<http://wiki.vctec.co.kr/opensource/arduino/servocontrol>

[임베디드 오픈소스](http://wiki.vctec.co.kr/opensource)‎ > ‎[아두이노](http://wiki.vctec.co.kr/opensource/arduino)‎ > ‎

### [아두이노 강좌] 14. 아두이노로 서보모터 제어하기

|  |
| --- |
| 본 게시글에서는 [아두이노](http://vctec.co.kr/front/php/category.php?cate_no=145)를 이용하여 서보모터를 어떻게 제어하는지 살펴보도록 하겠습니다. 서보를 자동으로 앞뒤로 움직이게 만들어보고, 포텐셔미터를 추가하여 서보의 포지션을 제어하도록 하겠습니다.  knob.jpg 브레드보드 레이아웃1 아래의 브레드보드에서는 서보 모터만이 [아두이노](http://vctec.co.kr/front/php/category.php?cate_no=145)에 연결되어 있습니다. 아래와 같이 브레드보드를 셋업합니다.  fritzing_sweep.jpg  서보모터는 3개의 리드선을 가지고 있는데 보통 빨강이 5V, GND는 검정이나 갈색입니다. 나머지  리드선은 제어 리드선으로 보통 오렌지색이나 노랑색을 띄고 있습니다. 제어 리드선을 디지털 핀 9번에 연결합니다.  서보 리드선 끝의 소켓에 점퍼를 연결하여 아래와 같이 브레드보드에 셋업합니다.  sweep.jpg 만약 서보가 오동작을 한다면, 위와 같이 셋업을 하였는데 만약 서보가 이상하게 동작을 한다면, 그것은 서보가 너무 많은 전력을 끌어다 쓰기 때문일 수 있습니다. USB로 아두이노에 전원을 공급하기 때에만 이러한 현상이 발생할 수 있습니다. 보통 모터가 움직이기 시작할때 전력을 많이 쓰기 때문에 이는 [아두이노](http://vctec.co.kr/front/php/category.php?cate_no=145) 보드의 전압을 떨어 뜨릴 수 있으며, 이는 다시 [아두이노](http://vctec.co.kr/front/php/category.php?cate_no=145) 보드를 리셋하게 만듭니다.  만약 이러한 현상이 발생한다면, 캐페이시터(470uF이상)를 GND와 5V 사이에 추가함으로써 해결할 수 있습니다.  fritzing_with_cap.jpg  캐패에시터는 모터가 사용하는 전력 저장소와 같은 역활을 하여 모터가 시작할때 [아두이노](http://vctec.co.kr/front/php/category.php?cate_no=145) 전원뿐아니라 캐패이시터에서 전기를 끌어다 쓸수 있습니다.  캐패이시터의 긴 다리가 양극이며 이 리드선이 5V에 연결되어야 합니다. 음극 리드선은 보통 '-' 심볼로 표시됩니다. [아두이노](http://vctec.co.kr/front/php/category.php?cate_no=145" \t "_blank) 코드 1 아래의 스케치 코드를 [아두이노](http://vctec.co.kr/front/php/category.php?cate_no=145)에 업로드하면 서보가 한방향으로 돌다가 다른 방향으로 다시 도는 것을 확인 할 수 있습니다.  본 스케치 코드는 [아두이노](http://vctec.co.kr/front/php/category.php?cate_no=145) servo 예제 폴더에 있는 sweep 코드를 기반으로 작성된 코드입니다.   1. #include <Servo.h> 3. int servoPin = 9; 4. Servo servo; 5. int angle = 0; // servo position in degrees 6. void setup() 7. { 8. servo.attach(servoPin); 9. } 10. void loop() 11. { 12. // scan from 0 to 180 degrees 13. for(angle = 0; angle < 180; angle++) 14. { 15. servo.write(angle); 16. delay(15); 17. } 18. // now scan back from 180 to 0 degrees 19. for(angle = 180; angle > 0; angle--) 20. { 21. servo.write(angle); 22. delay(15); 23. } 24. }   서보 모터는 펄스에 의해 제어되어 사용하기 쉽습니다. 