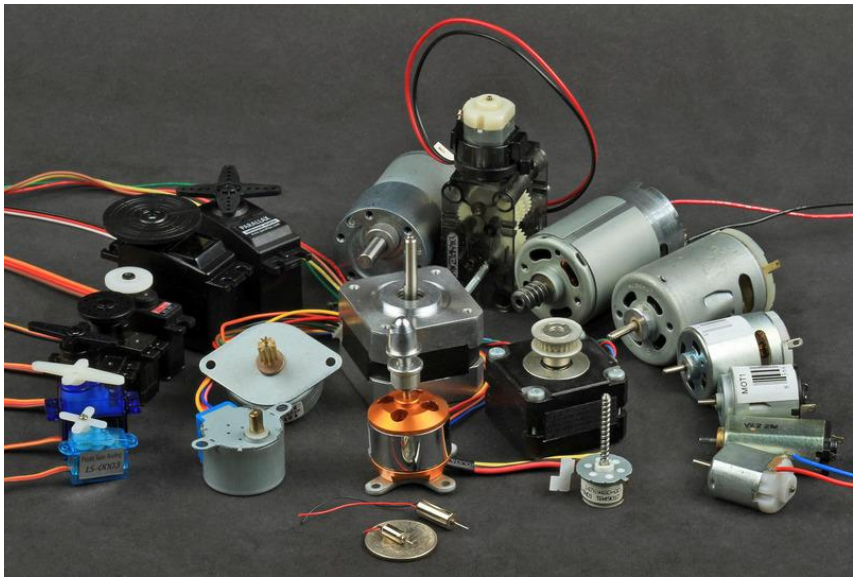
Mechatronics Engineering and Automation
Faculty of Engineering, Ain Shams University
MCT-151, Spring 2015
Lab-4: Electric Actuators

Ahmed Okasha, Assistant Lecturer

okasha1st@gmail.com

Objective

- Have a general overview about actuators.
- Understanding DC-Motor and how to control its direction and speed.
- Understanding how to control the Servo Motor.
- Understanding how to control the Stepper Motor.



Actuators

- An Actuator is a type of motor for moving or controlling a mechanism or system.
- It is operated by a source of energy, typically electric current, hydraulic fluid pressure, or pneumatic pressure, and converts that energy into motion.
- An actuator is the mechanism by which a control system acts upon an environment.

DC-Motors

- A traditional DC motor uses direct current to create magnetic force, which turns an output shaft.
- When the polarity of the DC voltage is reversed, the motor reverses its direction of rotation. Usually, the force created by the motor is equal in either direction.
- DC motors are widely used in robotics because of their small size and high energy output. They are excellent for powering the drive wheels of a mobile robot as well as powering other mechanical assemblies.

DC-Motor Selection

- **Voltage:**

Typical DC motors are rated from about 6V-12V. The larger ones are often 24V or more.

- **Current:**
- The operating current: This is the average amount of current the motor is expected to draw under a typical torque.
- The stall current: This is when you power up the motor, but you put enough torque on it to force it to stop rotating.

DC-Motor Selection

Torque:

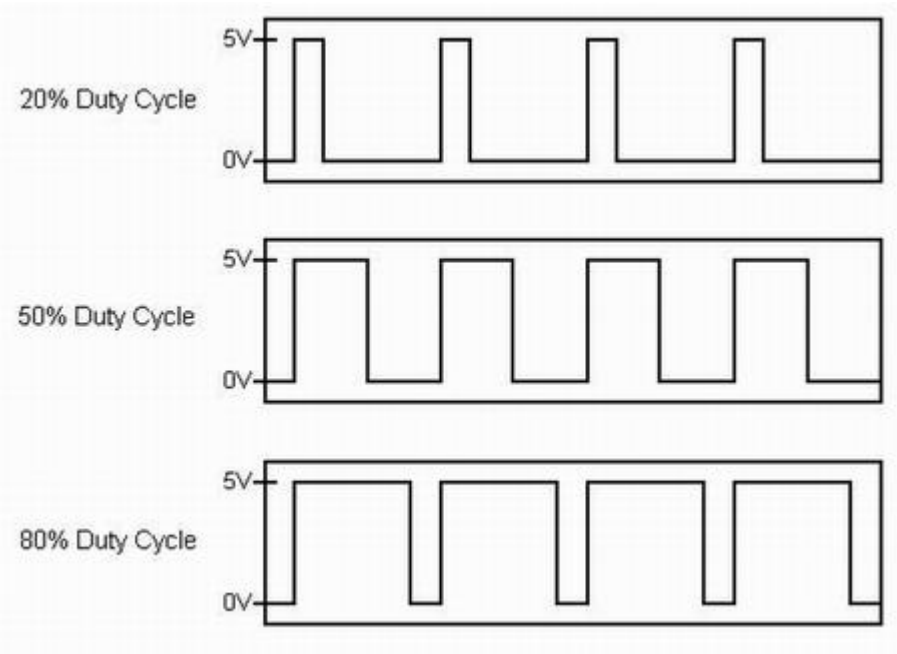
- The operating torque: This is the torque the motor was designed to give. Usually it is the listed torque value.
- The stall torque: This is the torque required to stop the motor from rotating.

