Potentiometer adjusting steering gear Experiment

Introduction of 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. Steering gear is a common name, is actually a servo motor. 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.

The working principle

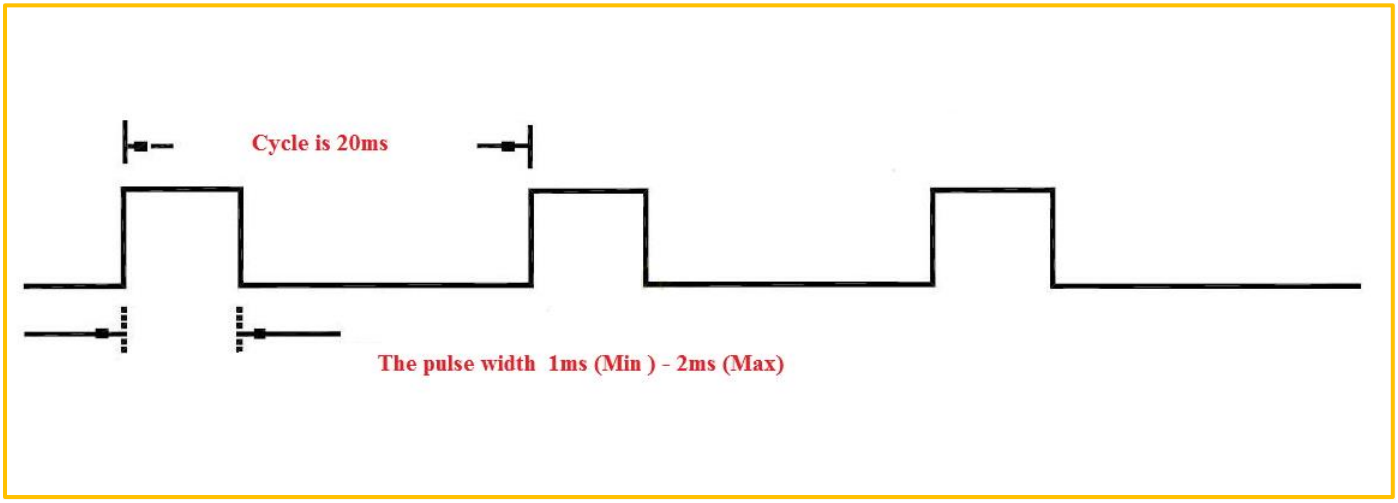
The steering gear control signal enters the signal modulation chip from the channel of the receiver to obtain the dc bias voltage. It has an internal reference circuit that generates a reference signal with a period of 20ms and a width of 1.5ms. The obtained dc offset voltage is

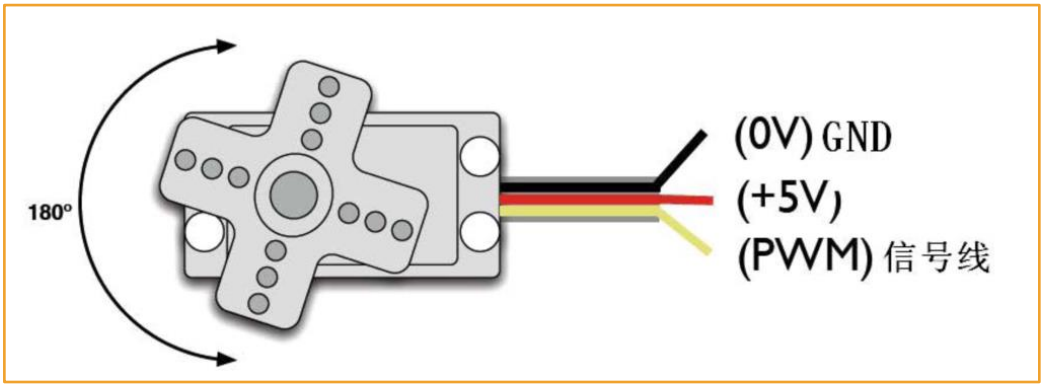
compared with the voltage of the potentiometer to obtain the voltage difference output. Finally, the positive and negative output of the voltage difference to the motor drive chip determines the positive and negative rotation of the motor. When the motor speed is constant, the potentiometer is rotated 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, i.e. 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:

