Arduino 实验记录

罗屹杰，物理学系

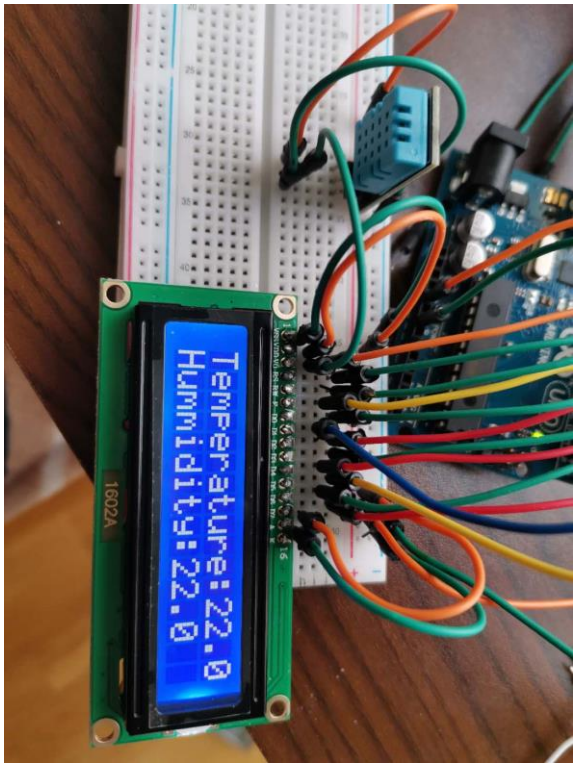
1. 温度计实验记录

实验器材：Arduino Uno 开发板，面包板，电位器，公母接线，DHT11 温湿度传感器，1602LCD 液晶屏

可以实现温度与湿度的输出。主要的代码是调用 DHT11 与 1602LCD 的库函数，需要将这二者的库函数放入指定文件夹内。

发现 1602LCD 液晶屏并未焊接，在家进行了焊接，下图为焊接前后图片

成品图



代码

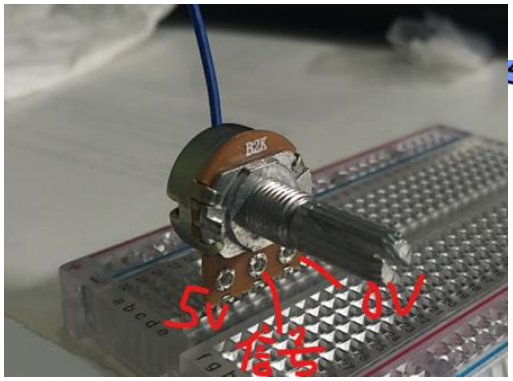
```
#include <LiquidCrystal.h>
#include <DHT11.h>
dht11 DHT11_1;
#define DHT11PIN 13 //传感器接在 13
```

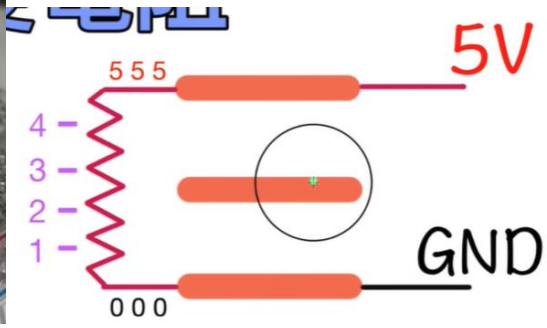
```
LiquidCrystal lcd(12,11,10,9,8,7,6,5,4,3,2);
void setup()
{
   Serial.begin(9600);
   lcd.begin(16,2);
}
void loop()
  {

  DHT11_1.read(DHT11PIN);
    Serial.print("Humidity: "); //  串口输出
    Serial.println((float)DHT11_1.humidity, 2);
    Serial.print("Temperature: ");
    Serial.println((float)DHT11_1.temperature, 2);
  lcd.clear();//  液晶输出
    lcd.print("Temperature: ");
    lcd.print(DHT11_1.temperature);
    lcd.setCursor(0,1);
    lcd.print("Hummidity:");
    lcd.print(DHT11_1.humidity);
delay(1000);       //  重复输出间隔 1s

}
```