서보용 [아두이노](http://vctec.co.kr/front/php/category.php?cate_no=145) 라이브러리가 있어 서보에게 단지 동작할 각도를 알려주기만 하면 됩니다.  이러한 서보 라이브러리를 사용하기 위해서는 아래와 같은 코드로 [아두이노](http://vctec.co.kr/front/php/category.php?cate_no=145) IDE에게 서보 라이블러리를 사용할 것임을 알려주어야 합니다.   * #include <Servo.h>   그리고, 서보를 제어하기 위해 servoPin이라는 이름의 변수를 정의하였습니다.   * 1. Servo servo;   위의 코드에서 보면 Servo 타입 servo변수를 정의한 것을 볼 수 있습니다. Servo는 라이브러리에서 제공하는 변수 타입으로 서보사용시 사용합니다. 8개까지의 서보를 정의할 수 있습니다. 만약 두개의 서보를 가지고 있다면, 아래와 같이 정의 할 수 있을 것입니다.   * 1. Servo servo1;   2. Servo servo2;   servo변수에게는 실제로 제어할 서보의 제어핀이 어떤 핀인지를 알려주어야 니다. 아래의 코드로 servo변수에게 제어핀을 알려줍니다.  servo.attach(servoPin);  변수 angle은 서보의 현재각도를 저장하기 위해 사용됩니다. loop함수에서 우리는 두개의 for loop를 사용하는 것을 볼 수 있는데, 첫번째로 각도를 한방향으로 180도까지 증가 시키고, 다음 for 루프에서는 반대방향으로 욺직이게 합니다.  servo.write(angle);  위의 코드는 서보에게 파라메터로 들어온 각도로 위치를 업데이트 하는 명령입니다. 브레드보드 레이아웃2 그럼 두번째 브레드보드 레이아웃을 살펴보겠습니다. 이 레이아웃에는 포텐셔미터가 포함되어 포텐셔미터를 돌려 서보의 위치를 조절할 수 있습니다. 포텐셔미터의 슬라이더 리드를 [아두이노](http://vctec.co.kr/front/php/category.php?cate_no=145)의 A0에 연결합니다.  fritzing_knob.jpg [아두이노](http://vctec.co.kr/front/php/category.php?cate_no=145" \t "_blank) 코드  1. #include <Servo.h> 3. int potPin = 0; 4. int servoPin = 9; 5. Servo servo; 6. void setup() 7. { 8. servo.attach(servoPin); 9. } 10. void loop() 11. { 12. int reading = analogRead(potPin); // 0 to 1023 13. int angle = reading / 6; // 0 to 180-ish 14. servo.write(angle); 15. }   두번째 코드도 첫번째 코드와 비슷한데 potPin이라는 변수가 추가되었습니다. 서보의 위치를 셋팅하기 위해 포텐셔미터로부터 나오는 아날로그 값을 A0핀으로부터 analogRead함수를 통하여 읽어 드리면, 0에서 1023의 값이 읽히게 됩니다.  서보는 오직 180도만 회전할 수 있기 때문에 180도에 맞게 0~1023까지의 값을 스케일 하여야 합니다. 0-1023을 6으로 나누면 0~170이 되어 알맞게 스케일 되었습니다 서보 모터 서보 모터의 위치는 펄스의 길이에 따라서 설정됩니다. 서보는 대략 매 20ms마다 펄스를 받게되는데, 만약 이 펄스가 1ms동안 high이면 각은 0 이며, 1.5ms동안 high이면 중간위치에 위치하게 되고 2ms인 경우는 180도가 되게 됩니다..  servos.png  서버가 움직일 수 있는 끝부분은 제품에 따라 차이가 있는데 많은 서보가 170도까지만 회전을 합니다. 360도까지 회전하는 서보도 있으니 필요하다면 제품을 찾아보는 것도 좋습니다. |