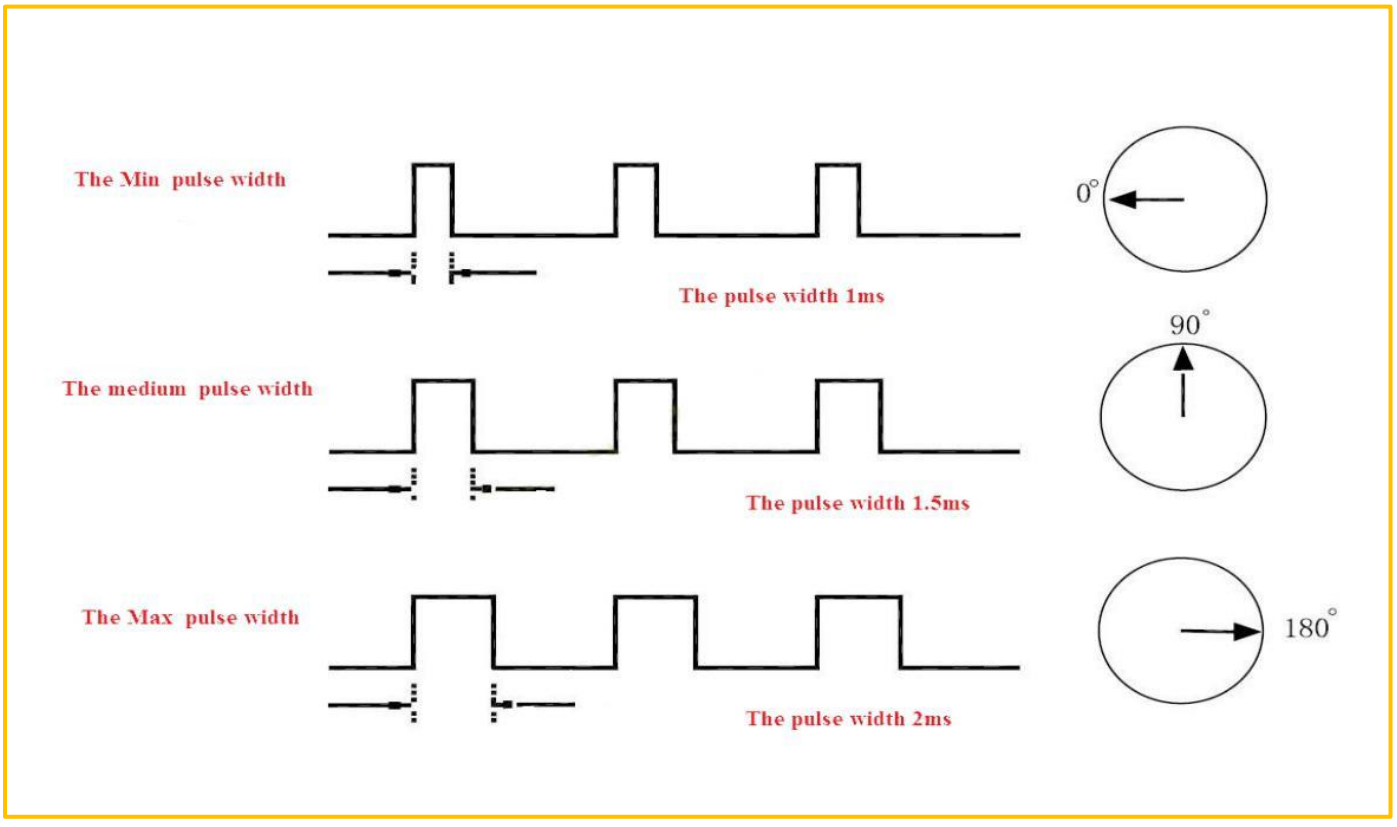
- 0.5ms----------0degree；
- 1.0ms----------45degree；
- 1.5ms----------90degree；
- 2.0ms----------135degree；
- 2.5ms----------180degree；



