NYU | TANDON SCHOOL OF ENGINEERING

NSF

Promoting robotic design and entrepreneurship experiences among students and teachers

Lesson 6:
Introduction to Motors

Innovative Technology Experiences for Students and Teachers (ITEST), Professional Development Program, July 2017-19
Mechatronics, Controls, and Robotics Laboratory, Department of Mechanical and Aerospace Engineering, NYU Tandon School of Engineering

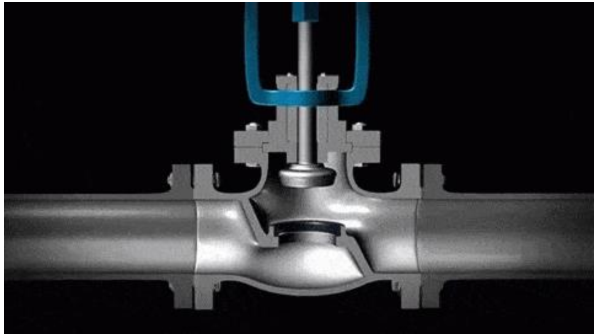
CONTENTS

- DC motors (brushed and brushless)
- Pulse width modulation (PWM)
- Servo motors
- Motor control commands

- **TASK/ACTIVITY**: Motor control

ACTUATOR

- An actuator is a component that is responsible for moving and controlling a mechanism or system
- Examples of systems that use actuators:



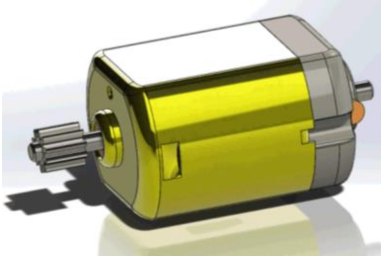
Valve actuators

Electric motors

Pneumatic actuator

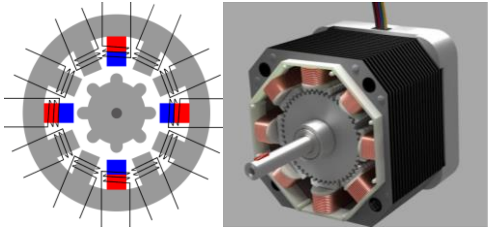
TYPES OF ACTUATORS

- DC motors
- Servo motors
- Stepper motor
- Hydraulics
- Pneumatic actuator

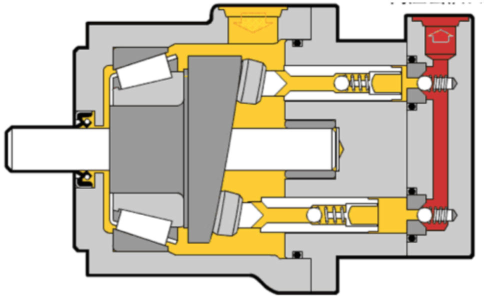


DC motor

Servo motor



Stepper motor



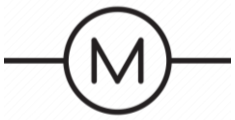
Hydraulic motor

Pneumatic cylinder

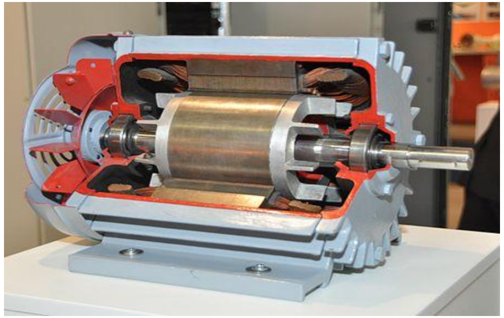
ELECTRIC MOTOR

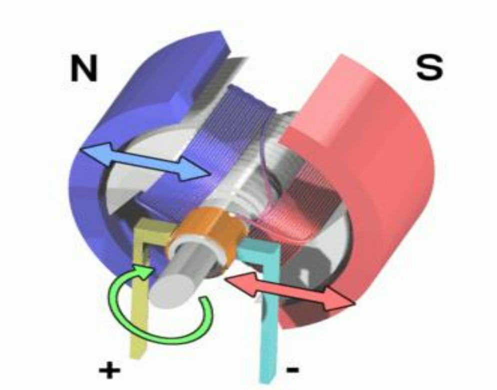
- An electric **motor** is an electrical machine that converts electrical energy into mechanical energy

Circuit symbol:

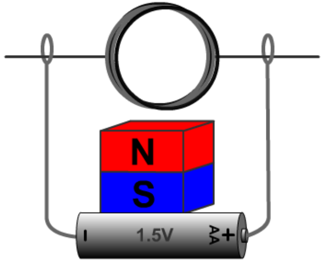


- An electric **generator** operates in the reverse direction, converting mechanical energy into electrical energy

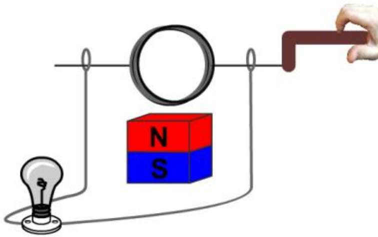




Electric Motor



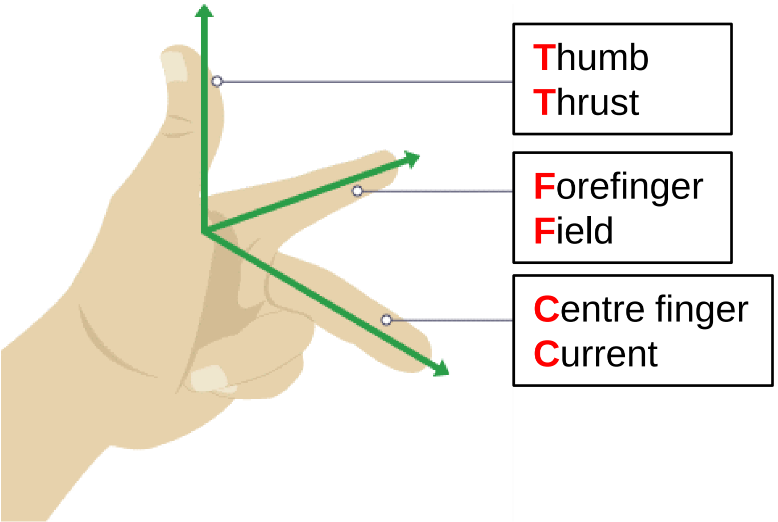
Motor



Generator

ELECTRIC MOTOR - WORKING PRINCIPLE

- Interaction between a **magnetic field** and a **current carrying conductor** produces a **force** (called “**Lorentz force**”)