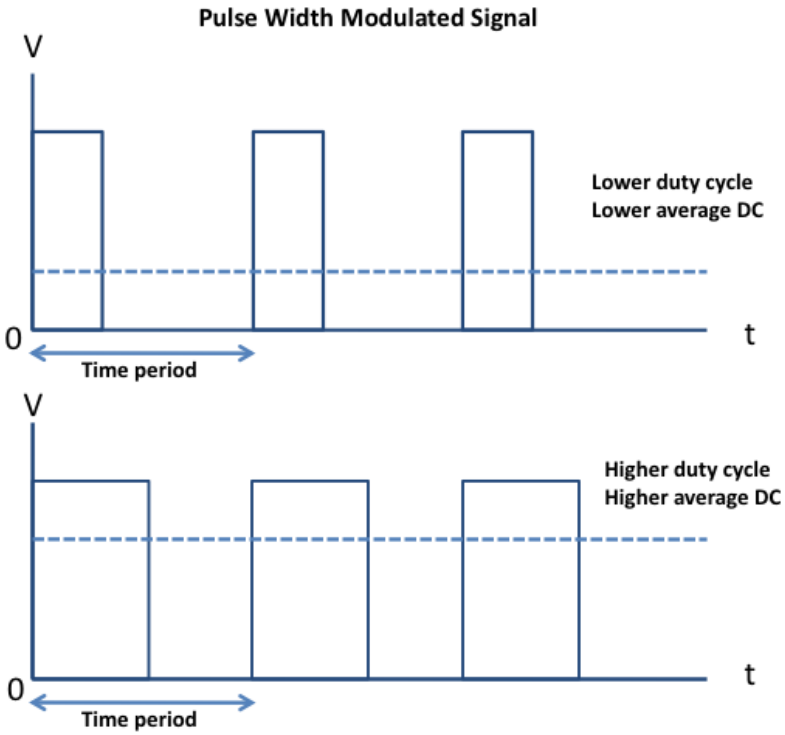
DC-Motor Speed Control

- Most Microcontrollers cannot generate analog output. However, this is tackled by a technique called PWM.
- Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means.
- Digital control is used to create a square wave, a signal switched between on and off.
- This on-off pattern can simulate voltages in between full on (5 Volts) and off (0 Volts) by changing the portion of the time the signal spends on versus the time that the signal spends off.
- The duration of "on time" is called the pulse width.
- If you repeat this on-off pattern fast enough with an LED for example, the result is as if the signal is a steady voltage between 0 and 5v controlling the brightness of the LED.

Pulse Width Modulation

- Arduino's PWM frequency at about 500Hz (Period: 2 ms)