2. 可变电阻与类比信号输入

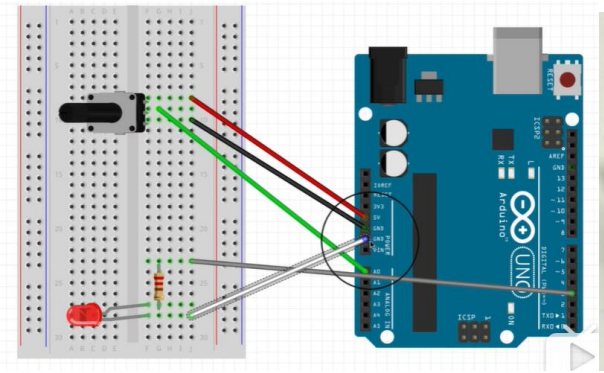
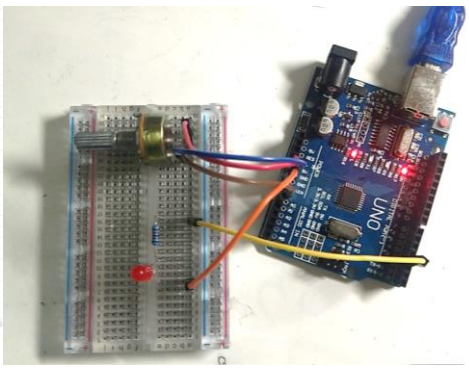
实验器材：Arduino Uno 开发板，面包板，电位器，LED。
类比信号只能从 A 开头的引脚输入！

原理图





接线

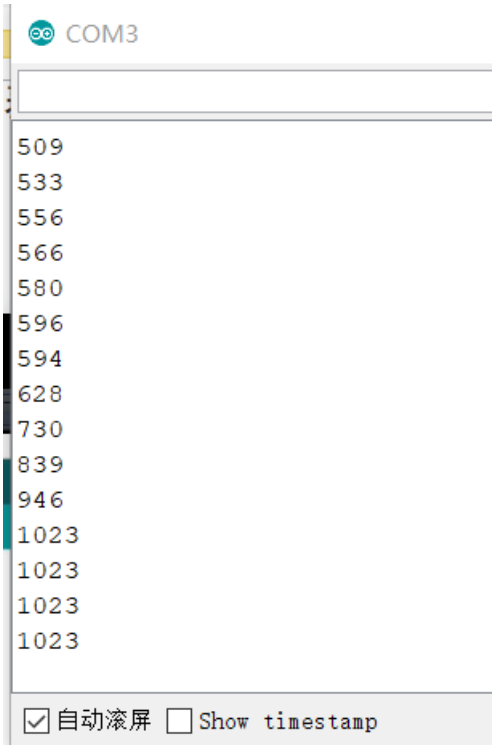
led 接在有小蚯蚓的右方引脚。

（一）改变电阻，输出模拟信号序列

代码：

```
int sensor = A0;
int sensorRead = 0;

void setup() {
  Serial.begin(9600);   //开启序列部输出
}

void loop() {
   sensorRead = analogRead(sensor);   //由 A0 的接脚读取模拟信号
   Serial.println(sensorRead);   //将读出的信号数值传到序列部输出
   delay(200);
}
```

默认数字范围：0-1023 一直读取信号



（二）改变值域

代码

```
int sensor = A0;
int sensorRead = 0;
int newdata = 0;

void setup() {
  Serial.begin(9600);
}

void loop() {
    sensorRead = analogRead(sensor);
    newdata = map(sensorRead, 0, 1023,0,255);    //把 0-1023 对应到 0-255
    Serial.println(newdata);
    delay(200);
}
```

