

# Lab 3-Run a Servo

## Learning Outcomes:

- Learn about servos and pulse width modulation (PWM) signals
- Learn about voltage regulators
- Learn how to spin a servo

## Background:

A servo is a low speed, high torque motor. They are fairly simple to use. Most of these servos are useful because they can determine their angular position using a potentiometer ([link](#)). These cheap lightweight control options are used in robotic and hobby aircraft applications where position is a concern. For example, to control aircraft control surfaces. The downside to a potentiometer servo is that it can't spin more than a certain angle (usually 180deg). The servos that we'll use are called continuous rotation servos. They remove the potentiometer and essentially operate the same as a motor. However, that comes at the cost of not being able to determine angular position.



9g servo. Image courtesy of [Amazon](#).

These servos are simple to use which is why they are used in many applications including hobby aircraft, robotics, drones, etc. We will be using these as the motors for our AutoNoMouse. To operate the servos, we only need 3 wires per servo. 1) black for ground, 2) red for 5V power, and 3) orange for a signal (this signal is called pulse-width modulation [Pulse-width modulation, Servo control](#)).

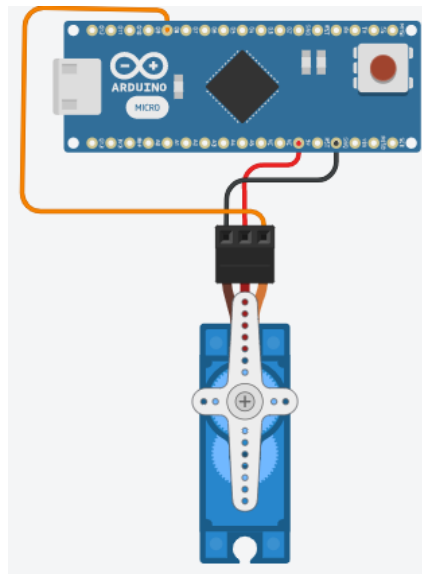
The objectives for today's lab are to:

- 1) Provide the servo with power and a signal from the Arduino Nano
- 2) Provide the servo with a signal from the Arduino Nano but power from an external source

## Part 1 Power Servo from Arduino Nano:

We will first test out our servo using the 5V pin from our Arduino Nano. This is not good for high current applications because running too much current through the Arduino Nano can break it...so don't put too much torque on the servo or run it for too long. For a simple test to make it move clockwise or counter-clockwise it'll be fine.

1. Create the circuit shown below
  - a. Connect 5-volt pin (5V) to red
  - b. Connect ground pin (GND) to brown
  - c. Connect Pin 9 (D9) to orange
2. Copy/paste the code into Arduino IDE from GitHub, and upload it to your Arduino Nano
3. Watch the servo spin clockwise
4. Comment out the first void loop() in the code and uncomment the second (you comment things by adding `//` in front the line of code)
5. Upload your code again and watch your servo spin the other direction.
6. BONUS QUESTION: Continuous servos are controlled in Arduino using the Servo.h library. 0 makes the servo spin fastest in the clockwise direction and 180 makes the servo spin fastest in the counterclockwise direction. 90 makes the servo not spin. (These values might change slightly depending on your servo). Play around with these values to make them spin slower or faster.



*Circuit created in Tinkercad*

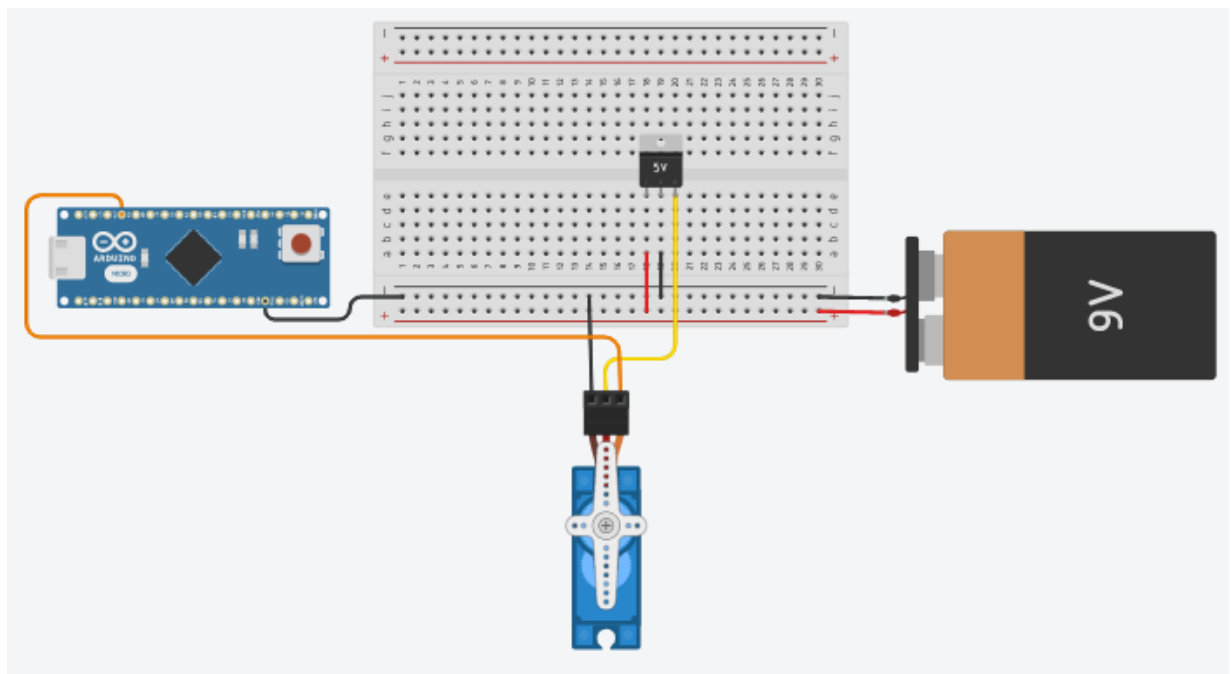
## Part 2 Power a Servo using a Voltage Regulator:

Our 9V battery could provide power directly to the servo but the 9V current wouldn't be good for our 5V rated servo. To limit the voltage, we are going to use a voltage regulator. Like its name suggests, a voltage regulator restricts the voltage to ensure it doesn't pass a certain threshold.

Different regulators are made for different voltage levels. You might have a 3.3, 5V, etc. We will pass the 9V current from the battery through a 5V voltage regulator which will output 5V current to spin the servos.

The voltage regulator we're using has 3 pins. 1 for output, 2 for ground, and 3 for input voltage. When working with voltage regulators, make sure to consult their datasheets to ensure you use them properly. If you pass too high a current through a voltage regulator you can ruin it. Another thing to remember with voltage regulators is that they can't increase voltage. The input voltage must be the same or higher than the desired output voltage. They also will get hot after being used for some time.

1. Create the circuit shown below
2. Use the same code as before (the control signal—orange wire—from the Arduino Nano is the same we're just powering it with something else now)
3. Do the same thing as before (spin the servo clockwise, spin the servo counterclockwise)
4. BONUS QUESTION: how much current is the AN7805 voltage regulator designed to withstand? (Lookup its datasheet online)



*Circuit created in Tinkercad*

### Extra Credit:

Put a wire between power and VIN pin on the Arduino while the battery is plugged in. Your Arduino will then be powered by the battery and can run the code while it's not plugged into the computer.