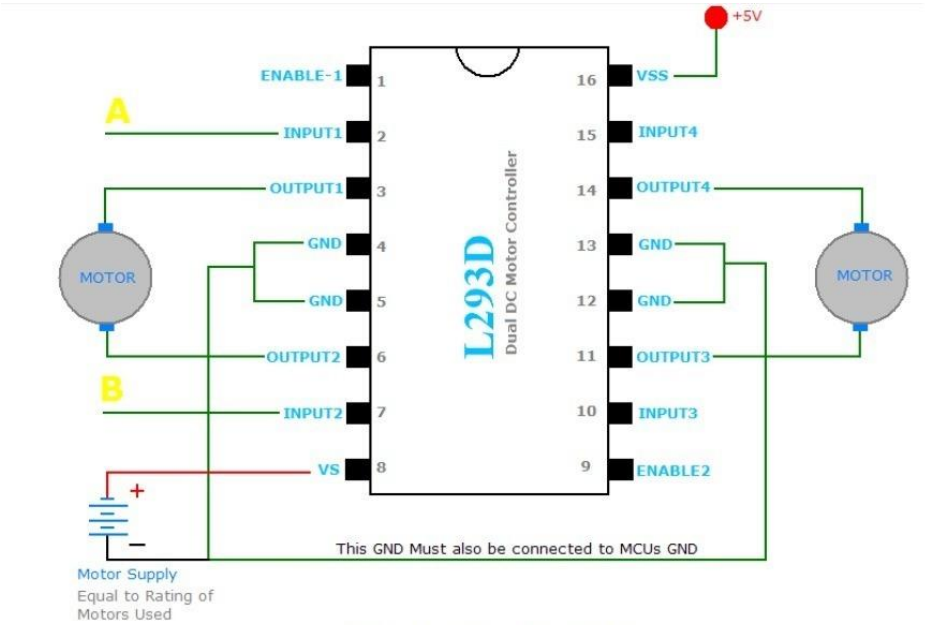


Pulse Width Modulation

- Ex: To create a 3V signal given a digital source that can be either high (on) at 5V or low (off) at 0V, you can use PWM with a duty cycle of 60% which outputs 5V 60% of the time.
- Selecting a duty cycle of 80% would yield 4V, 20% would yield 1V, and so on.

DC-Motor Direction Control

- To control the DC-Motor direction without reversing the battery polarity, we need to use the L293/L293D chip that enables you to control TWO DC-Motors, and their speed as well.

