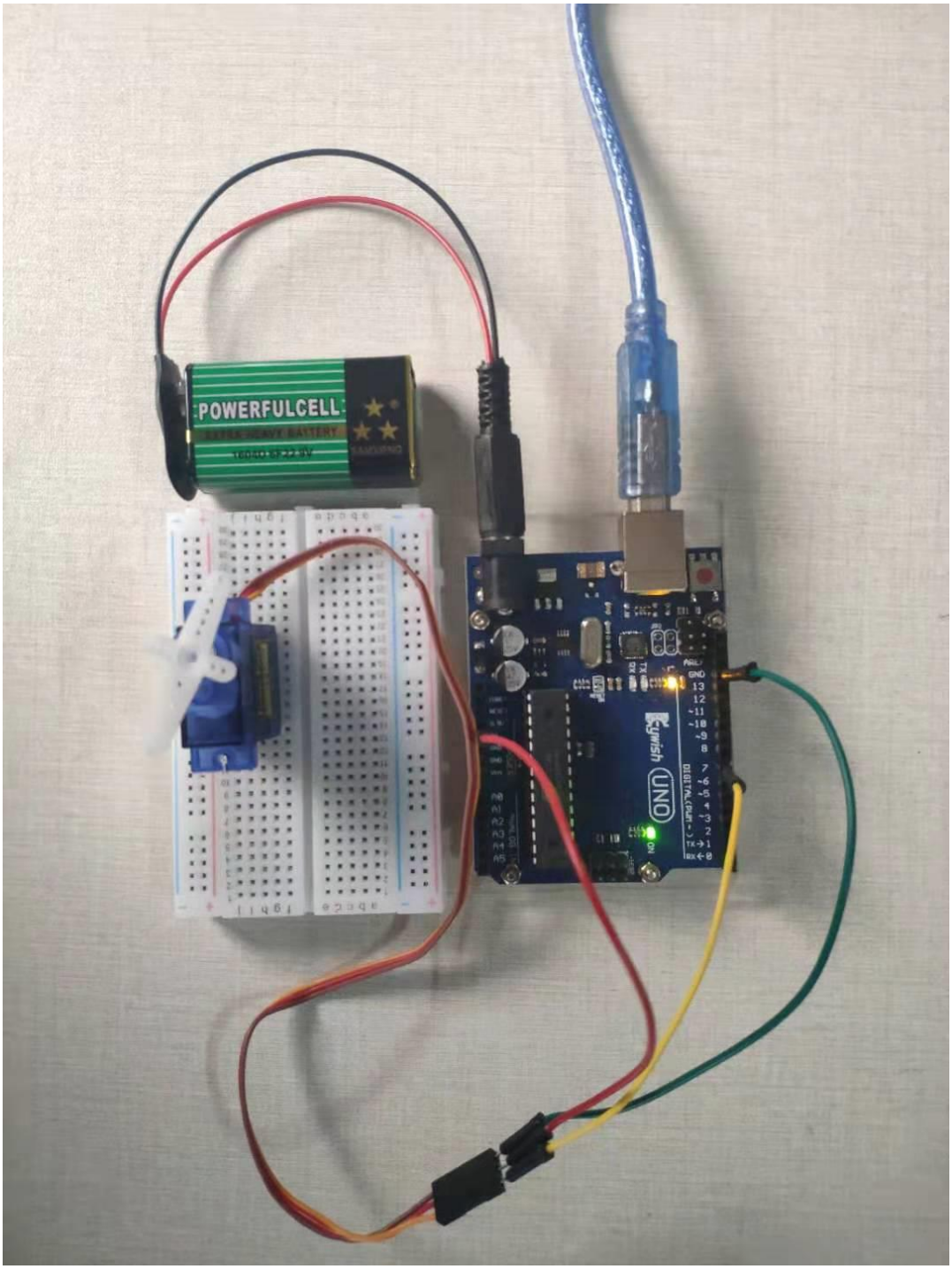
```
    myservo.attach(7);
  }
  void loop()
 {
       for(pos = 0; pos < 180; pos += 1)
   {
     myservo.write(pos);
     delay(15);
    }
    for(pos = 180; pos>=1; pos-=1)
    {
     myservo.write(pos);
     delay(15);
    }
 }
```

Arduino IDE programming program (no helm library files)

```
int servopin = 7;
void servopulse(int angle)
{
  int pulsewidth = (angle * 11) + 500;
  digitalWrite(servopin, HIGH);
  delayMicroseconds(pulsewidth);
  digitalWrite(servopin, LOW);
  delayMicroseconds(20000 - pulsewidth);
}
void setup()
{
  pinMode(servopin, OUTPUT);
}
void loop()
{
  for ( int angle = 0; angle < 180; angle += 2)
  {
    for (int i = 0; i < 10; i++)
    {
      servopulse(angle);
    }
  }
  for ( int angle = 180; angle>0; angle -= 2)
  {
```