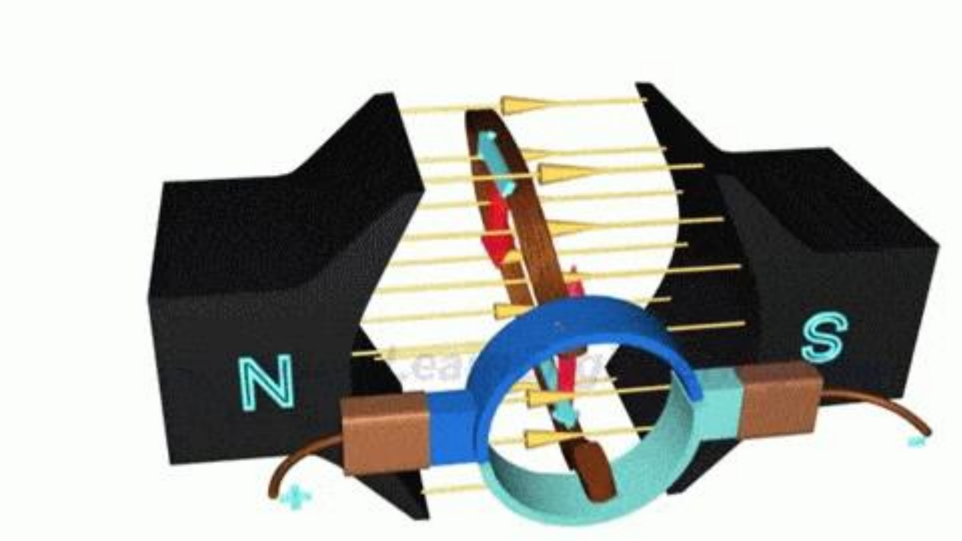


Thumb
Thrust

Forefinger
Field

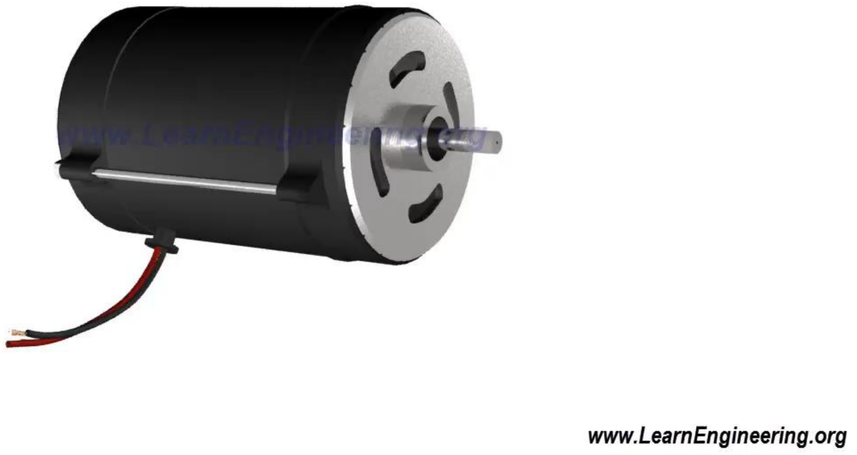
Centre finger
Current

Source



Source

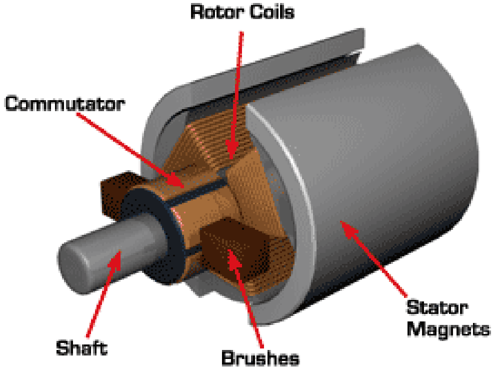
WORKING OF A DC MOTOR



Video

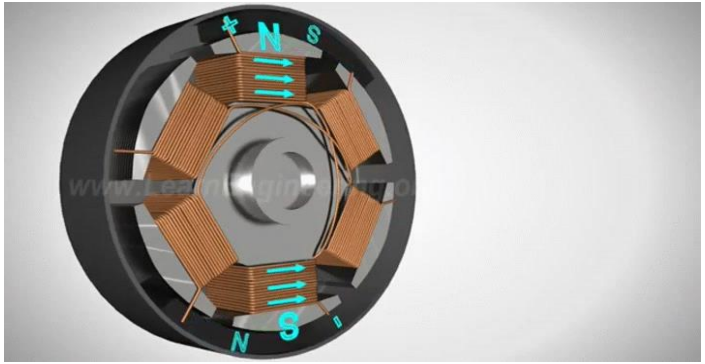
BRUSHED V/S BRUSHLESS DC MOTORS

Brushed DC motor



- Permanent magnets for outer stator
- Rotating coils for inner rotor
- Commutator with **metal contact brushes** to reverse the polarity of the rotor
- May cause **sparking** due to wear of brushes

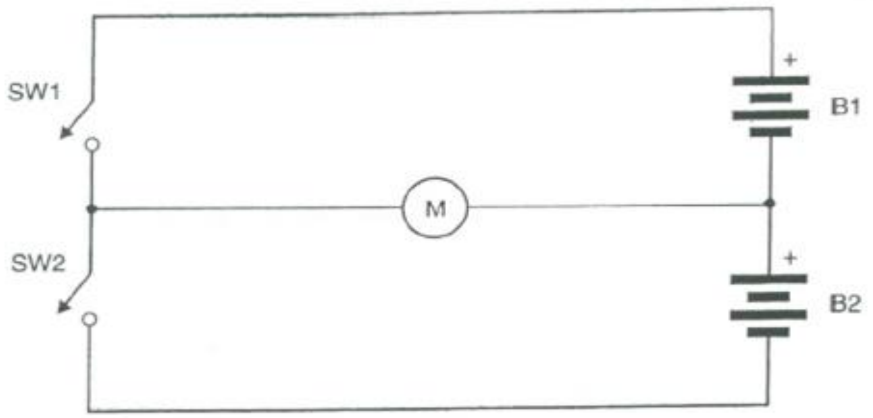
Brushless DC Motor, How it works ? - Lesics



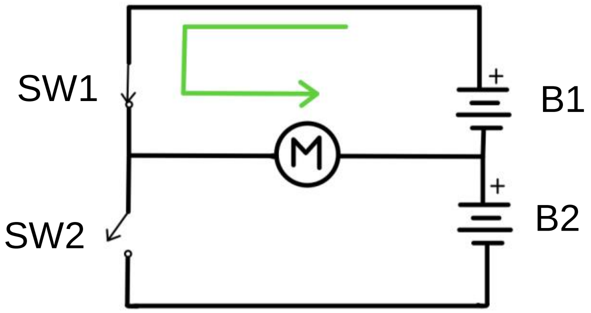
- Permanent magnets for outer rotor
- Rotating coils for inner stator
- **No brushes**
- **No sparking**, less noisy, longer life

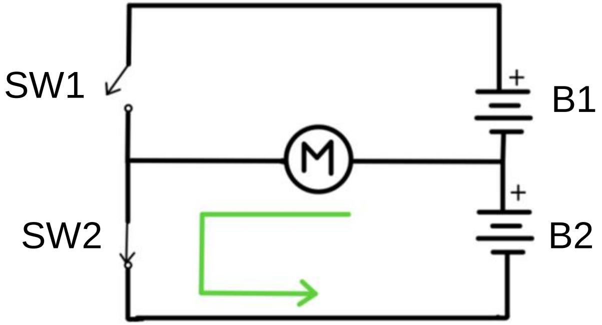
DC MOTOR DIRECTION CONTROL (Half-bridge)

- An **H bridge** is an electronic circuit that enables a voltage to be applied across a motor in the opposite directions
- To explain the H bridge, we begin with the consideration of a **half bridge**
- Consider the circuit below with 2 voltage sources, 2 switches and a DC motor:



HALF-BRIDGE CIRCUIT (using switches)



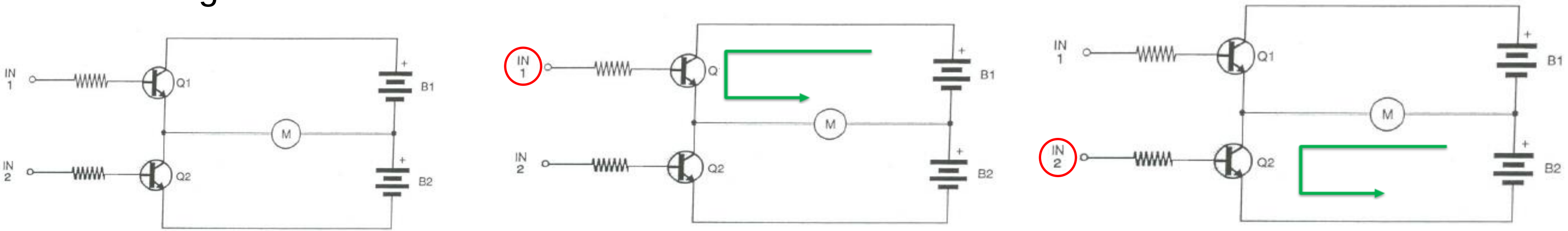


- When **SW1 is closed**, B1 is connected to the motor, current flows from left to right, motor turns in **one direction**
- When **SW2 is closed**, B2 is connected to the motor, current flows from right to left, motor turns in the **opposite direction**
- **SW1 and SW2 cannot be closed simultaneously**, as this leads to B1 and B2 being in **short-circuit**

What electronic component can be used instead of a mechanical switch?

HALF-BRIDGE CIRCUIT (using transistors)

- Consider the circuit below with switches replaced by two NPN **transistors** for electrical switching:

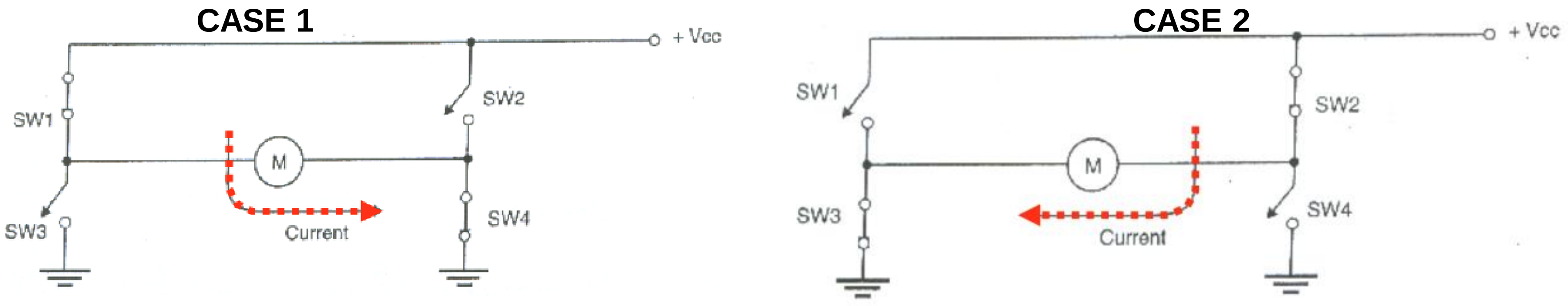


- ➢ CASE 1: **IN1 is high**, Q1 conducts → motor turns in **forward direction** by B1
- ➢ CASE 2: **IN2 is high**, Q2 conducts → motor turns in **reverse direction** by B2
- **IN1 and IN2 cannot be driven high simultaneously**, as this leads to B1 and B2 being in **short-circuit**