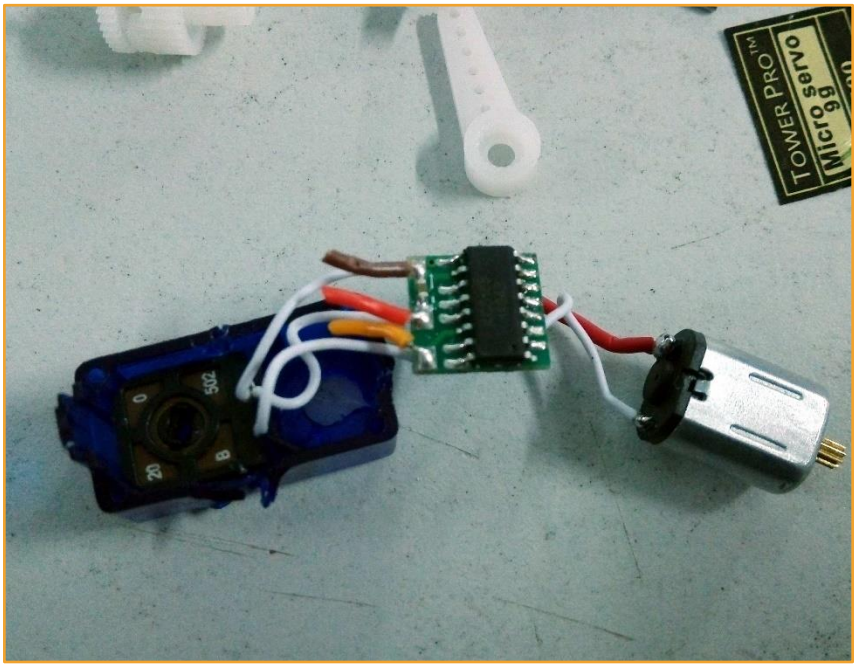


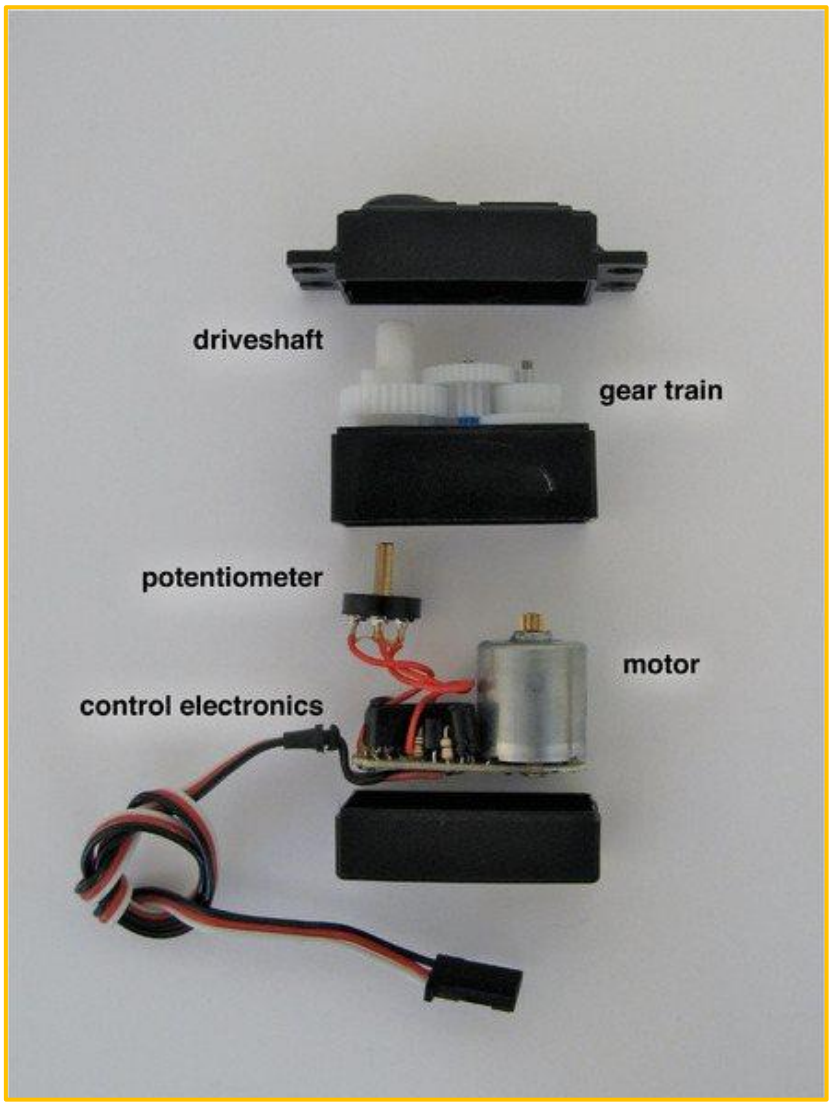
The rotation Angle is generated by a continuous pulse from the control line. This method is called pulse modulation. The length of the pulse determines the rotation Angle of the steering gear. For example, the steering gear rotates to a 1.5 millisecond pulse in the middle position (for a 180 °steering gear, the middle position is 90 °). When the control system issues a command to move the steering gear to a specific position and hold it at an Angle, the effect of external forces does not change the Angle. The Angle will not remain constant until the control system sends out pulses continuously to stabilize the steering Angle.

