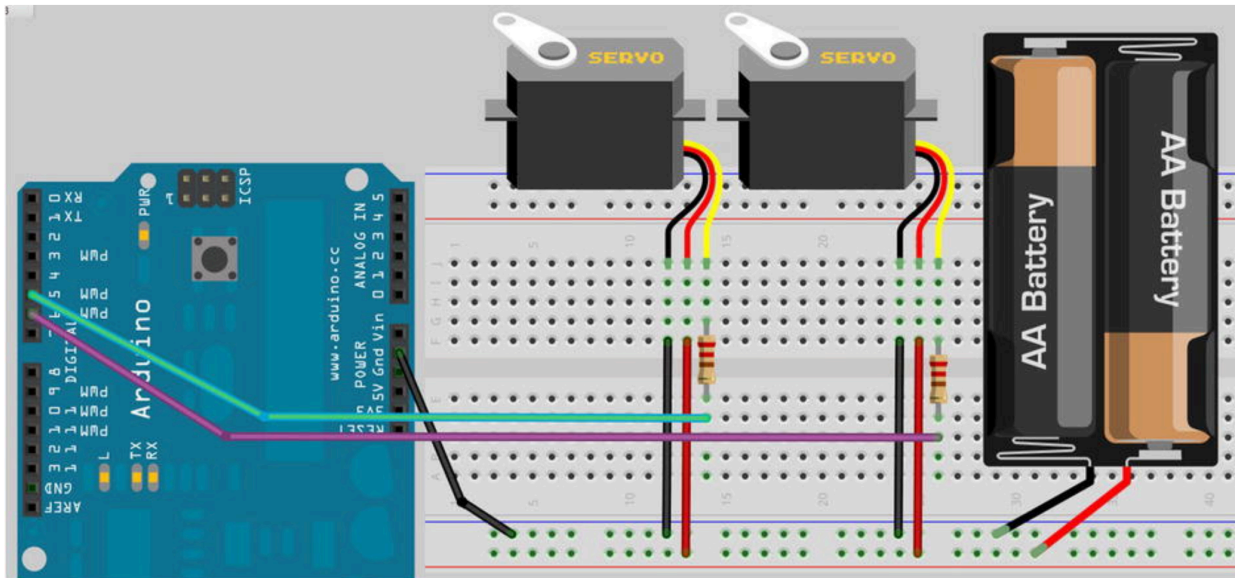


Sooooo many Servos



```
#include <Servo.h>
char buffer[11];
Servo servo1; // Create a servo object
Servo servo2; // Create a second servo object
void setup()
{
  servo1.attach(5); // Attaches the servo on pin 5 to the
  servo1 object
  servo2.attach(6); // Attaches the servo on pin 6 to the
  servo2 object
  Serial.begin(9600);
  while(Serial.available())
  Serial.read();
  servo1.write(90); // Put servo1 at home position
  servo2.write(90); // Put servo2 at home position
  Serial.println("STARTING...");
}
```

```

}
void loop()
{
  if (Serial.available() > 0) { // Check if data has been
    entered
    int index=0;
    delay(100); // Let the buffer fill up
    int numChar = Serial.available(); // Find the string
    length
    if (numChar>10) {
      numChar=10;
    }
    while (numChar--) {
      // Fill the buffer with the string
      buffer[index++] = Serial.read();
    }
    buffer[index]='\0';
    splitString(buffer); // Run splitString function
  }
}

void splitString(char* data) {
  Serial.print("Data entered: ");
  Serial.println(data);
  char* parameter;
  parameter = strtok (data, " ,"); //String to token
  while (parameter != NULL) { // If we haven't reached
    the end
    of the string...
    setServo(parameter); // ...run the setServo function
    parameter = strtok (NULL, " ,");
  }
  while(Serial.available())

```

```

Serial.read();
}
void setServo(char* data) {
if ((data[0] == 'L') || (data[0] == 'l')) {
int firstVal = strtol(data+1, NULL, 10); // String to
long integer
firstVal = constrain(firstVal,0,180); // Constrain
values
servo1.write(firstVal);
Serial.print("Servo1 is set to: ");
Serial.println(firstVal);
}
if ((data[0] == 'R') || (data[0] == 'r')) {
int secondVal = strtol(data+1, NULL, 10); // String
to long integer
secondVal = constrain(secondVal,0,255); //
Constrain the values
servo2.write(secondVal);
Serial.print("Servo2 is set to: ");
Serial.println(secondVal);
}
}
}

```

To run the code, open up the serial monitor window. The Arduino will reset, and the servos will move to their central locations. You can now use the serial monitor to send commands to the Arduino.

The left servo is controlled by sending a L and then a number between 0 and 180 for the angle. The right servo is controlled by sending an R and the number.

You can send individual commands to each servo or send both commands at the same time by separating the commands with space or comma, like so: L180 L45 R135 L180,R90 R77 R25 L175 This is a simple example of how you could send commands down a wire to an Arduino-controlled robot arm or an animatronic toy. Note that the serial commands don't have to come from the Arduino serial monitor