The main disadvantage of a half bridge DC motor drive circuit is that it requires a **dual power supply**

NOTE: These circuit diagrams are for conceptual understanding only, diodes will be required in the half-bridge circuit for control of a DC motor in real life

DC MOTOR DIRECTION CONTROL (Full-bridge)



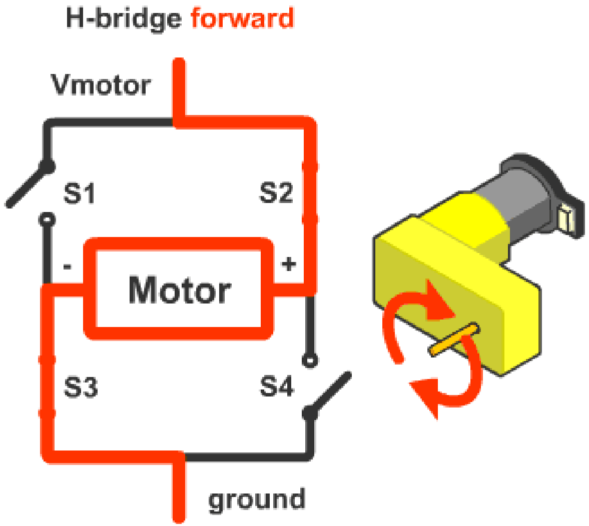
CASE 1

CASE 2

- A full-bridge circuit is called a **H-bridge** (the shape of the circuit resembles the letter “H”)
- Consider the circuit show, which consists of 4 switches, 1 voltage source, and a DC motor:
- CASE 1: **SW1 and SW4** are closed (SW2 and SW3 open) → V_{CC} drives motor in **forward direction**
- CASE 2: **SW2 and SW3** are closed (SW1 and SW4 open) → V_{CC} drives motor in **reverse direction**

NOTE: These circuit diagrams are for conceptual understanding only, diodes will be required in the H-bridge circuit for control of a DC motor in real-life

H-BRIDGE CIRCUIT (using switches)

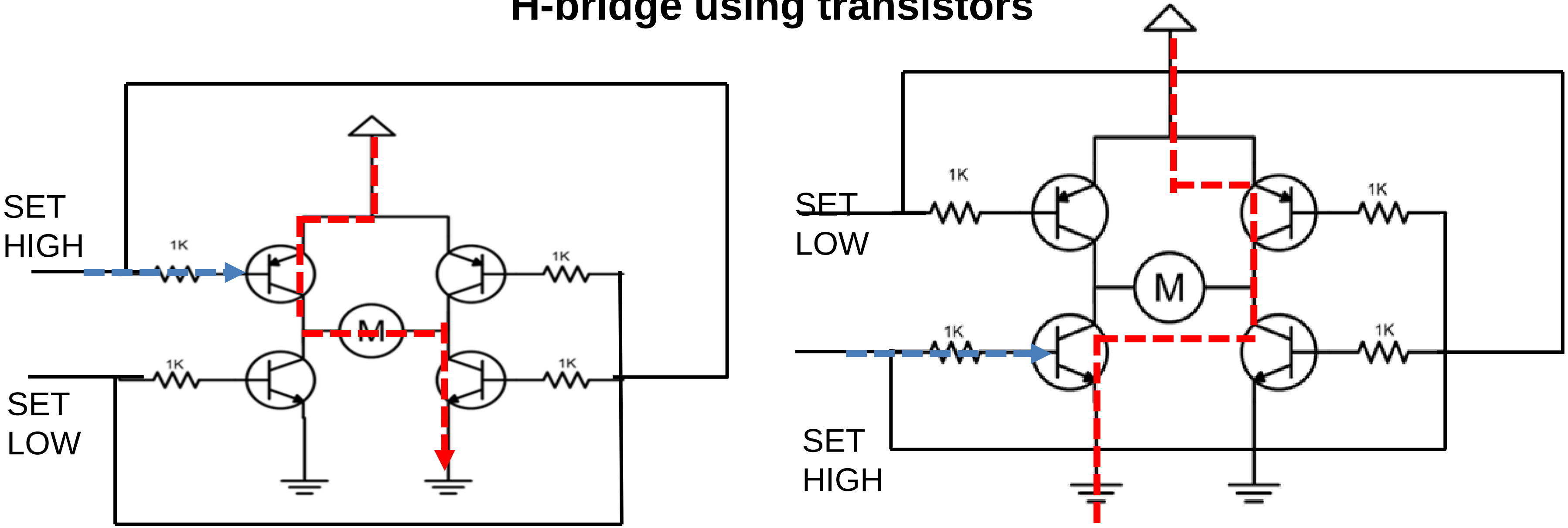


H-Bridge

NOTE: These circuit diagrams are for conceptual understanding only, diodes will be required in the H-bridge circuit for control of a DC motor in real life

H-BRIDGE CIRCUIT (using transistors)

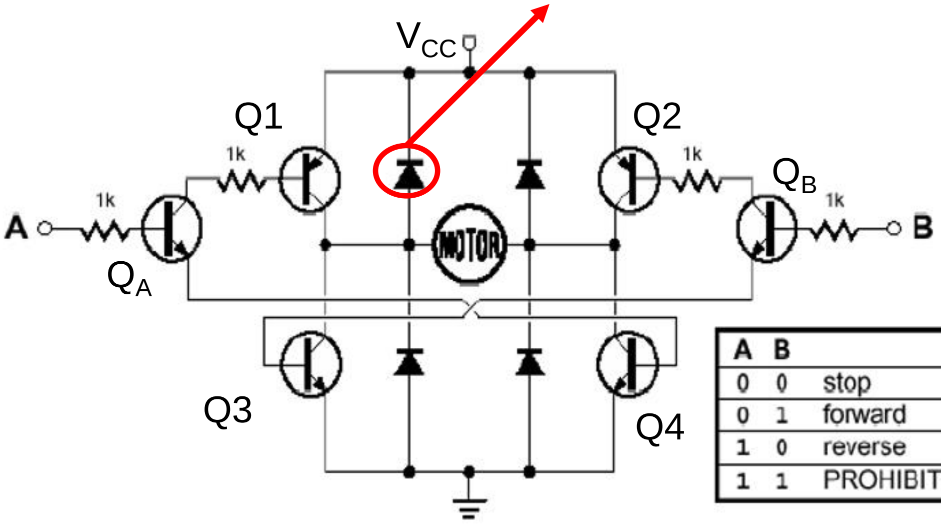
H-bridge using transistors



NOTE: These circuit diagrams are for conceptual understanding only, diodes will be required in the H-bridge circuit for control of a DC motor in real life

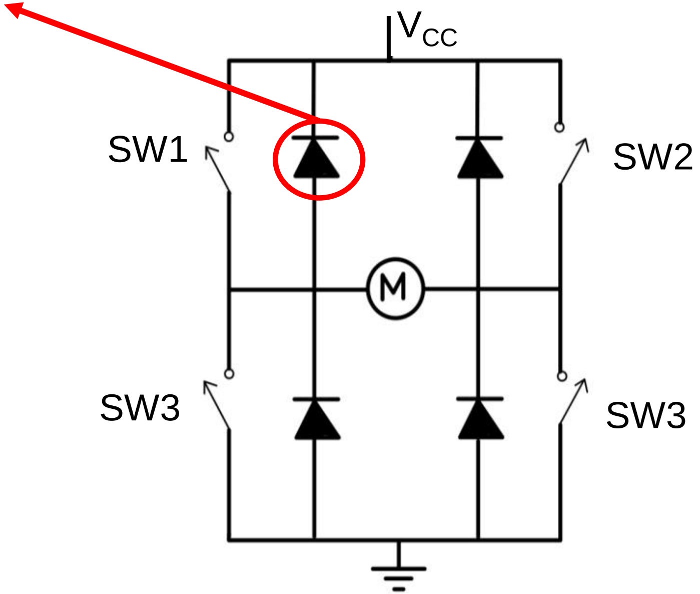
DC MOTOR DIRECTION CONTROL (H-bridge)

Diodes are used to prevent damage to switching elements from **inductive kickback**



A	B	
0	0	stop
0	1	forward
1	0	reverse
1	1	PROHIBITED

(1) Using Transistors



(2) Using switches

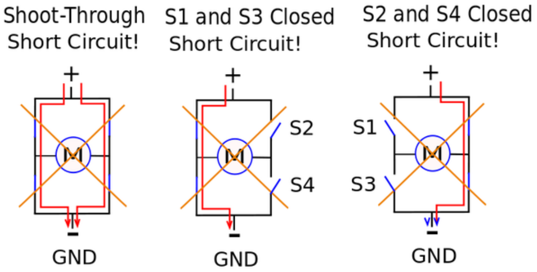
To control a DC motor in real-life, the **full-fledged H-Bridge circuit in figure (1)** can be used

WHY IS DIODE REQUIRED IN A H-BRIDGE?