When the steering gear receives a pulse less than 1.5ms, the output shaft will be taken as the standard middle position and rotated anticlockwise at a certain Angle. When the received pulse is greater than 1.5ms, the output axis rotates clockwise. The maximum and minimum values may be different for different brands of steering gear, or even for different steering gear of the same brand.



Internal Structure of Steering Gear





Experiment Purpose

The aim is to controlling Steering Gear rotation through the potentiometer

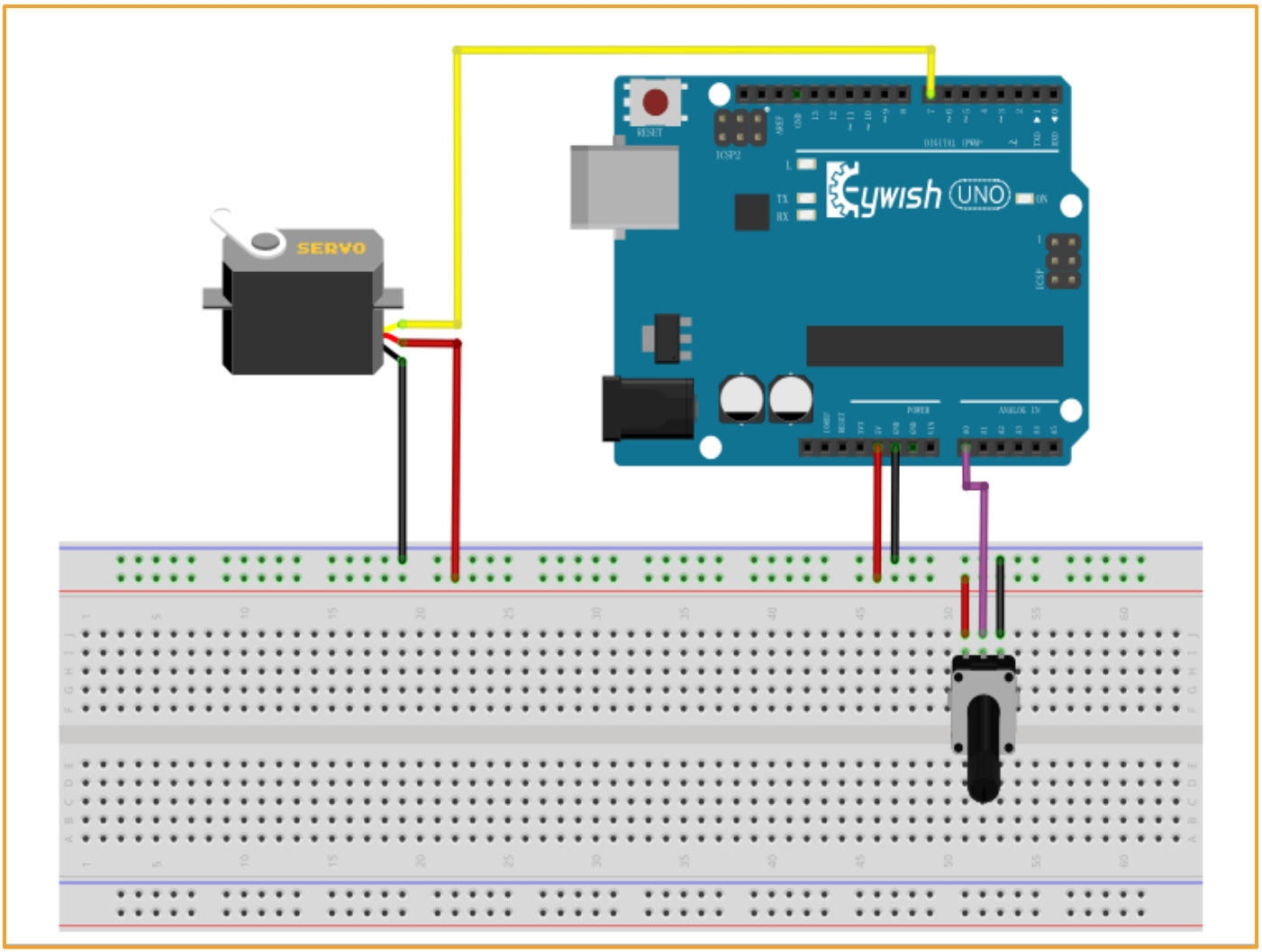
Component List

- Arduino Uno board
- Breadboard
- USB cable
- 10k potentiometer*1
- SG90 steering gear*1
- Several jumper wires