|  |
| --- |
| #include <Servo.h>    int servoPin = 8;  Servo servo;  int angle = 0; // servo position in degrees  void setup()  {  Serial.begin(9600);  servo.attach(servoPin);  }  void loop()  {  Serial.print("start");  // scan from 0 to 180 degrees  for(angle = 0; angle < 180; angle++)  {  Serial.print(angle);  servo.write(angle);  delay(15);  }  // now scan back from 180 to 0 degrees  for(angle = 180; angle > 0; angle--)  {  Serial.print(angle);  servo.write(angle);  delay(15);  }  } |

#include <Servo.h>

//int servoPins[] = { 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 };

Servo servo2;Servo servo3;Servo servo4;

Servo servo5;Servo servo6;Servo servo7;

Servo servo8;Servo servo9;Servo servo10;

Servo servo11;Servo servo12;Servo servo13;

Servo servos[] = {servo2,servo3,servo4, servo5,servo6,servo7, servo8,servo9,servo10, servo11,servo12,servo13};

int angle = 0; // servo position in degrees

int incomingByte = 0;

void setup()

{

Serial.begin(9600);

servo2.attach(2);

servo3.attach(3);

servo4.attach(4);

servo5.attach(5);

servo6.attach(6);

servo7.attach(7);

servo8.attach(8);

servo9.attach(9);

servo10.attach(10);

servo11.attach(11);

servo12.attach(12);

servo13.attach(13);

}

void setAngle(int angle){

Serial.print("servo angle: ");

Serial.print(angle, DEC);

Serial.println();

servo2.write(angle);

servo3.write(angle);

servo4.write(angle);

servo5.write(angle);

servo6.write(angle);

servo7.write(angle);

servo8.write(angle);

servo9.write(angle);

servo10.write(angle);

servo11.write(angle);

servo12.write(angle);

servo13.write(angle);

}

void loop()

{

Serial.println("start");

/\*

setAngle(0);

delay(1000);

setAngle(90);

delay(1000);

setAngle(180);

delay(1000);

\*/

// scan from 0 to 180 degrees

for(angle = 20; angle < 180; angle++)

{

//if(angle%10==0){

Serial.println(angle);

setAngle(angle);

delay(200);

//}

}

/\*

// now scan back from 180 to 0 degrees

for(angle = 180; angle > 0; angle--)

{

setAngle(angle);

delay(20);

}

\*/

}

#include <Servo.h>

//int servoPins[] = { 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 };

Servo servo2;Servo servo3;Servo servo4;

Servo servo5;Servo servo6;Servo servo7;

Servo servo8;Servo servo9;Servo servo10;

Servo servo11;Servo servo12;Servo servo13;

Servo servos[] ={servo2,servo3,servo4, servo5,servo6,servo7, servo8,servo9,servo10, servo11,servo12,servo13};

//Servo servos[] ={servo2};

int angle = 0; // servo position in degrees

int incomingByte = 0;

void setup()

{

Serial.begin(9600);

/\*servo2.attach(2);

delay(100);

servo3.attach(3);

delay(100);

servo4.attach(4);

delay(100);

servo5.attach(5);

delay(100);

servo6.attach(6);

delay(100);

servo7.attach(7);

delay(100);

servo8.attach(8);

delay(100);

servo9.attach(9);

delay(100);

servo10.attach(10);

delay(100);

servo11.attach(11);

delay(100);

servo12.attach(12);

delay(100);

servo13.attach(13);

delay(100);

\*/

}

void setAngle(int angle){

int delayAngle=10;

/\*\*/ setNewAngle(servo2, 2, angle);

delay(delayAngle);

setNewAngle(servo3, 3, angle);

delay(delayAngle);

setNewAngle(servo4, 4, angle);

delay(delayAngle);

setNewAngle(servo5, 5, angle);

delay(delayAngle);

setNewAngle(servo6, 6, angle);

delay(delayAngle);

setNewAngle(servo7, 7, angle);

delay(delayAngle);

setNewAngle(servo8, 8, angle);

delay(delayAngle);

setNewAngle(servo9, 9, angle);

delay(delayAngle);

setNewAngle(servo10, 10, angle);

delay(delayAngle);

setNewAngle(servo11, 11, angle);

delay(delayAngle);

setNewAngle(servo12, 12, angle);

delay(delayAngle);

setNewAngle(servo13, 13, angle);

delay(delayAngle);

}

void setNewAngle(Servo servo,int pin, int angle){

servo.attach(pin);

servo.write(angle);

delay(100);

servo.detach();

}

void loop()

{

Serial.print("start");

setAngle(0);

delay(1000);

setAngle(90);

delay(1000);

setAngle(180);

delay(1000);

// scan from 0 to 180 degrees

/\*for(angle = 0; angle < 180; angle++)

{

setAngle(angle);

delay(10);

}

// now scan back from 180 to 0 degrees

for(angle = 180; angle > 0; angle--)

{

setAngle(angle);

delay(10);

}

\*/

}

#include <Servo.h>

int servoPins[] = { 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 };

Servo servo2;Servo servo3;Servo servo4;

Servo servo5;Servo servo6;Servo servo7;