- In the prohibited case, i.e., turning ON switching elements on same side of the bridge simultaneously, creates a **short circuit**

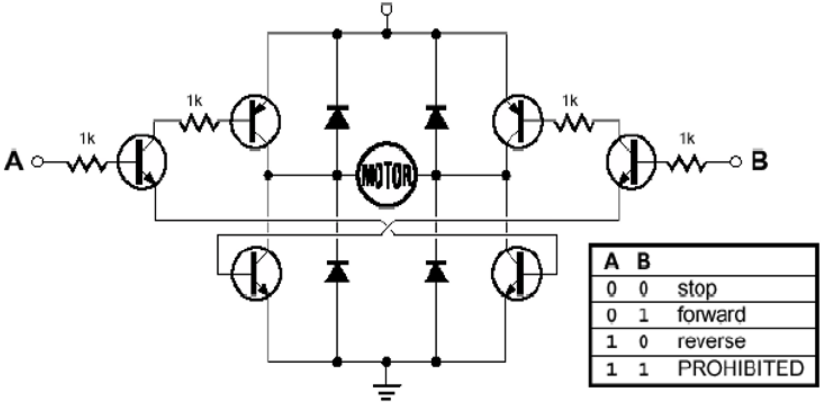
- While in other cases, when current is instantaneously cut off in a H-bridge (when switching from forward to reverse or vice-versa), the **stored magnetic energy results in a high spike in voltage** across the motor (inductor):

V = L $\frac{di}{dt}$, V increases as the $\frac{di}{dt}$ term increases, expressing a high **flyback voltage**, which damages the transistors



Shoot-Through Short Circuit!
S1 and S3 Closed Short Circuit!
S2 and S4 Closed Short Circuit!
+
M
S2
S4
S1
S3
-
GND

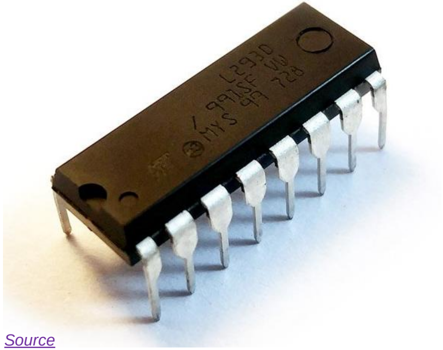
Source



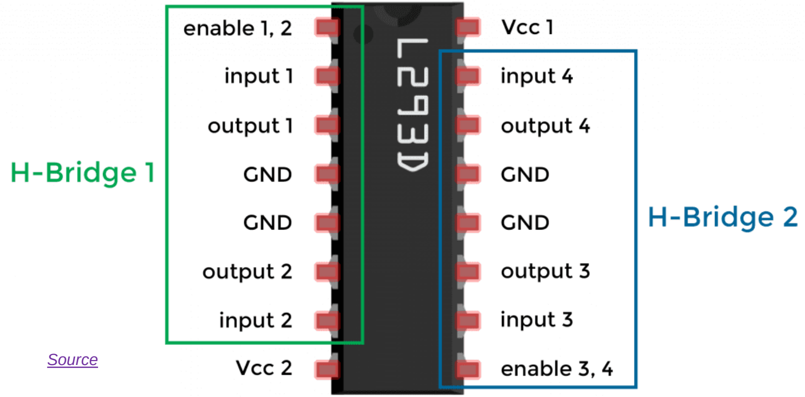
A	B	
0	0	stop
0	1	forward
1	0	reverse
1	1	PROHIBITED

- In either case, the **diodes** provide a path for the current during the **switching periods**, dissipating the energy as heat, to protect the switching elements.

DC MOTOR DIRECTION CONTROL (L293D IC)



Source

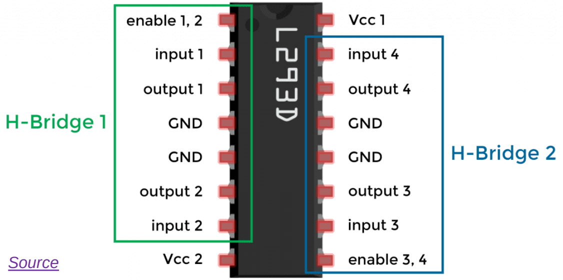


Source

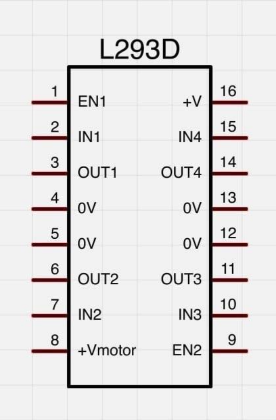
- Motor driver ICs are **integrated circuit chips** that simplify control of motors
- The L293D is a 16-pin Motor Driver IC which can control two DC motors simultaneously and independently
- It can provide **600mA** per channel at a supply voltage range of **4.5V to 36V**
- It has an **internally embedded diode** so there is no need of external diodes for interfacing the DC motor

ACTIVITY -1 (DC MOTOR: DIRECTION)