Wiring of Circuit

Connecting the left pin of the potentiometer to 3.3V, the right to GND and the middle to analog interface 0.

Connecting 5V and GND to the steering gear, the signal port to number 7 interface.



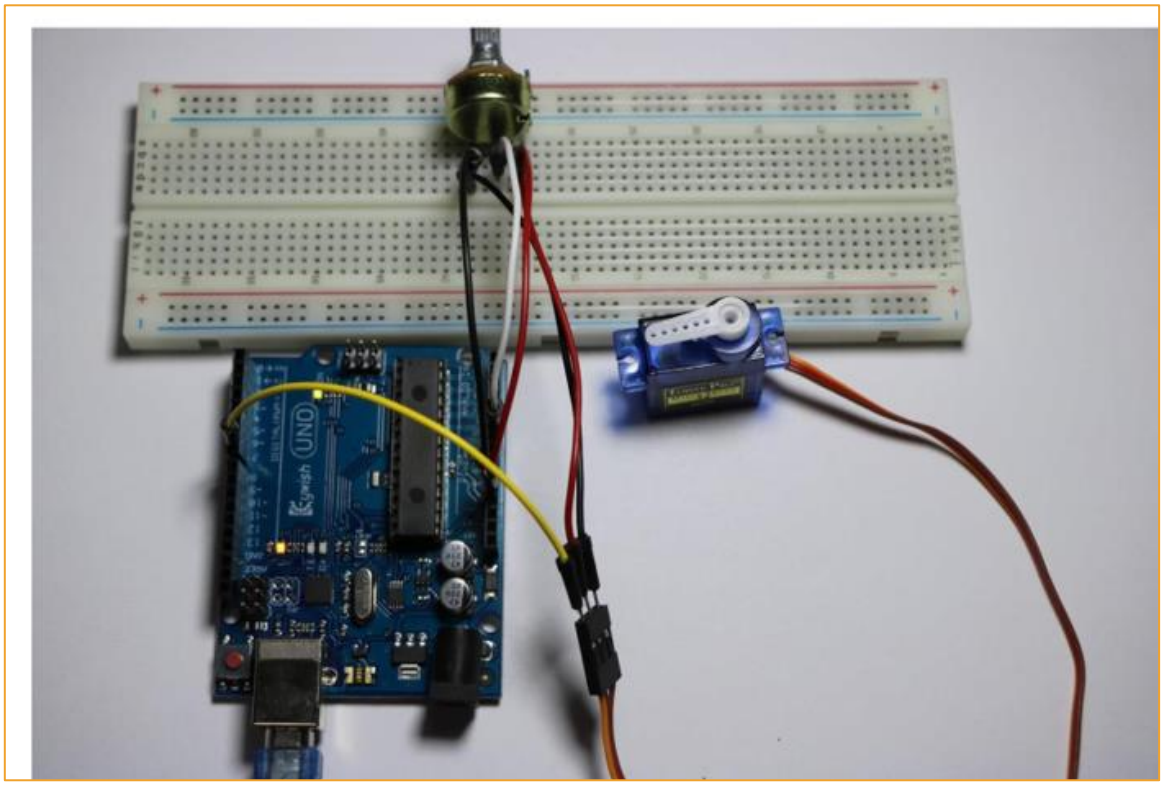
Code

```
#include <Servo.h>
Servo myservo;  // create servo object to control a servo
int readPin = A0;
int servopin = 7;

void setup() {
    pinMode(servopin,OUTPUT);
    myservo.attach(servopin);  // attaches the servo on pin 9 to the servo object
}

void loop() {
    int readValue = analogRead(readPin);
    int angle = readValue / 4;
    myservo.write(angle);              // tell servo to go to position in variable 'pos'
    delay(15);
}
```

