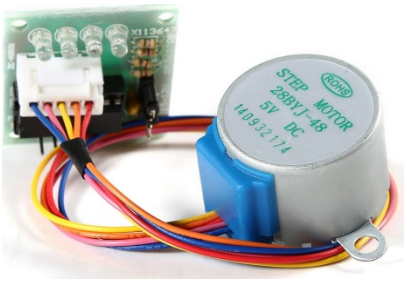
Chapter 12 – Adding Motors: Steppers and Servos

April, 2020
David Ackmann – Gateway Division NMRA

Want to make something rotate on your model railroad? You need a low voltage motor, probably a stepper motor or a servo motor. If you visit AMAZON and do a search on "Arduino Motors", you will get over 1,000 hits. Many of the hits are relevant to applications other than model railroading. The ones that I use are generally classified as "Stepper Motors" and "Servo Motors". So, what's the difference?

- Servos have absolute positioning; if you tell one to go to a specific position, like 0°, it will always do so. Steppers have relative positioning; if you tell one to rotate a certain number of degrees, it will do so, but it cannot remember where any specific position is.
- Servos are generally limited to 180° of travel, sometimes only 90°, whereas steppers have unlimited rotation capability.
- Rotational granularity:
- Servos generally require a bit less space under the layout.
- Servos require only 1 control pin from an Arduino, plus 5V power and Ground, but only 6 digital pins can generate the "Pulse Width Modulation" ("PWM") required by a servo, pins 3, 5, 6, 9,10 and 11. Steppers require 4 digital pins (any will work) plus a ground pin.
- Servos come with either plastic or metal gears. Steppers that I have found come only with plastic gears.
- Servos with plastic gears (model SG90) are about $3.50 when bought in a 2-pack, and about $1.80 when bought in a 10 pack. Servos with metal gears (model MG90-S) are about $3.50 each when bought in a 2-pack or a 6-pack.
- Steppers require and generally come with a controller board. Servos do not require one.



Stepper Motor

Servo Motor

That's not to say that these are the only types I use for Arduino projects; for the Carousel project I used a Hankscraft 12VDC, 4RPM motor to drive the carousel, but it's current draw was more than I wanted to require of an Arduino, so I powered it through a relay and controlled it through the Arduino. But I found that stepper motors with their plastic gears were prone to stripping, and since I don't use motors often, and only then with limited rotational requirements, (only in the Lumberjack project, at least for now), I have decided to use servo motors with metal gears.

Although both types of motors usually come with plastic attachment arms (aka: "horns"), I found that I sometimes wanted something stronger than came with the motor (or I lost the ones that did), and I found that arms from DUBRO fit the stepper motors I use. Their model 675 fit the stepper motor described above (AMAZON ASIN: B0006MZR5A). The servo motor I purchased was fairly inexpensive, and when I emailed tech support for the dimensions of the servo arm, they said they were the only ones who sold arms that fit, but wouldn't tell me how to get replacements. I think it is 4.8mm, 20 tooth, but I am not sure.

I did discover that I could take Evergreen Polystyrene tubing, part number 227 and jam it over the spline, and it was tight enough to hold. I inserted some Evergreen 225 tubing to reduce the interior size. Once I did so, the inside Diameter was smaller than the Evergreen 213 rod (0.100") which I was using with my lumberjack (I thought I was using 0.125", I need to check on that). The hole I drilled in the diorama was 0.128" (#30).

I do want to document an anomaly I discovered when using servo motors in conjunction with the DFPlayerMini MP3 player. In the Lumberjack project, I initially decided to place the motor's "attach" function in the SETUP routine (this function logically associates a specific servo instance to a specific Arduino pin). But I discovered that every time I made a call to the MP3 player, the motor would rotate slightly and quickly, or what I call "jitter". This was unacceptable to me, and I could find no way in my hardware setup to avoid the problem. Eventually I discovered that if I removed the motor's "attach" function from the SETUP routine and placed it in the LOOP, right before I wanted to use the motor, and then used the motor's "detach" function immediately after I was done using the motor, the "jitter" went away. To save you some effort, I did try to power the servo external to the Arduino, but the problem persisted. Only when I decided to turn the motor off and on by using "attach" and "detach" in the LOOP did the problem go away.