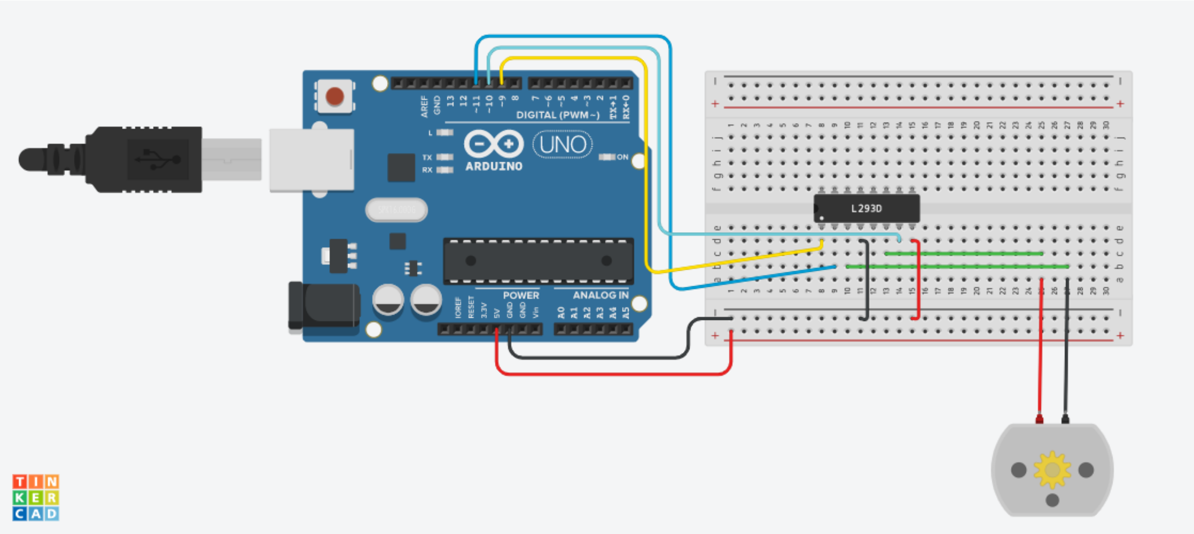
Wiring Arduino with L293D and DC Motor



Source

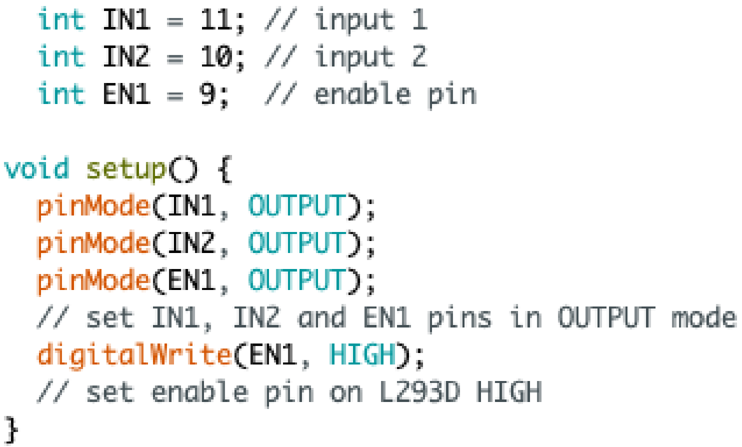


Source

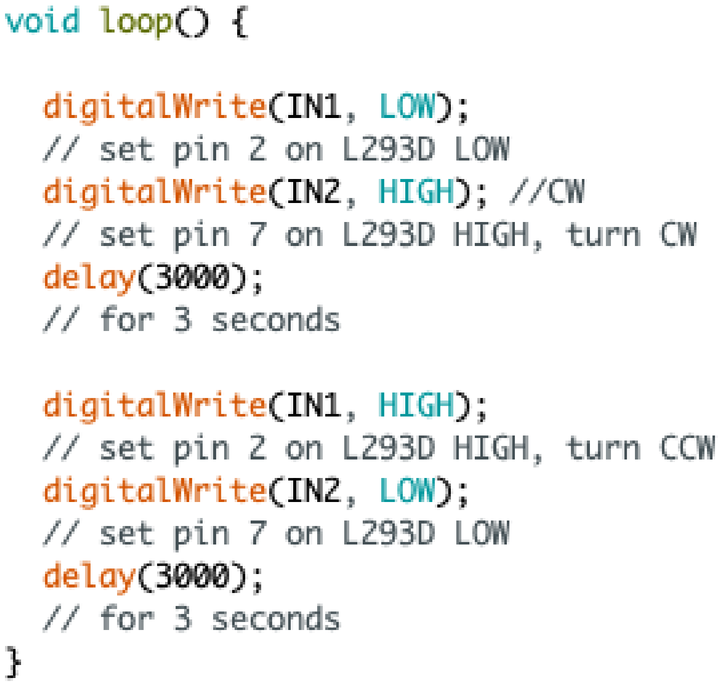


ACTIVITY -1 (DC MOTOR: DIRECTION)

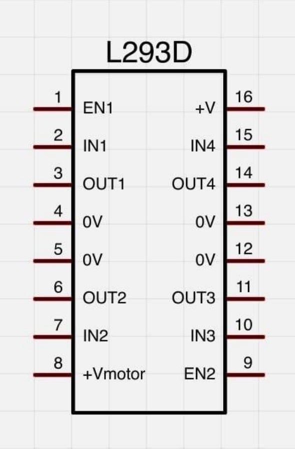
CODE



```
  int IN1 = 11; // input 1
  int IN2 = 10; // input 2
  int EN1 = 9;  // enable pin

void setup() {
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(EN1, OUTPUT);
  // set IN1, IN2 and EN1 pins in OUTPUT mode
  digitalWrite(EN1, HIGH);
  // set enable pin on L293D HIGH
}
```



```
void loop() {

  digitalWrite(IN1, LOW);
  // set pin 2 on L293D LOW
  digitalWrite(IN2, HIGH); //CW
  // set pin 7 on L293D HIGH, turn CW
  delay(3000);
  // for 3 seconds

  digitalWrite(IN1, HIGH);
  // set pin 2 on L293D HIGH, turn CCW
  digitalWrite(IN2, LOW);
  // set pin 7 on L293D LOW
  delay(3000);
  // for 3 seconds
}
```

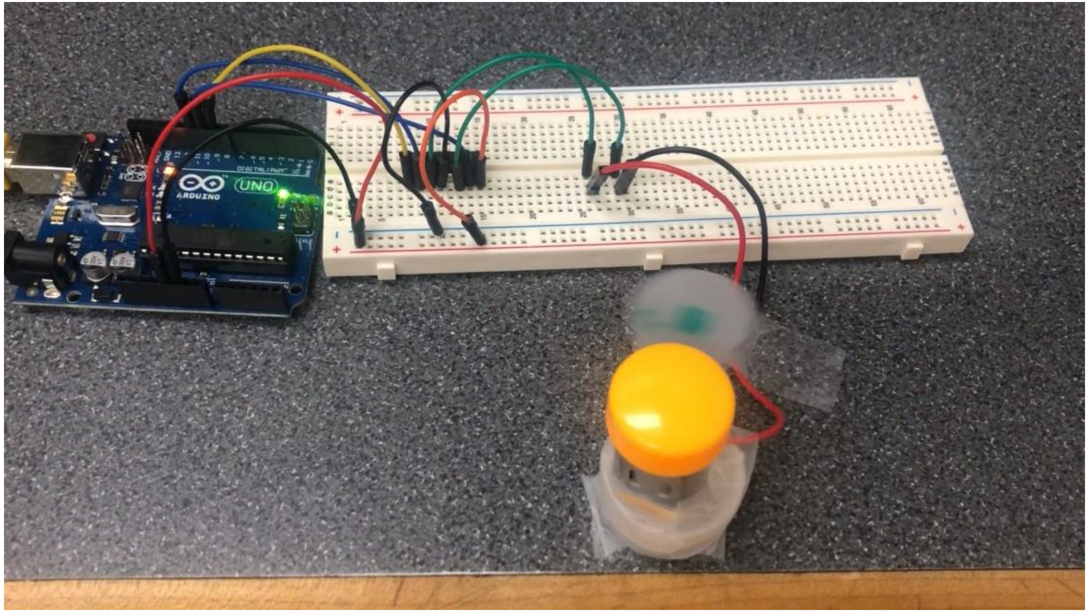


Source

Program

ACTIVITY -1 (DC MOTOR: DIRECTION)

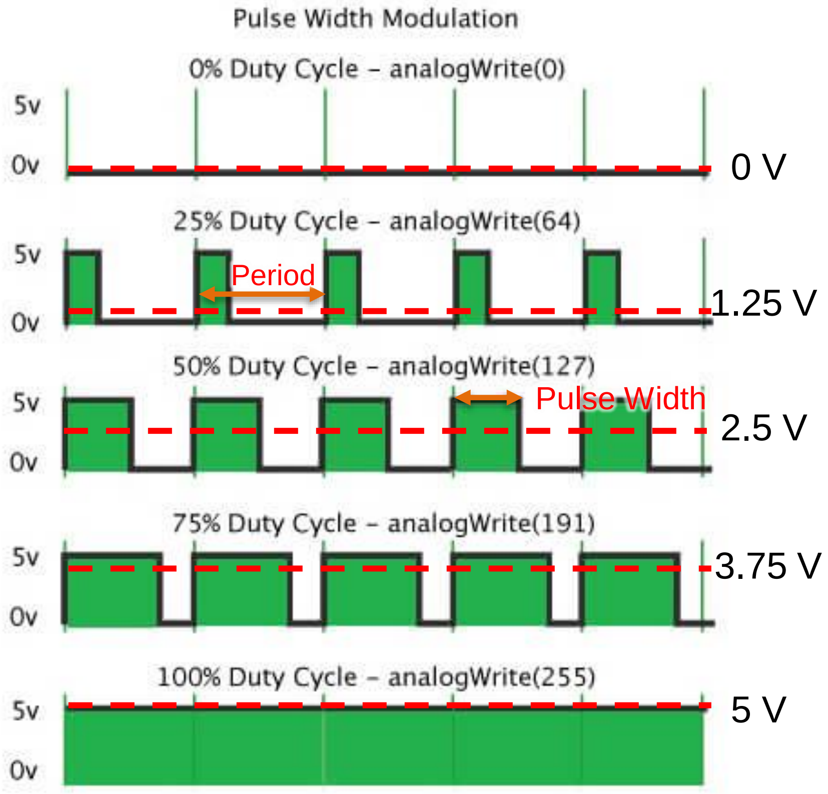
NOTE: Speed of the DC motor was reduced to clearly show the change in direction in the video



Video

PULSE WIDTH MODULATION

Effective voltage – – – –


Pulse Width Modulation
0% Duty Cycle – analogWrite(0)
5v
0v
0 V
25% Duty Cycle – analogWrite(64)
5v
0v
Period
1.25 V
50% Duty Cycle – analogWrite(127)
5v
0v
Pulse Width
2.5 V
75% Duty Cycle – analogWrite(191)
5v
0v
3.75 V
100% Duty Cycle – analogWrite(255)
5v
0v
5 V