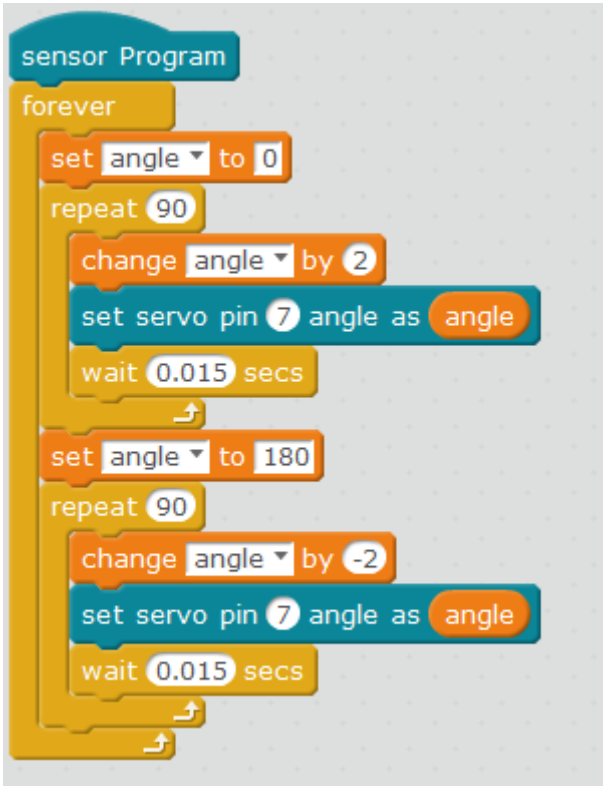
```
    for (int i = 0; i < 10; i++)
    {
      servopulse(angle);
    }
  }
}
```

Experiment Result

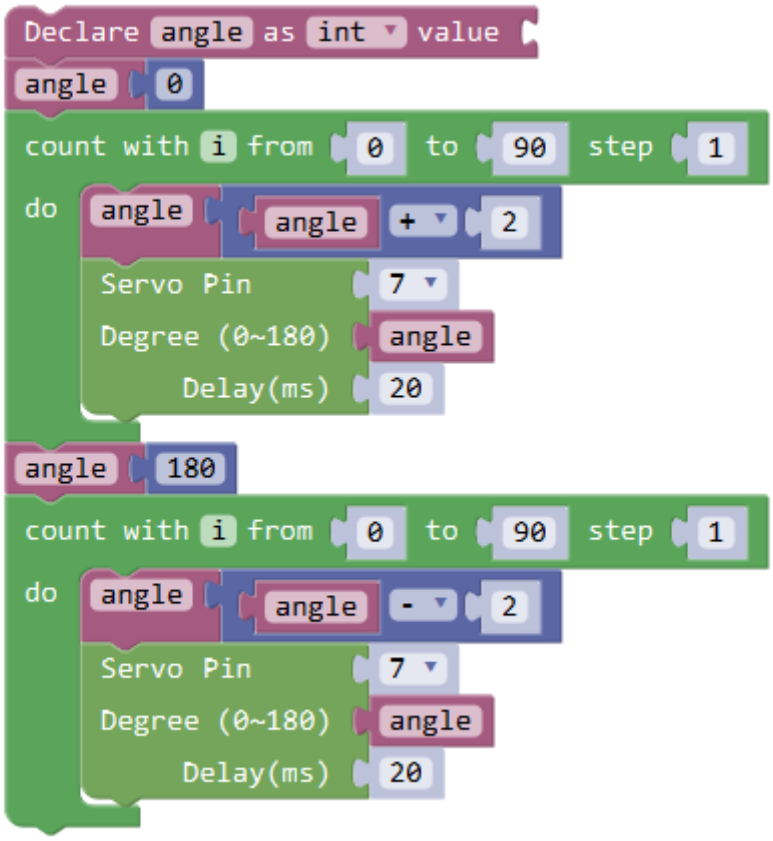


Mblock programming program

MBlock writes the program as shown in the figure below:


sensor Program
forever
set angle to 0
repeat 90
change angle by 2
set servo pin 7 angle as angle
wait 0.015 secs
set angle to 180
repeat 90
change angle by -2
set servo pin 7 angle as angle
wait 0.015 secs

Mixly programming program



MagicBlock programming program