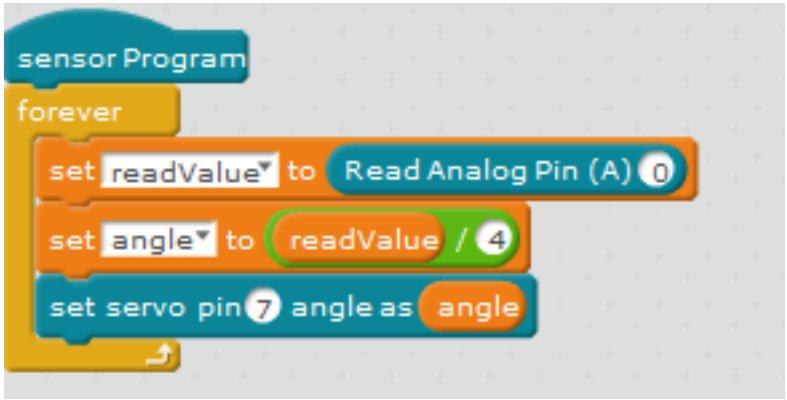
Experiment Result



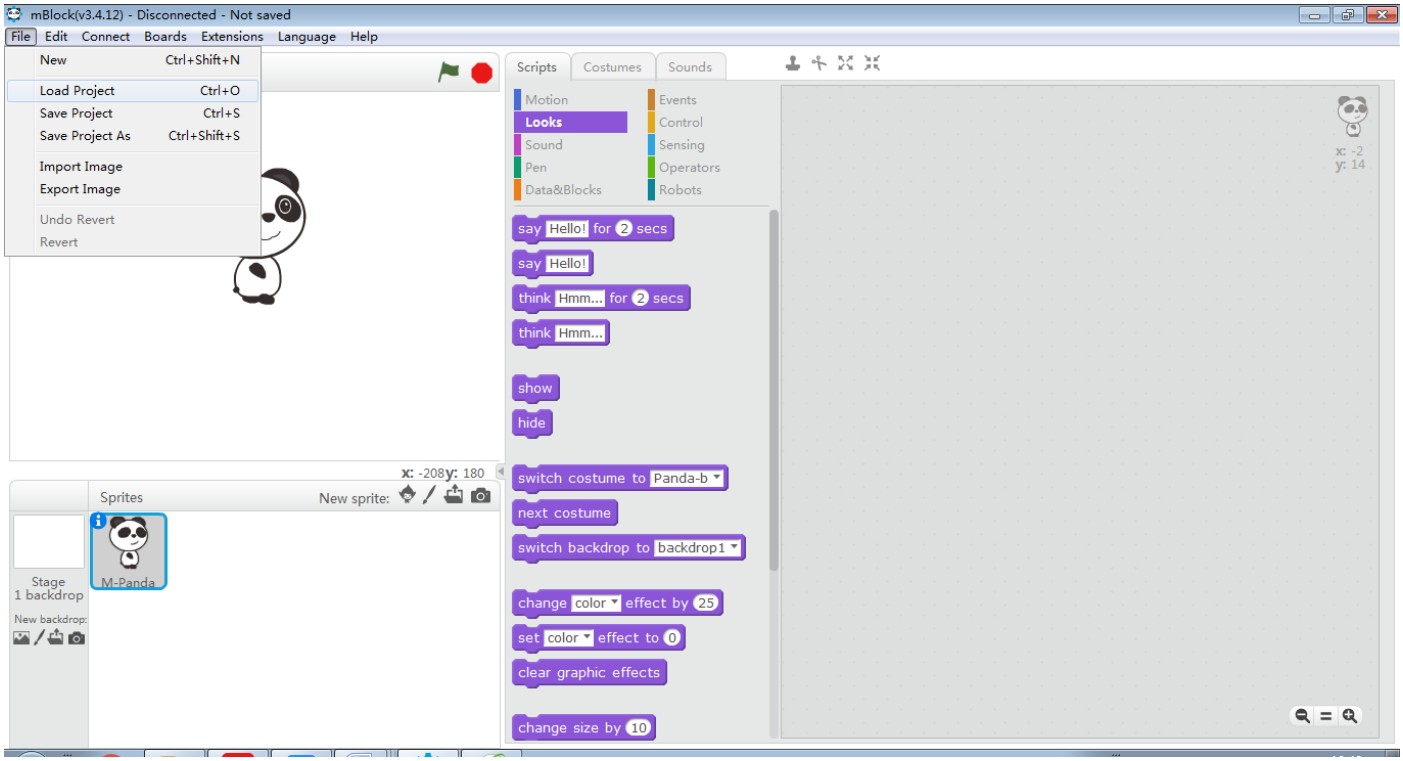
After uploading the program, we can change the Angle of the steering gear by rotating the point locator. The Angle of the steering gear is 0-180 degrees, laying the foundation for the later robotic arm