Source

DC motor speed can be controlled using PWM

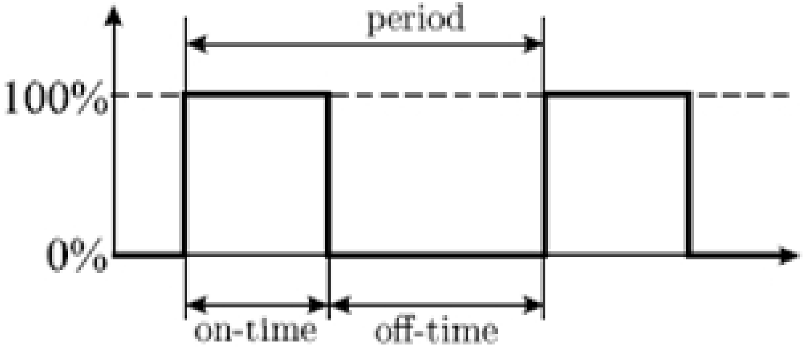
DUTY CYCLE

$$\text{duty cycle} = \frac{t_{ON}}{t_{ON} + t_{OFF}}$$

t_{ON} = ON time

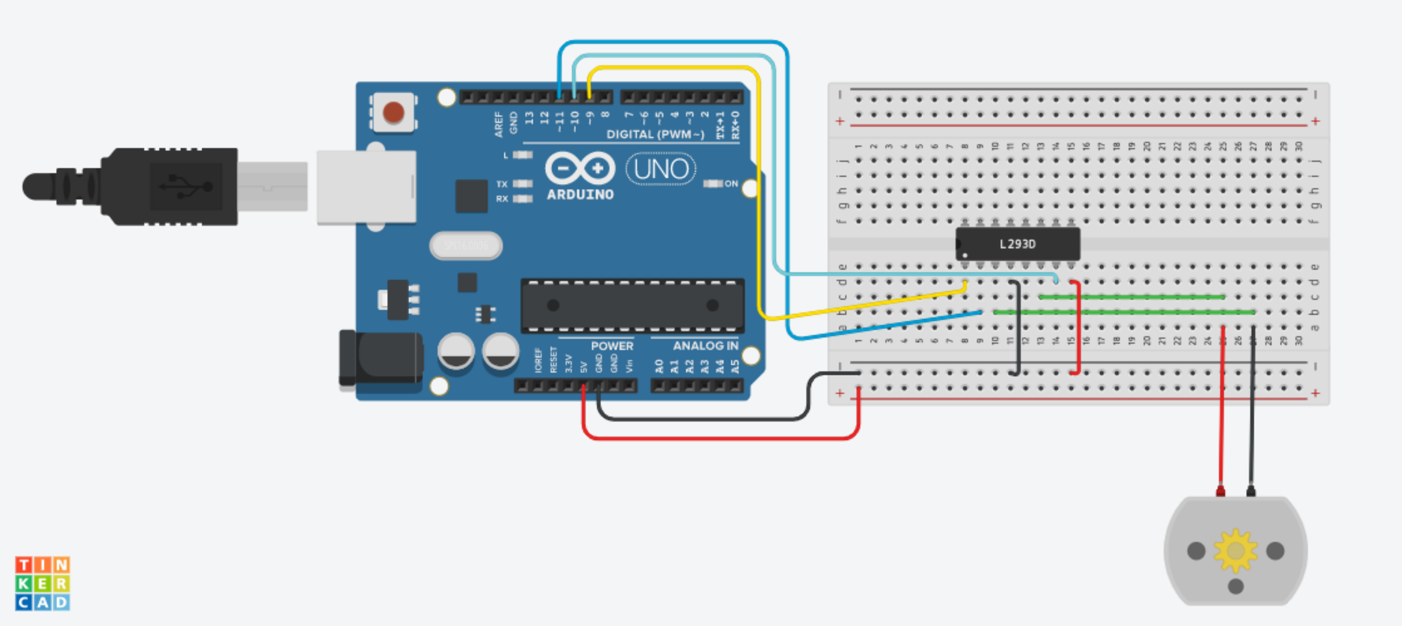
t_{OFF} = OFF time

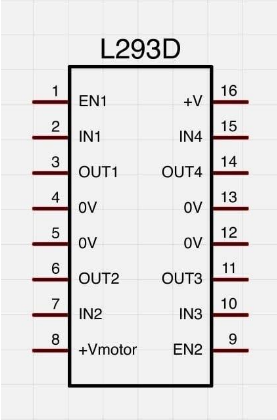
$t_{ON} + t_{OFF}$ = Time period



ACTIVITY - 2 (DC MOTOR: SPEED)

Same circuit as ACTIVITY - 1



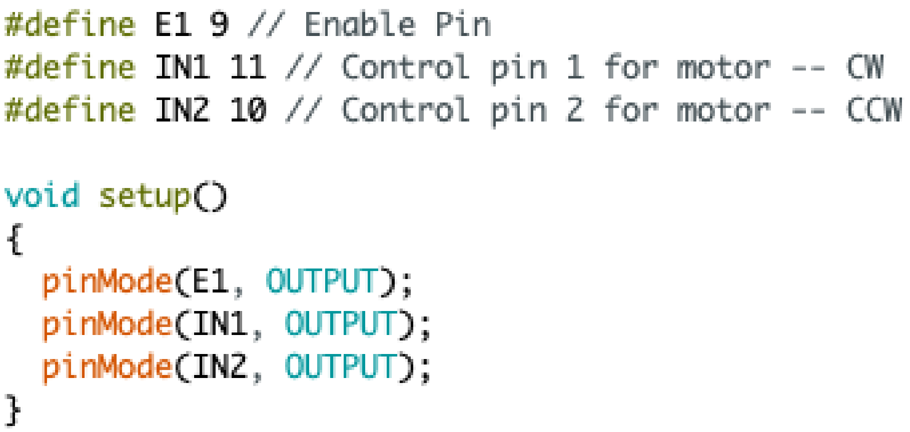


Source

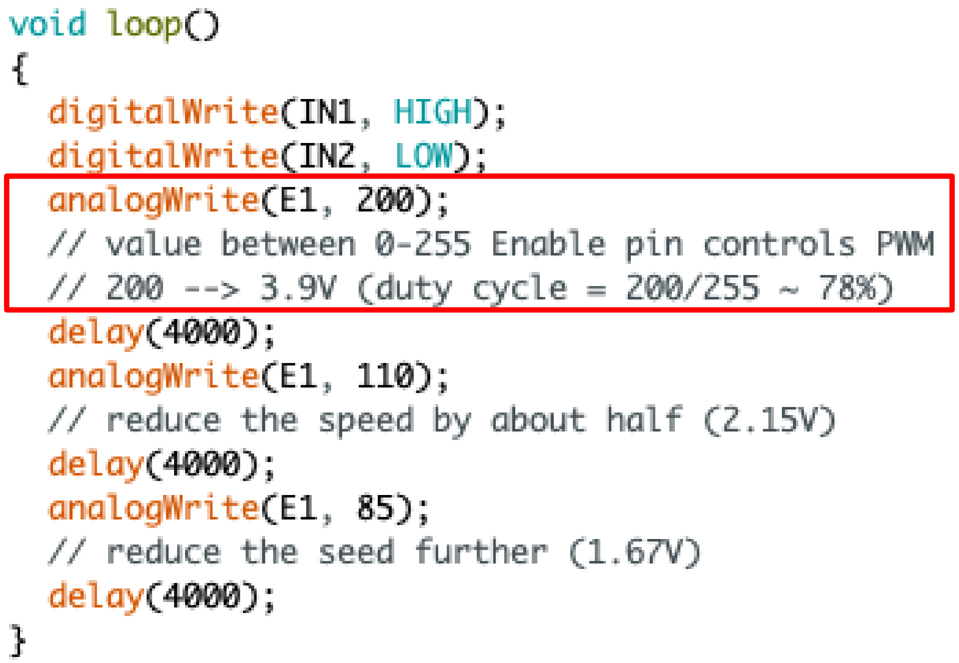
NOTE: Arduino pins with ”~” sign next to the pin number are **PWM pins,** used to control actuators using PWM (here, pins 10 & 11 is used to control a DC motor)

ACTIVITY - 2 (DC MOTOR: SPEED)