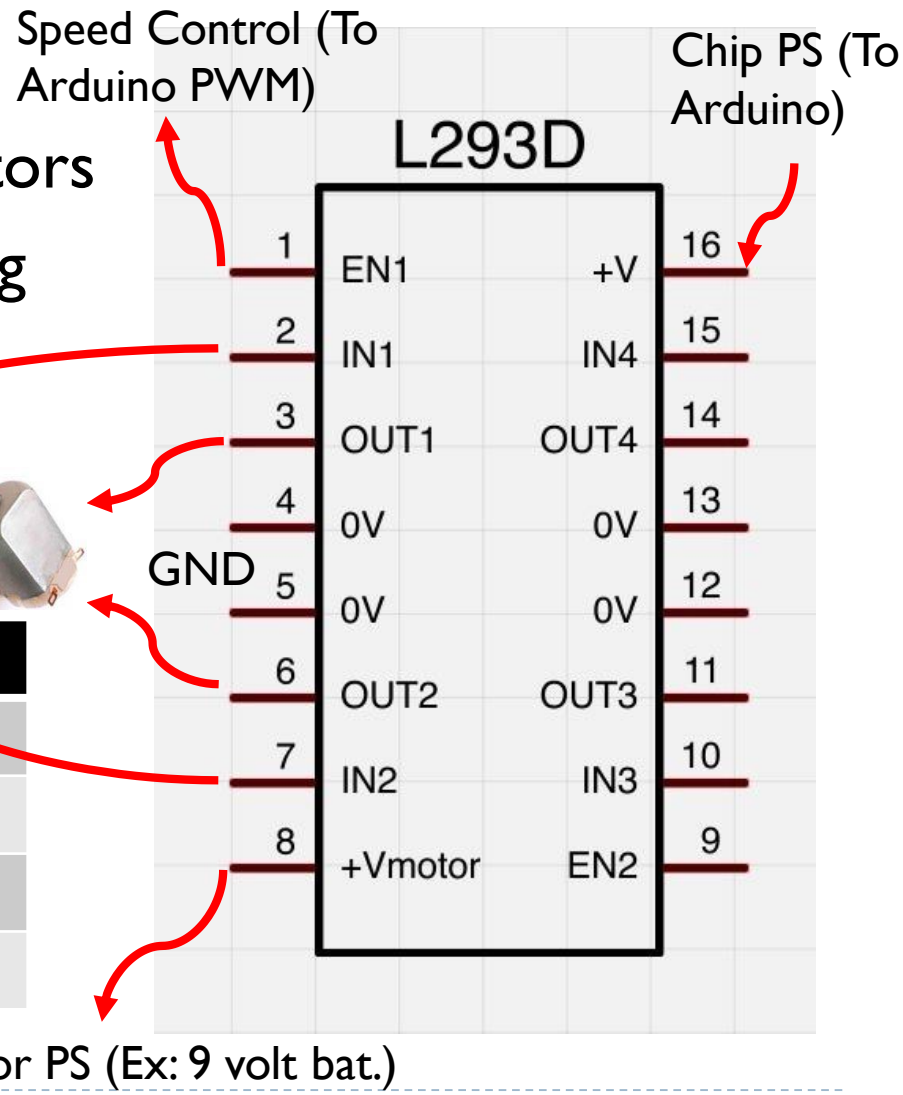


Motor Controller Using L293D

L293D

This is a very useful chip.

It can actually control two motors independently. We are just using half the chip in this lesson.

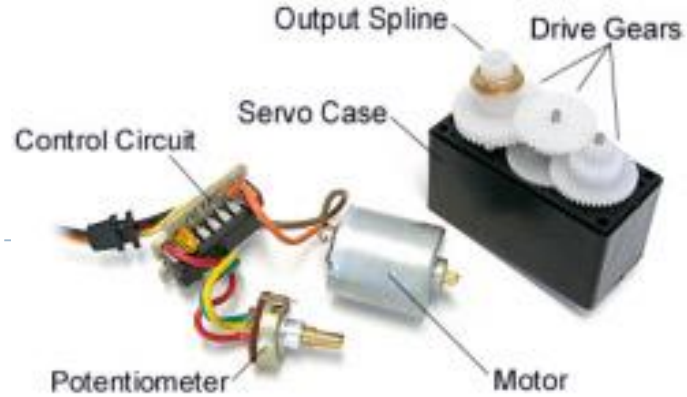


IN1	IN2	Motor Status
GND	GND	Stopped
5V	GND	Turns in Direction A
GND	5V	Turns in Direction B
5V	5V	Stopped

Exercise

1. Connect the 9 volt battery to your motor, the motor should spin one way.
2. Reverse the battery polarity, the motor should spin in the other direction.
3. Connect your motor to the L293 chip and to your Arduino.
4. Write a code to run the motor through the L293 in one direction. Change the code to reverse the motor rotation direction.
5. Add a potentiometer to allow for speed control.

Servo Motors



- **Servos** are DC motors with built in **gearing** and **feedback control loop circuitry**, and no motor drivers required!
- Unlike DC motors, with servo motors you can position the motor shaft at a specific position (angle) using control signal.
- The motor shaft will hold at this position as long as the control signal not changed.
- This is very useful for controlling robot arms, unmanned airplanes control surface or any object that you want it to move at certain angle and stay at its new position.
- Most servo motors can rotate about 90 to 180 degrees. Some rotate through a full 360 degrees or more.

Types of servo motors

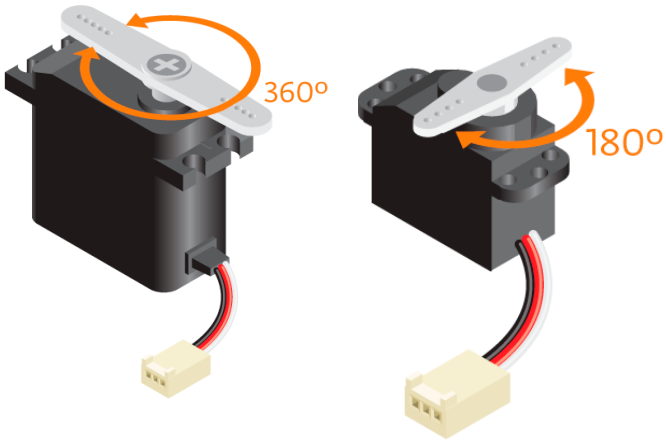
- **Positional rotation servo**:

This is the most common type of servo motor. The output shaft rotates in about half of a circle, or 180 degrees. It has physical stops placed in the gear mechanism to prevent turning beyond these limits to protect the rotational sensor. These common servos are found in radio-controlled cars and water- and aircraft, toys, robots, and many other applications.