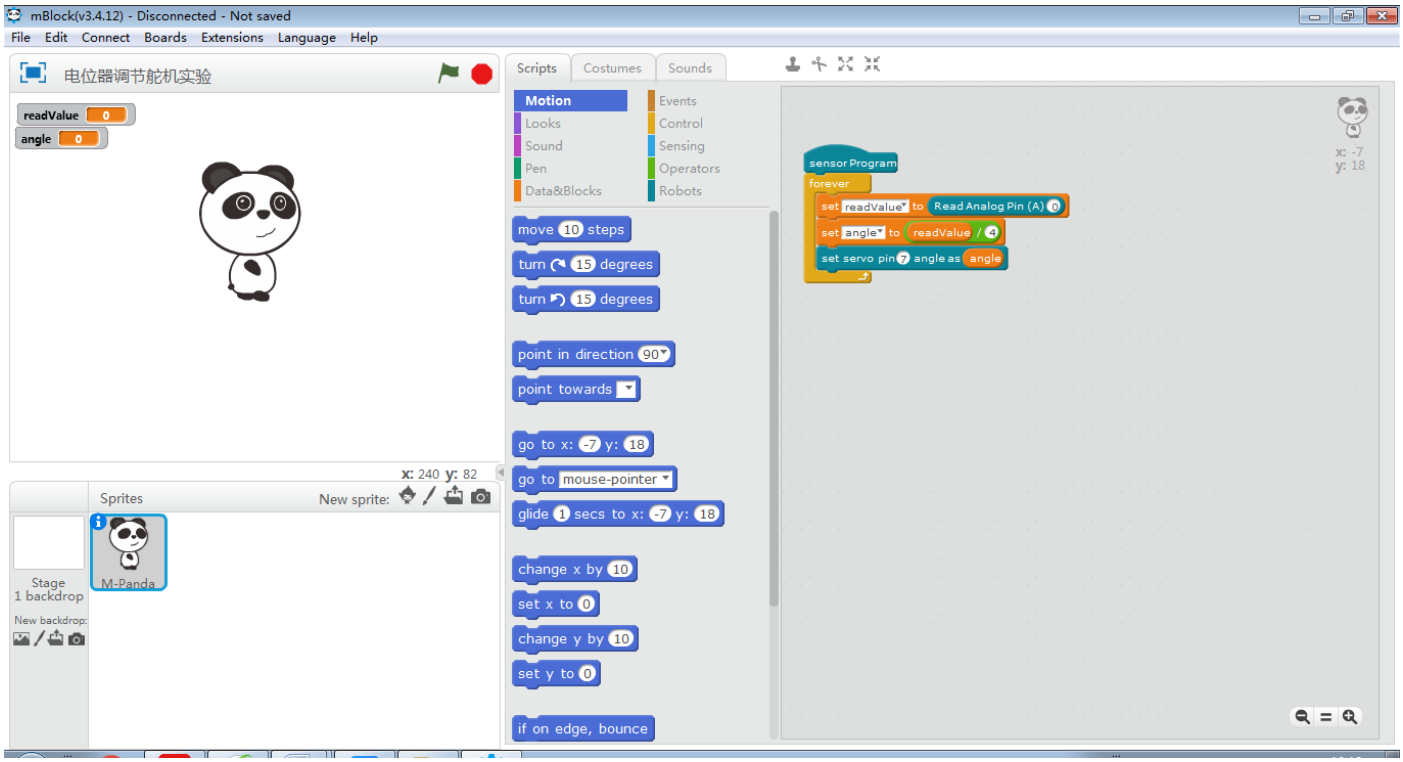
Mblock programming program

The program prepared by mBlock is shown in the figure below:

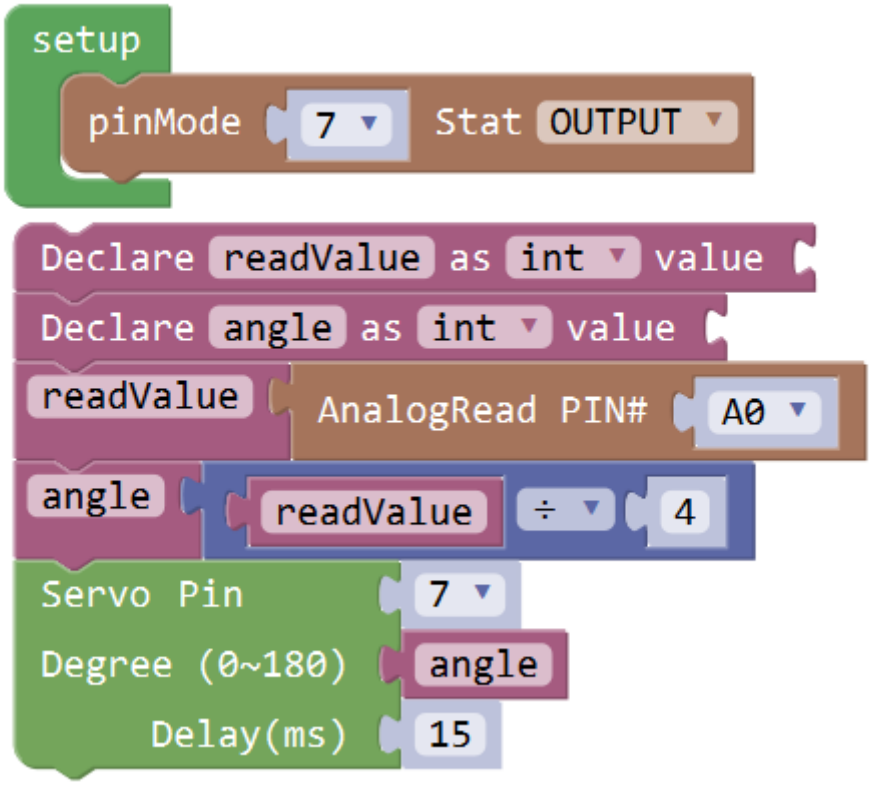


You can also open the program file directly with mblock, which is a. Sb2 file. Here are the steps to open it:


mBlock(v3.4.12) - Disconnected - Not saved
File Edit Connect Boards Extensions Language Help
New Ctrl+Shift+N
Load Project Ctrl+O
Save Project Ctrl+S
Save Project As Ctrl+Shift+S
Import Image
Export Image
Undo Revert
Revert
Scripts Costumes Sounds
Motion Looks Sound Pen Data&Blocks
Events Control Sensing Operators Robots
say Hello! for 2 secs
say Hello!
think Hmm... for 2 secs
think Hmm...
show
hide
switch costume to Panda-b
next costume
switch backdrop to backdrop1
change color effect by 25
set color effect to 0
clear graphic effects
change size by 10
x: -208 y: 180
Sprites
New sprite:
Stage
1 backdrop
New backdrop:
M-Panda
x: -2
y: 14


mBlock(v3.4.12) - Disconnected - Not saved
File Edit Connect Boards Extensions Language Help
电位器调节舵机实验
readValue 0
angle 0
Scripts Costumes Sounds
Motion Looks Sound Pen Data&Blocks
Events Control Sensing Operators Robots
move 10 steps
turn 15 degrees
turn 15 degrees
point in direction 90
point towards
go to x: -7 y: 18
go to mouse-pointer
glide 1 secs to x: -7 y: 18
change x by 10
set x to 0
change y by 10
set y to 0
if on edge, bounce
sensor Program
forever
set readValue to Read Analog Pin (A) 0
set angle to readValue / 4
set servo pin 7 angle as angle
x: 240 y: 82
Sprites
New sprite:
Stage
1 backdrop
New backdrop:
M-Panda
x: -7
y: 18

Mixly programming program



MagicBlock programming program