Servo servo8;Servo servo9;Servo servo10;

Servo servo11;Servo servo12;Servo servo13;

Servo servos[] ={servo2,servo3,servo4, servo5,servo6,servo7, servo8,servo9,servo10, servo11,servo12,servo13};

int angle = 0; // servo position in degrees

void setup()

{

Serial.begin(9600);

}

void setAngle(Servo servo, int pin, int angle){

servo.attach(pin);

servo.write(angle);

delay(100);

servo.detach();

}

void loop()

{

Serial.print("start");

for (int i=0 ; i < 12 ; i++){

int pin = servoPins[i];

Servo servo = servos[i];

// scan from 0 to 180 degrees

for(angle = 0; angle < 180; angle++)

{

Serial.print(angle);

setAngle(servo, pin, angle);

delay(15);

}

// now scan back from 180 to 0 degrees

for(angle = 180; angle > 0; angle--)

{

Serial.print(angle);

setAngle(servo, pin, angle);

delay(15);

}

}

}

영점과 5단위

#include <Servo.h>

int servoPins[] = { 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 };

Servo servo2;Servo servo3;Servo servo4;

Servo servo5;Servo servo6;Servo servo7;

Servo servo8;Servo servo9;Servo servo10;

Servo servo11;Servo servo12;Servo servo13;

Servo servos[] ={servo2,servo3,servo4, servo5,servo6,servo7, servo8,servo9,servo10, servo11,servo12,servo13};

int angle = 0; // servo position in degrees

void setup()

{

Serial.begin(9600);

}

void setAngle(Servo servo, int pin, int angle){

servo.attach(pin);

servo.write(angle);

delay(100);

servo.detach();

}

void loop()

{

Serial.print("start");

//return;

for (int i=0 ; i < 12 ; i++){

int pin = servoPins[i];

Servo servo = servos[i];

//setAngle(servo, pin, 0);

//delay(1000);

//continue;

// scan from 0 to 180 degrees

for(angle = 20; angle < 170; angle++)

{

if(angle%5==0){

Serial.println(angle);

setAngle(servo, pin, angle);

delay(10);

}

}

// now scan back from 180 to 0 degrees

for(angle = 170; angle > 20; angle--)

{

if(angle%5==0){

Serial.println(angle);

setAngle(servo, pin, angle);

delay(10);

}

}

}

}

시리얼로 서보모터 제어

|  |
| --- |
| //zoomkat 11-22-12 simple delimited ',' string parse  //from serial port input (via serial monitor) //and print result out serial port //multi servos added   String readString; #include <Servo.h>  Servo myservoa, myservob, myservoc, myservod;  // create servo object to control a servo   void setup() {   Serial.begin(9600);    //myservoa.writeMicroseconds(1500); //set initial servo position if desired    myservoa.attach(6);  //the pin for the servoa control   myservob.attach(7);  //the pin for the servob control   myservoc.attach(8);  //the pin for the servoc control   myservod.attach(9);  //the pin for the servod control    Serial.println("multi-servo-delimit-test-dual-input-11-22-12"); // so I can keep track of what is loaded }  void loop() {    //expect single strings like 700a, or 1500c, or 2000d,   //or like 30c, or 90a, or 180d,   //or combined like 30c,180b,70a,120d,    if (Serial.available())  {     char c = Serial.read();  //gets one byte from serial buffer     if (c == ',') {       if (readString.length() >1) {         Serial.println(readString); //prints string to serial port out          int n = readString.toInt();  //convert readString into a number          // auto select appropriate value, copied from someone elses code.         if(n >= 500)         {           Serial.print("writing Microseconds: ");           Serial.println(n);           if(readString.indexOf('a') >0) myservoa.writeMicroseconds(n);           if(readString.indexOf('b') >0) myservob.writeMicroseconds(n);           if(readString.indexOf('c') >0) myservoc.writeMicroseconds(n);           if(readString.indexOf('d') >0) myservod.writeMicroseconds(n);         }         else         {              Serial.print("writing Angle: ");           Serial.println(n);           if(readString.indexOf('a') >0) myservoa.write(n);           if(readString.indexOf('b') >0) myservob.write(n);           if(readString.indexOf('c') >0) myservoc.write(n);           if(readString.indexOf('d') >0) myservod.write(n);         }          readString=""; //clears variable for new input       }     }       else {            readString += c; //makes the string readString     }   } } |