- **Continuous rotation servo**:

This is quite similar to the common positional rotation servo motor, except it can turn in either direction indefinitely. The control signal, controls the motor speed rather than direction. You might use a servo of this type as a drive motor on a mobile robot.

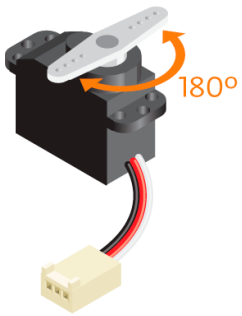
Servo Motors

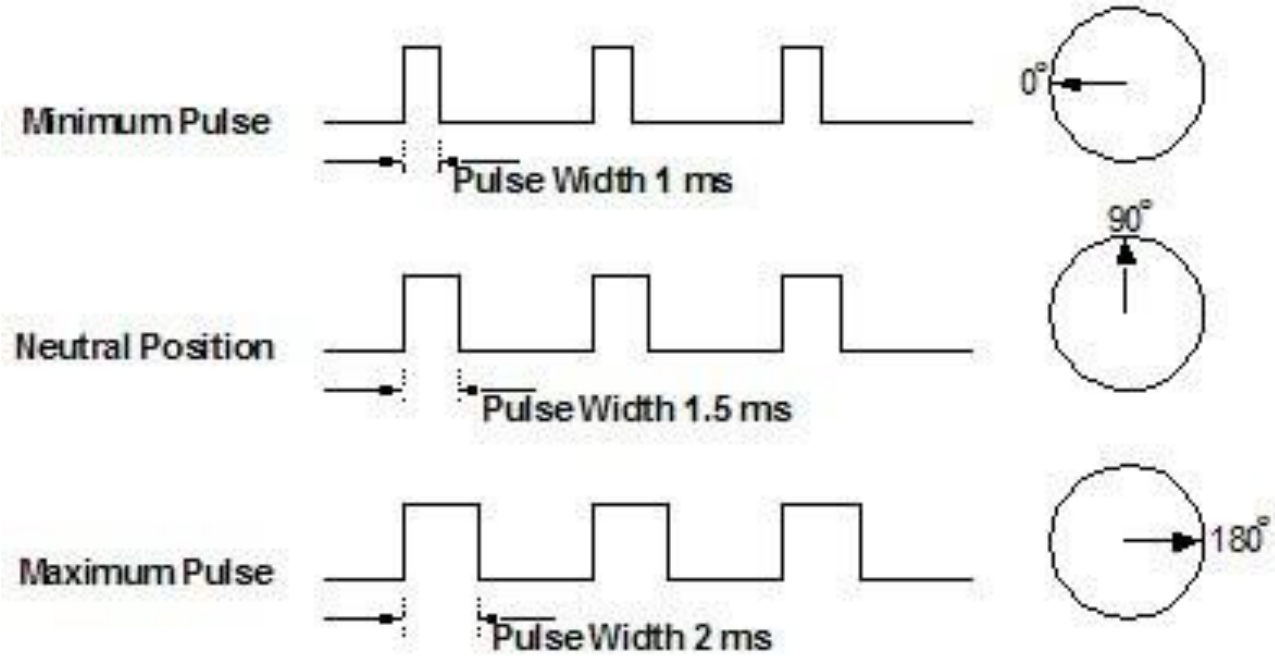


- All servos have three wires:
  Black or **Brown** is for ground.
  Red is for power (~4.8-6V).
  Yellow, **Orange**, or **White** is the **signal** wire (3-5V).
- While the black and red wires provide power to the motor, the signal wire is what you use to command the servo.
- The general concept is to simply send an ordinary logic **square wave** to your servo at a specific wave length, **and your servo goes to a particular angle (or velocity if your servo is modified).** *The wavelength directly maps to servo angle.*

Controlling 180º Servo Motor

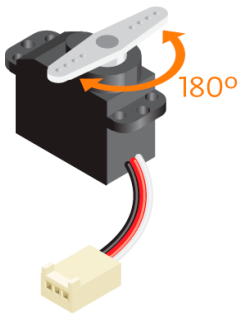
- Bring high a digital port
  Wait between 1-2ms
  Bring low the same digital port
  Cycle a few dozen times per second





