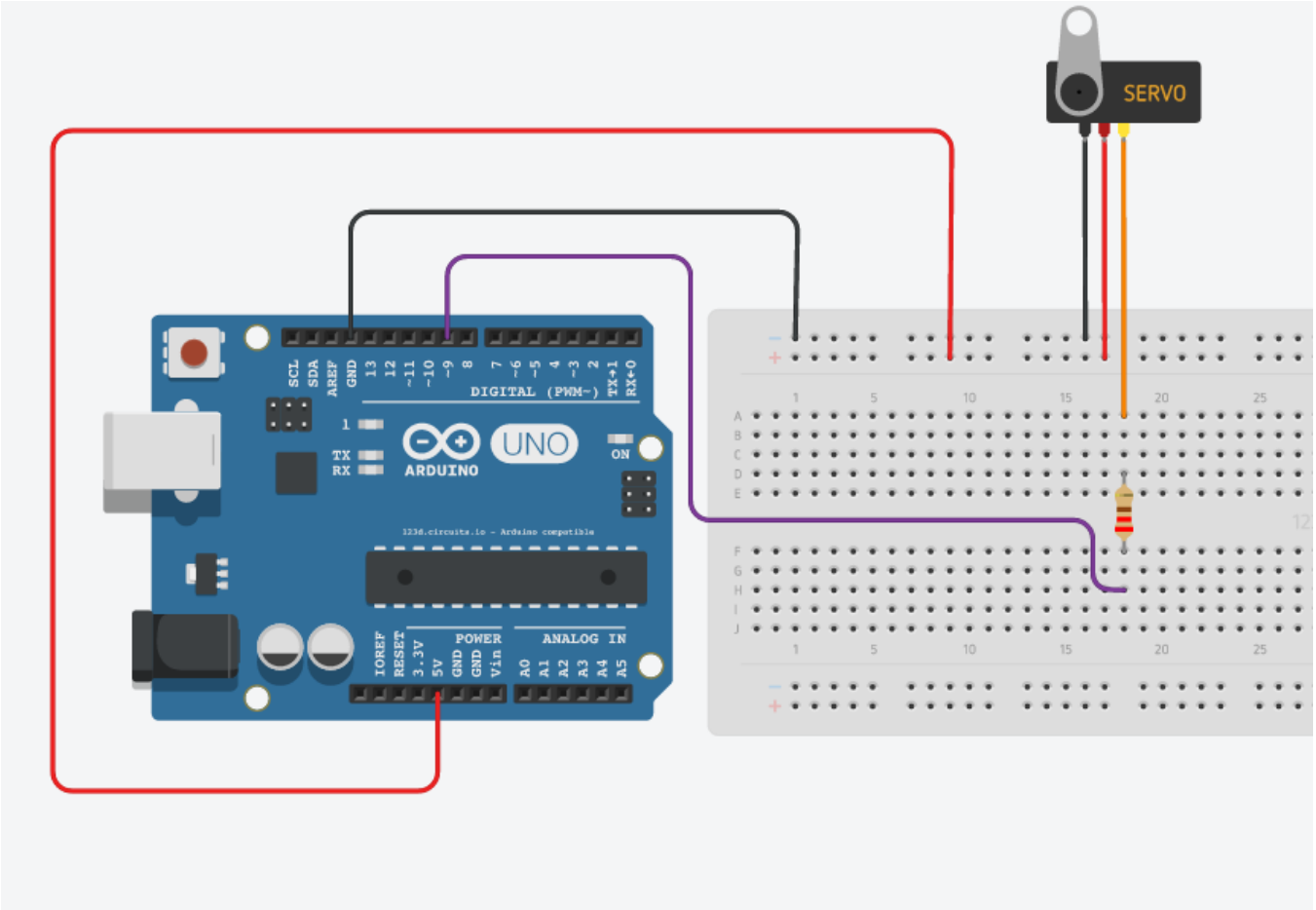
INTRODUCTION TO SERVO MOTORS

Required Materials for each student: (1) servo motor, (1) 220Ω resistor, (1) AAx4 batteries & holder or power supply, (1) breadboard, (1) Arduino microcontroller, jumper wires

We will be using a servo motor to hold the marble of your Rube Goldberg Machine in place. Then, when the preceding group's marble strikes your piezo element, your Arduino will tell your servo motor to rotate 90 or 180 degrees so that the lever that holds your marble in place on the motor shaft moves out of the way, releasing your marble. This activity introduces you to a basic servo motor, and simple controls for one.

On the virtual circuits simulator, set up the following circuit and arrangement.



Upload the following sketch into the Arduino and observe what occurs.

```
/* INTRO TO SERVO CONTROL
   *
   * This program utilizes the Arduino Servo Library to control a single servo motor.
   *
   */
```

```
#include <Servo.h>                  // Include the Arduino servo library when running this program.
Servo servo1;                       // Create a servo object called "servo1".
int angle1 = 0;                     // Define an integer beginning angle of zero.
int angle2 = 180;                   // Define an integer ending angle of 180.

void setup()  {
    servo1.attach(9);               // Attaches the servo object just identified to Pin 9.
    servo1.write(90);               // Put servo at central position of 90 degrees to start.
    delay(1000);                    // Delay for 1 second after placing servo in its mid-position.
}

void loop()   {
    servo1.write(angle1);           // Write the beginning angle to the servo (0 degrees)
    delay(1000);                    // Delay of 1000ms to allow servo to reach position
    servo1.write(angle2);           // Write the ending angle to the servo (180 degrees).
    delay(3000);                    // Delay of 1000ms to allow servo to reach position
}
```

Next, try this code.

```
/* Sweep
*
*This code demonstrates how you can smoothly rotate a servo, rather than have it go to a position as fast as it can.
*/

#include <Servo.h>

Servo myservo;                      // create servo object to control a servo
int pos = 0;                        // variable to store the servo position

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

void loop() {
    for (pos = 0; pos <= 180; pos += 1) {       // goes from 0 degrees to 180 degrees in steps of 1 degree
        myservo.write(pos);                     // tell servo to go to position in variable 'pos'
        delay(15);                              // waits 15ms for the servo to reach the position
  }
     for (pos = 180; pos >= 0; pos -= 1) {      // goes from 180 degrees to 0 degrees
         myservo.write(pos);                    // tell servo to go to position in variable 'pos'
         delay(15);                             // waits 15ms for the servo to reach the position
  }
}
```

Finally, gather the physical materials necessary to build this actual circuit and run both of these sketches on your circuit. Show your instructor your second working circuit, and be prepared to describe how the two are different.