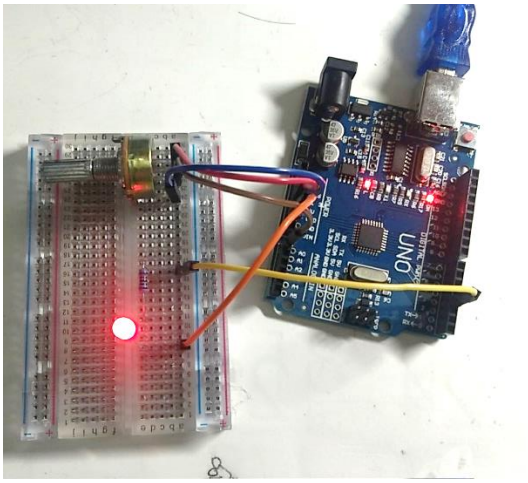
（三）利用模拟信号让 LED 亮起来

注意：亮度输出只能 0-255

脚位必须接在有小蚯蚓的地方！才能控制亮度！

```
int sensor = A0;
int sensorRead = 0;
int newdata = 0;

void setup() {
```

```
  Serial.begin(9600);
}

void loop() {
   sensorRead = analogRead(sensor);
   newdata = map(sensorRead, 0, 1023,0,255);
   Serial.println(newdata);
   analogWrite(3,newdata);   //在 3 号脚位（有小蚯蚓）输出 newdata 大小的电位，
让 led 亮
   delay(200);
}
```



3. 蓝牙操控多个步进电机和舵机

实验器材：Arduino Uno 开发板，面包板，舵机，步进电机，Hc06 蓝牙模块。

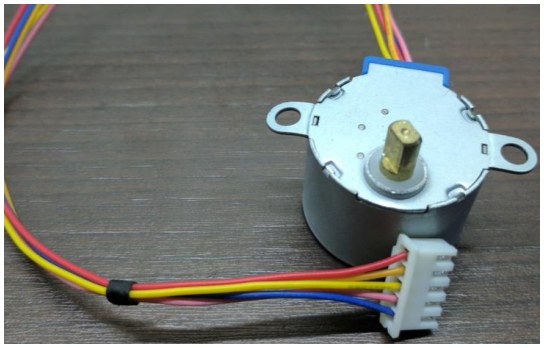
（1） 步进电机学习记录

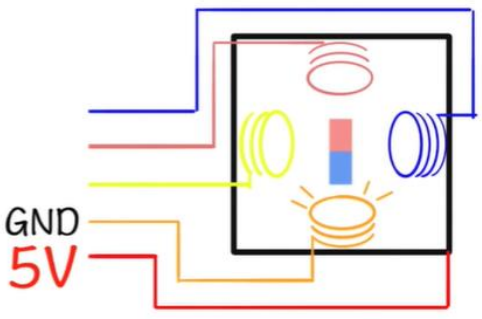
步进马达接线有 4-6 条

材料：步进马达；ULN2003 芯片；外接电源（5V）

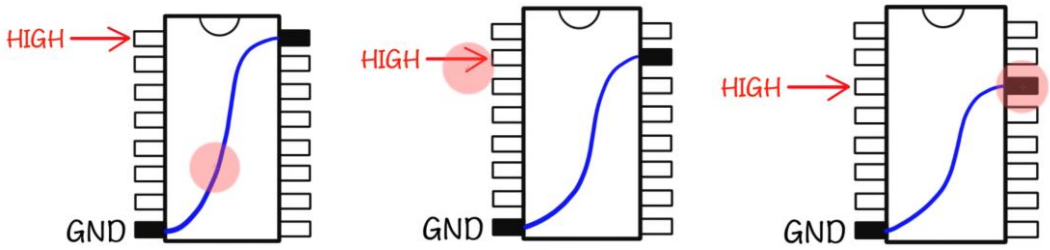
原理：其他四条线轮流接地，改变内部磁场，从而改变磁铁的朝向，带动电机转动.