DC Motor Speed Control Code



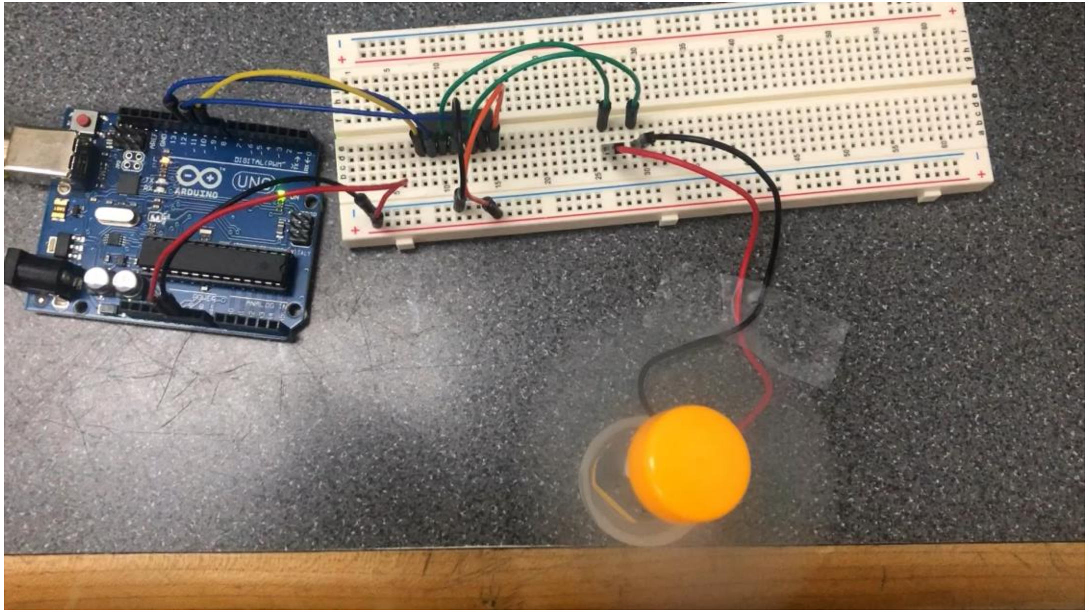
```
#define E1 9 // Enable Pin
#define IN1 11 // Control pin 1 for motor -- CW
#define IN2 10 // Control pin 2 for motor -- CCW

void setup()
{
  pinMode(E1, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
}
```



```
void loop()
{
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  analogWrite(E1, 200);
  // value between 0-255 Enable pin controls PWM
  // 200 --> 3.9V (duty cycle = 200/255 ~ 78%)
  delay(4000);
  analogWrite(E1, 110);
  // reduce the speed by about half (2.15V)
  delay(4000);
  analogWrite(E1, 85);
  // reduce the seed further (1.67V)
  delay(4000);
}
```

Program

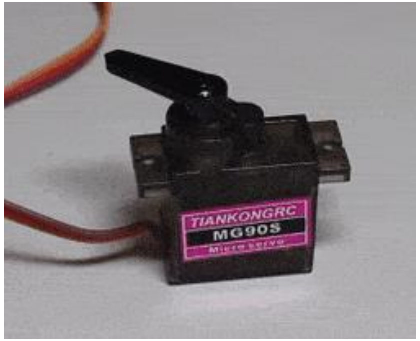
ACTIVITY - 2 (DC MOTOR: SPEED)

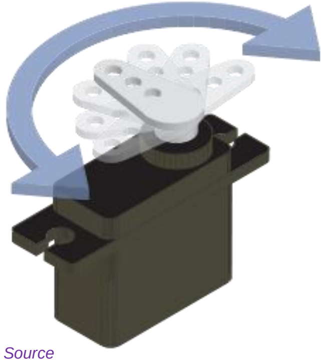


Video

SERVO

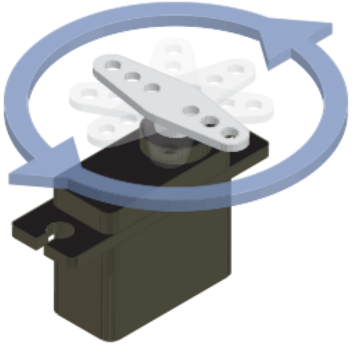
- **Servo motor** is a type of actuator used for angular positioning
- **Standard servo** typically has a movement range of 180 degrees
- **Continuous servo** has a freedom to complete one full rotation





Source

Standard Servo motor



Source

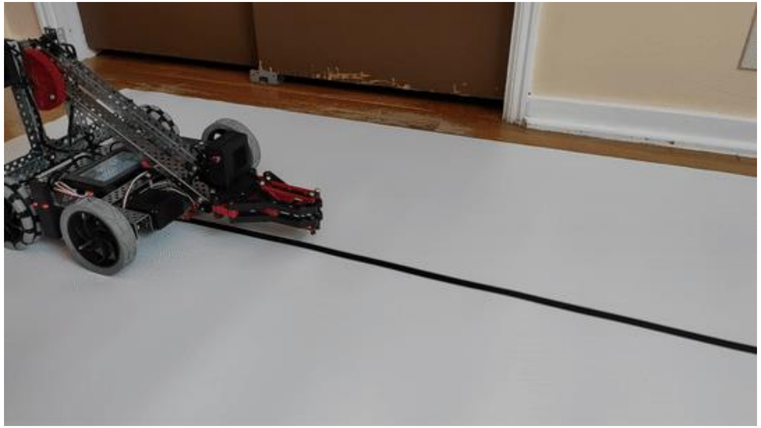
Continuous Servo Motor

STANDARD V/S CONTINUOUS SERVO

Standard servo only turns over a range (usually 0°-180°), with precise feedback control over its angular **position**

Continuous rotation servo turns continuously, with control over its **speed and direction**