Controlling 180º Servo Motor

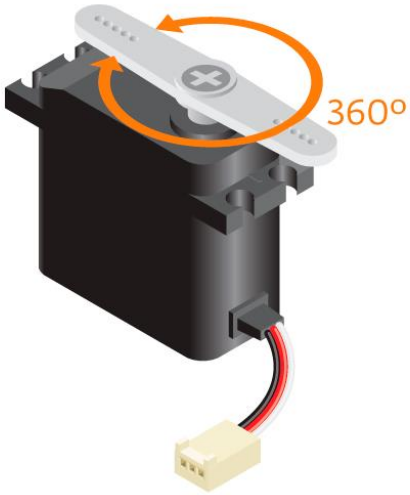
- Pretty simple Code using the library servo:




```
#include <Servo.h>      // Include the servo library
Servo myservo;          // Create servo object to control a servo
void setup() {
myservo.attach(9);      // Attaches the servo on pin 9 to the servo object
}
void loop() {
myservo.write(0);       // Tells the servo what rotation angle to turn to (0)
delay(1000);
myservo.write(180);  // Tells the servo what rotation angle to turn to (180)
delay(1000);
}
```

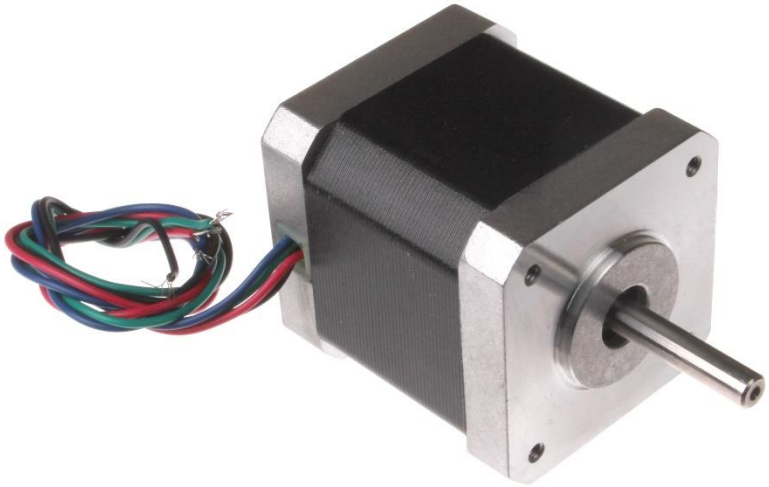
Controlling Continuous Servo Motor

- If your Servo is modified to be continuously rotate, you will loose position control, however, you will be able to control the speed.
- Same code of position control is used to control the speed where the value can still be from 0 to 180 where:
  - 0: is full speed in one direction and
  - 180: is full speed in the opposite direction.



Stepper Motors

- Stepper motors can be found in electronics where high precision is important such as scanners and printers.
- A stepper motor, in contrast to the DC motor, can be very precise with both position and speed.



Stepper Motors

- Stepper motors fall somewhere in between a regular DC motor and a servo motor. They have the advantage that they can be positioned accurately, moved forward or backwards one 'step' at a time, but they can also rotate continuously.
- However, stepper motors will never be very fast compared to a DC motor. It generally has 4 or more wires and usually needs more than 5 volts to work. This means it cannot be powered by an Arduino, but we can use an external power supply.

Stepper Motor Control

- Stepper Motor cannot be controlled directly from Arduino, rather a driver circuit/IC is needed.
- Several Stepper Motor drivers are available such as **L293D**, and the **Easy Driver Circuit**.
- **Tutorial on how to control a 5V Stepper motor using the L293D**: https://learn.adafruit.com/adafruit-arduino-lesson-16-stepper-motors?view=all
- **Tutorial on how to control a 5V Stepper motor using the Easy Driver Circuit**: https://www.sparkfun.com/tutorials/400

References

- www.societyofrobots.com
- http://www.fut-electronics.com/wp-content/plugins/fe_downloads/Uploads/Introduction%20to%20Servo%20Motors%20&%20Arduino.pdf
- http://www.sciencebuddies.org/science-fair-projects/project_ideas/Robotics_ServoMotors.shtml?from=Blog
- http://madrid.verkstad.cc/en/category/block4/block4-concepts/