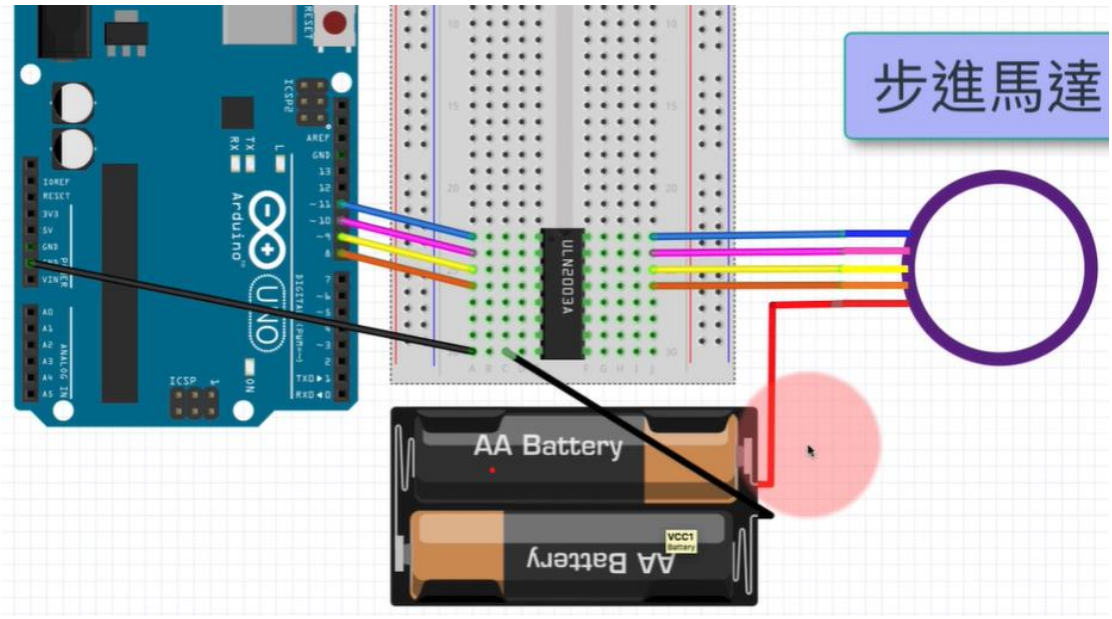
电机转一圈跟磁铁转一圈不等价的，中间是通过齿轮转动的。杠杆原理。传动比。





高电平的接脚对面直接等价于接地，相当于用低压控制高压，保护电路





注意：上面的颜色线的顺序可能需要调节。只要满足顺时针就好了。

代码

```
int apin = 8;    //orange
int bpin = 9;    //yellow
int cpin = 10;    //pink
int dpin = 11;    //blue
int delaytime = 10;    //time

void setup() {
    pinMode(apin, OUTPUT);
    pinMode(bpin, OUTPUT);
    pinMode(cpin, OUTPUT);
    pinMode(dpin, OUTPUT);
}

void loop() {
      digitalWrite(dpin, LOW);        //turn off blue
```

```
    digitalWrite(apin, HIGH);      // turn on orange
    delay(delaytime);
    digitalWrite(apin, LOW);       //turn off orange

    digitalWrite(bpin, HIGH);      // turn on yellow
    delay(delaytime);
    digitalWrite(bpin, LOW);       //turn off yellow

    digitalWrite(cpin, HIGH);      // turn on pink
    delay(delaytime);
    digitalWrite(cpin, LOW);       //turn off pink

    digitalWrite(dpin, HIGH);      // turn on blue
    delay(delaytime);

}
```

（2） 舵机学习记录

例程：Servo

舵机内容：

360 度舵机无法像 180 度舵机一样控制角度，它只能控制方向和速度。

舵机 3 条接线，直流马达只有两条接线（正负极）。

180 度舵机可以转特定角度，直流马达只能纯转动。

SG90

三种颜色接线：

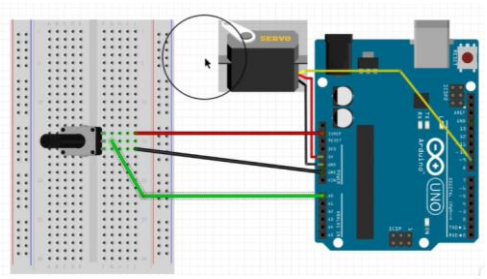
红色：正极 5V

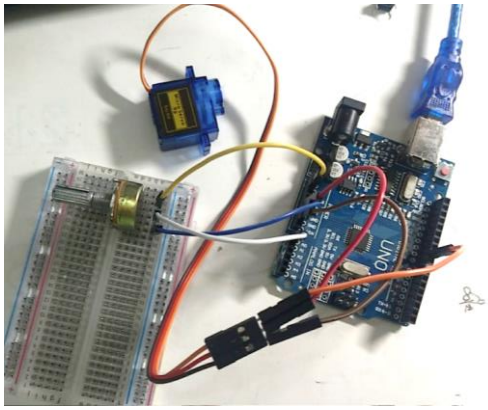
棕色：负极

橘子色：讯号

*接线需要公公线

本次实验中，正极接 5V，负极接地，橘子色线接 9 号引脚（必须要有小蚯蚓）





下面是针对 180 度舵机的实验：

（一）

```
#include <Servo.h>      // import the function

Servo myservo;    //name of the servo

void setup() {
   myservo.attach(9);      // servo uses pin9
}

void loop() {
   myservo.write(90);        //turn to 90 degree (0-180)
   delay(15);
}
```

（二）结合变阻器，使用模拟信号

注意：模拟信号输入引脚是 A 开头的，analogRead；模拟信号输出引脚是有小蚯蚓的

注意：要把模拟信号的 0-1023 对应到 0-180

```
#include <Servo.h>      // import the function

Servo myservo;    //name of the servo
int sensor = 0;
int angle = 0;
```

```
void setup() {
  myservo.attach(9);      // servo uses pin9
}

void loop() {
  sensor = analogRead(A0);
  angle = map(sensor,0,1023,0,180);
  myservo.write(angle);        //turn to 90 degree (0-180)
  delay(15);
}
```

（三）360 度舵机：
通过 x 设定舵机的速度（0 代表一个方向的全速运行，180 代表另一个方向的全速运行，90 则不动）。

```
#include <Servo.h>
Servo myservo;
void setup()
{
  myservo.attach(9);
  myservo.write(90);   //  舵机不动
}

void loop() {}
```

180 度舵机：
通过 x 设定舵机舵盘的角度。

```
#include <Servo.h>

Servo myservo;

void setup()
{
  myservo.attach(9);
  myservo.write(90);   //  在中间位置
}

void loop() {}
```

（3） HC06 蓝牙模块

HC06 蓝牙模块就是取代 com 端口输出与输入，用蓝牙来接受和发送信息。手机蓝牙来控制，这里用的是 Arduino Bluetooth Controller 的 app 进行控制。将 0、1 与 HC06 对应接口相接即可。（此时串口将失效）如果想同时控制，需要使用到 SoftwareSerial 的方式。

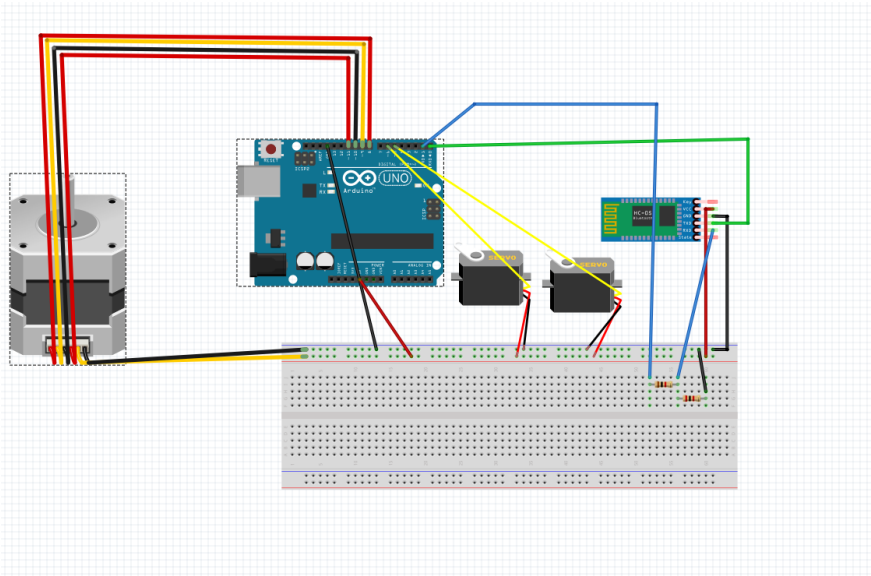
下面是蓝牙控制 arduino 版上的 LED 灯的例程。

```
char serialData;

void setup() {
  Serial.begin(9600);
  pinMode(11, OUTPUT);

}

void loop(){
  if( Serial.available()>0 ){

    serialData =   Serial.read();

    if (serialData == '1' ) {   //接收到点亮 LED 指令
      Serial.print("Got command: ");   Serial.println(serialData);
      Serial.println("LED-ON");

      digitalWrite(11, HIGH);   //点亮 LED 指令
    } else {      //接收到熄灭 LED 指令
      Serial.print("Got command: ");
      Serial.println(serialData);
      Serial.println("LED-OFF");

      digitalWrite(11, LOW);          //熄灭 LED 指令
    }
  }
}
```

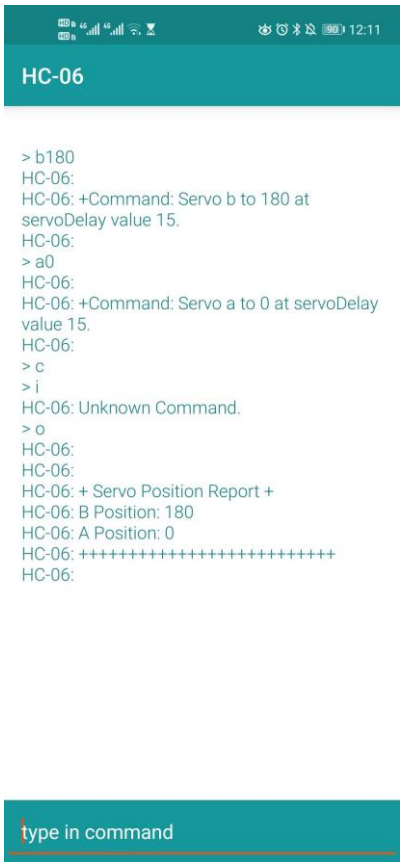
（4） 正式部分

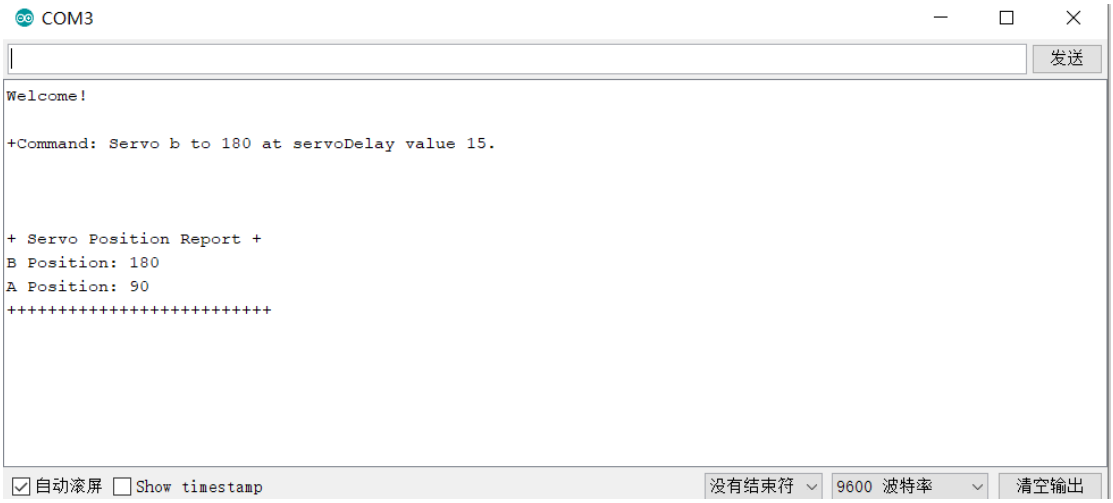
结合之前三个部分，我们可以编出一个程序，实现蓝牙对舵机和步进电机的控制，可以多个舵机步进电机协调工作，这样的好处是，我们可以实现，协同工作，以两个舵机，一个步进电机，我们可以实现一个点在三维空间的运动，也可以进行蓝牙电报机的设定

接线图



程序说明，可以利用手机 app 对舵机实现定角度运动，可以将舵机平稳的调整到想要的角度，可以设置舵机角度上下线，可以控制步进电机正传和反转的周数，可以读取舵机当前状态（输入 o），下图为蓝牙控制的图片以及串口控制的图片，两种控制方式均可。





代码

```
#include <Stepper.h>
#include <Servo.h>
Servo aser, bser;

int aserMin = 0,aserMax =180,bserMin=0,bserMax=180;

// 电机内部输出轴旋转一周步数
const int STEPS_PER_ROTOR_REV = 32;

const int GEAR_REDUCTION = 64;
const float STEPS_PER_OUT_REV = STEPS_PER_ROTOR_REV *
GEAR_REDUCTION;

int StepsRequired;
Stepper steppermotor(STEPS_PER_ROTOR_REV, 8, 10, 9, 11);

int DSD = 15;

void setup(){
  aser.attach(5);
  delay(200);
  bser.attach(6);
    delay(200);                 // 稳定性等待

  aser.write(90);
  delay(10);
   bser.write(90);
  delay(10);

  Serial.begin(9600);
  Serial.println("Welcome!");
}
```

```
void loop(){
   if (Serial.available()>0) {
      char serialCmd = Serial.read();
      armDataCmd(serialCmd);
   }
}

void armDataCmd(char serialCmd){

   if (serialCmd == 'c'||serialCmd == 'd')
      switch(serialCmd)
      { case 'c':
     { StepsRequired    =    STEPS_PER_OUT_REV;
   steppermotor.setSpeed(500);
   steppermotor.step(StepsRequired);
   delay(1000);}
      case 'd':
      { StepsRequired    =    - STEPS_PER_OUT_REV;
   steppermotor.setSpeed(800);
   steppermotor.step(StepsRequired);
   delay(2000);}
     }

//    o  输出机械臂舵机状态信息
  else if (serialCmd == 'a' || serialCmd == 'b' ){
      int servoData = Serial.parseInt();
      servoCmd(serialCmd, servoData, DSD);
         } else {
      switch(serialCmd){
         case 'o':    //  输出舵机状态信息
            reportStatus();
            break;
         default:
            Serial.println("Unknown Command.");
      }
   }
}

void servoCmd(char servoName, int toPos, int servoDelay){
   Servo servo2go;    //创建 servo 对象

   Serial.println("");
   Serial.print("+Command: Servo ");
   Serial.print(servoName);
   Serial.print(" to ");
   Serial.print(toPos);
   Serial.print(" at servoDelay value ");
   Serial.print(servoDelay);
   Serial.println(".");
```

```
  Serial.println("");

  int fromPos;
  switch(servoName){
    case 'a':
      if(toPos >= aserMin && toPos <= aserMax){
        servo2go = aser;
        fromPos = aser.read();
        break;
      } else {
        Serial.println("+Warning: A Servo Value Out Of Limit!");
        return;
      }

    case 'b':
      if(toPos >=    bserMin && toPos <=    bserMax){
        servo2go =    bser;
        fromPos =    bser.read();
        break;
      } else {
        Serial.println("+Warning: B Servo Value Out Of Limit!");
        return;
      }


  }


  if (fromPos <= toPos){    //如果“起始角度值”小于“目标角度值”
    for (int i=fromPos; i<=toPos; i++){
      servo2go.write(i);
      delay (servoDelay);
    }
  }    else { //否则“起始角度值”大于“目标角度值”
    for (int i=fromPos; i>=toPos; i--){
      servo2go.write(i);
      delay (servoDelay);
    }
  }
}

void reportStatus(){
  Serial.println("");
  Serial.println("");
  Serial.println("+ Servo Position Report +");
  Serial.print("B Position: "); Serial.println(bser.read());
  Serial.print("A Position: "); Serial.println(aser.read());
  Serial.println("++++++++++++++++++++++++++");
  Serial.println("");
}
```