Source

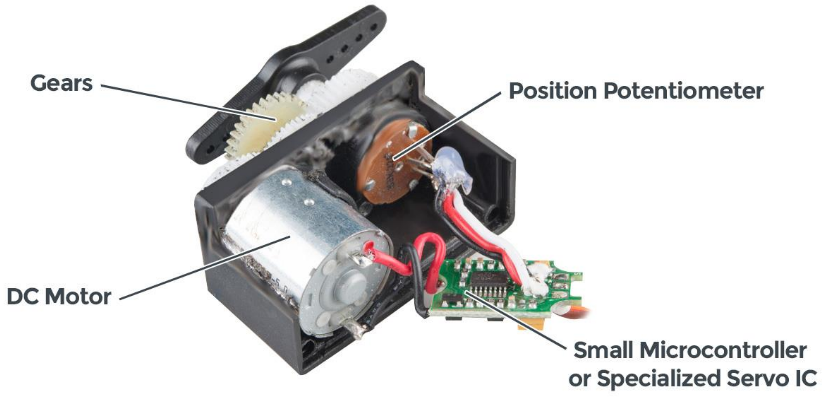


Source

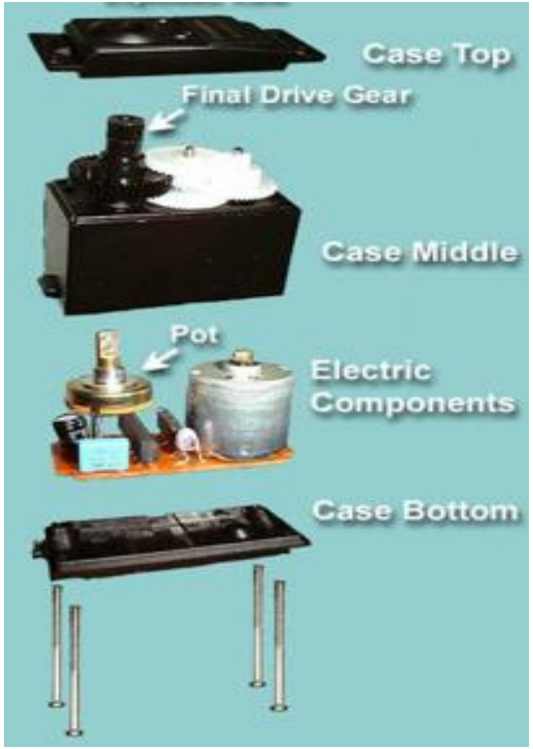
Standard servo example: **Robotic arm**

Continuous servo example: **Mobile Robot**

SERVO: SECTIONAL & EXPLODED VIEW

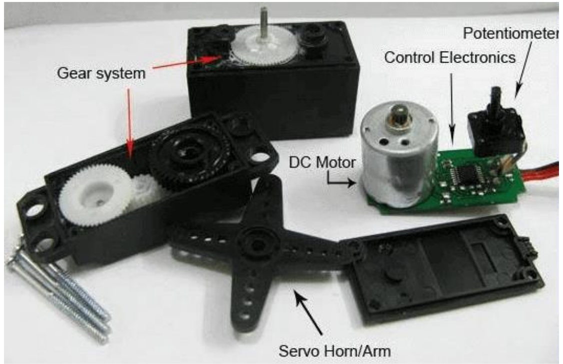


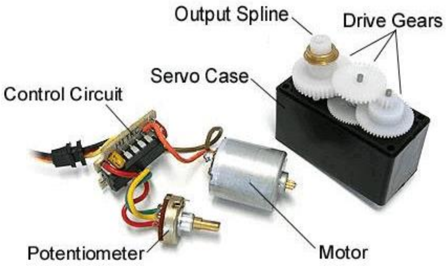
Sectional View



Exploded View

SERVO PARTS





Servo motors are constructed out of basic **DC motors**, by adding

- **Gear** reduction

- **Position sensor** for the motor shaft

- **Electronic circuit** that controls the motor's operation

SERVO CONNECTOR

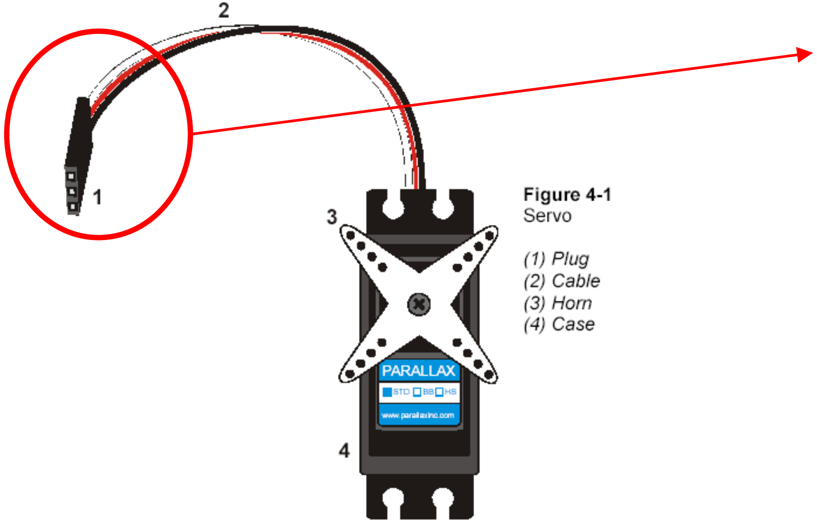
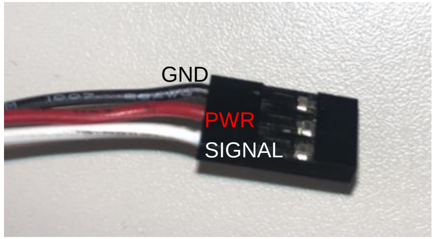
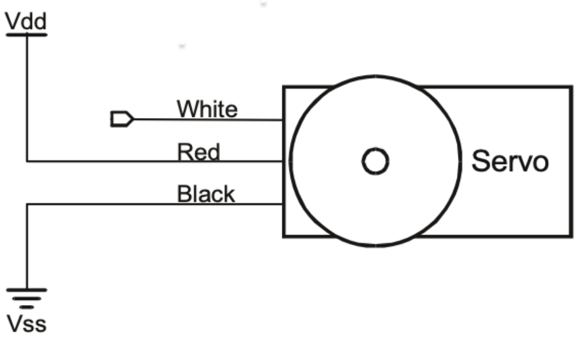


Figure 4-1
Servo

(1) Plug
(2) Cable
(3) Horn
(4) Case

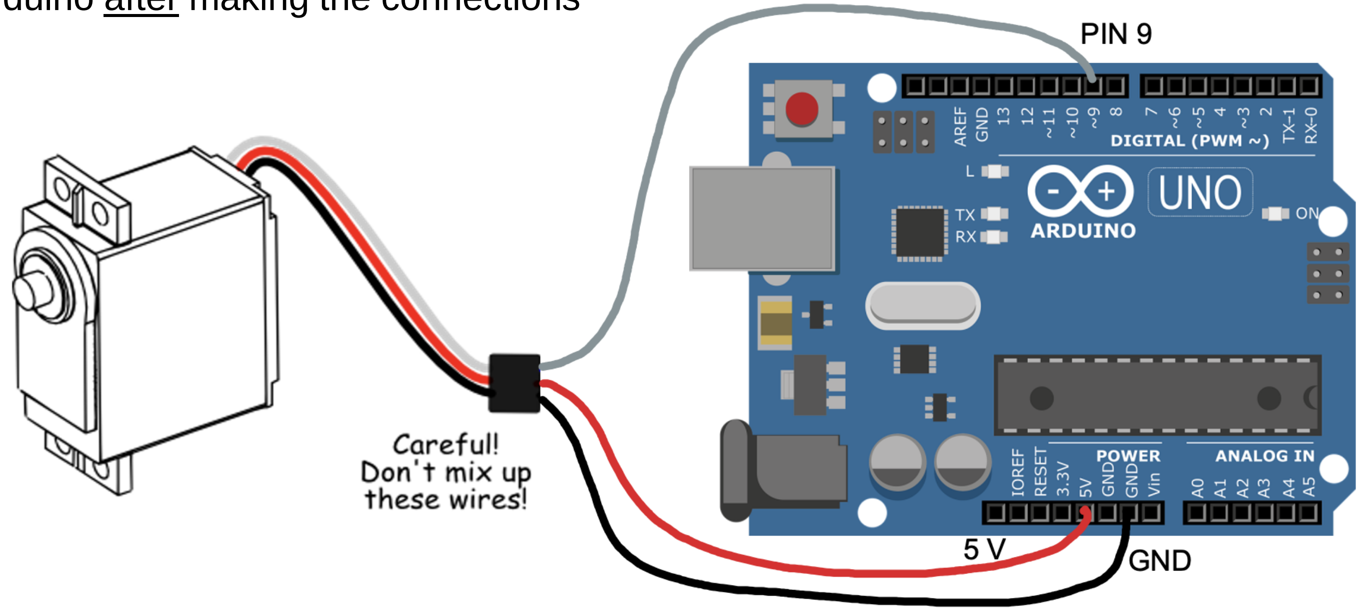


<u>Servo Connector</u>:
Black – ground
Red – power
White – signal



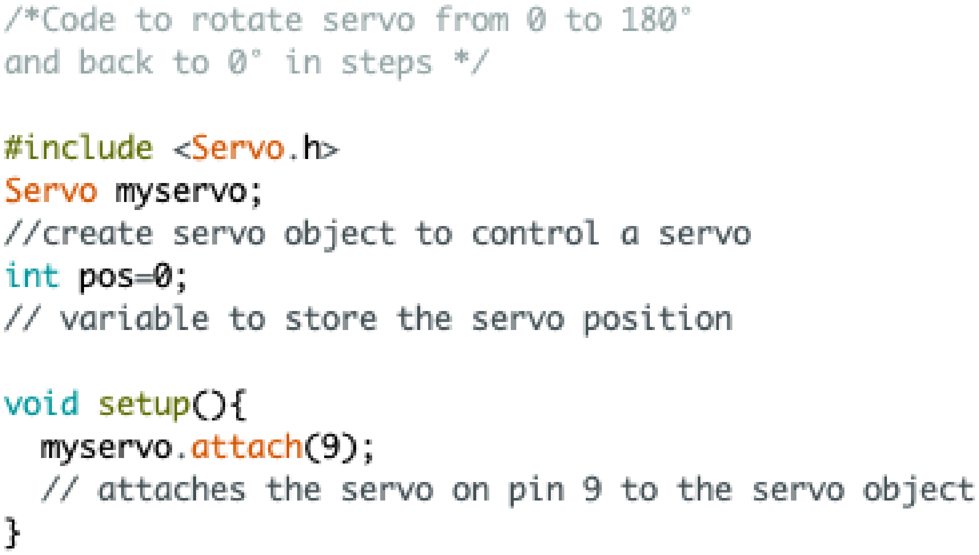
INTERFACING ARDUINO WITH SERVO

NOTE: Always power the Arduino after making the connections

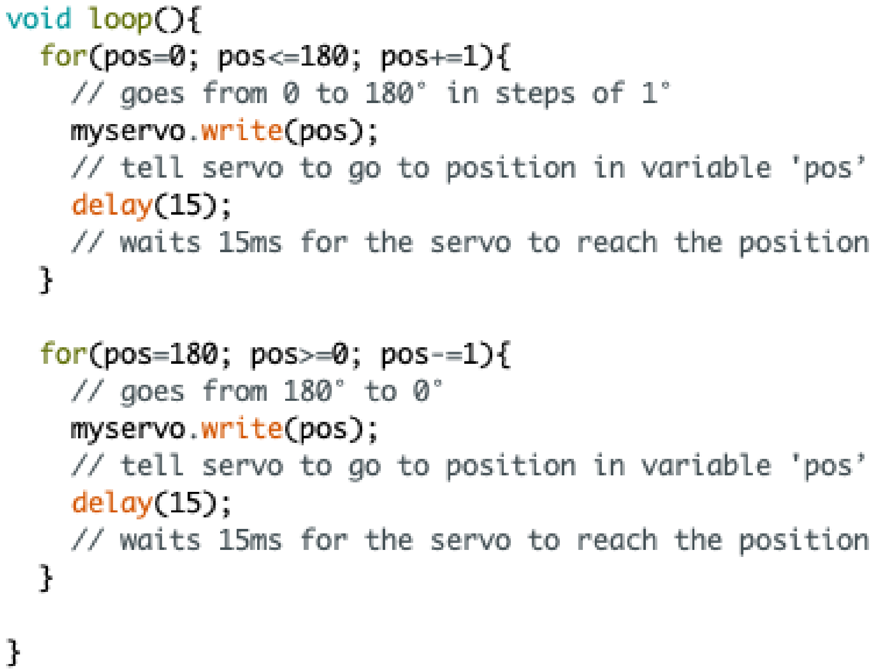


NOTE: Pins with ”~” sign next to the pin number are **PWM pins** used to control actuators using PWM (here, pin 9 is used to control the servo)

ACTIVITY - 3 (SERVO 0° TO 180°)



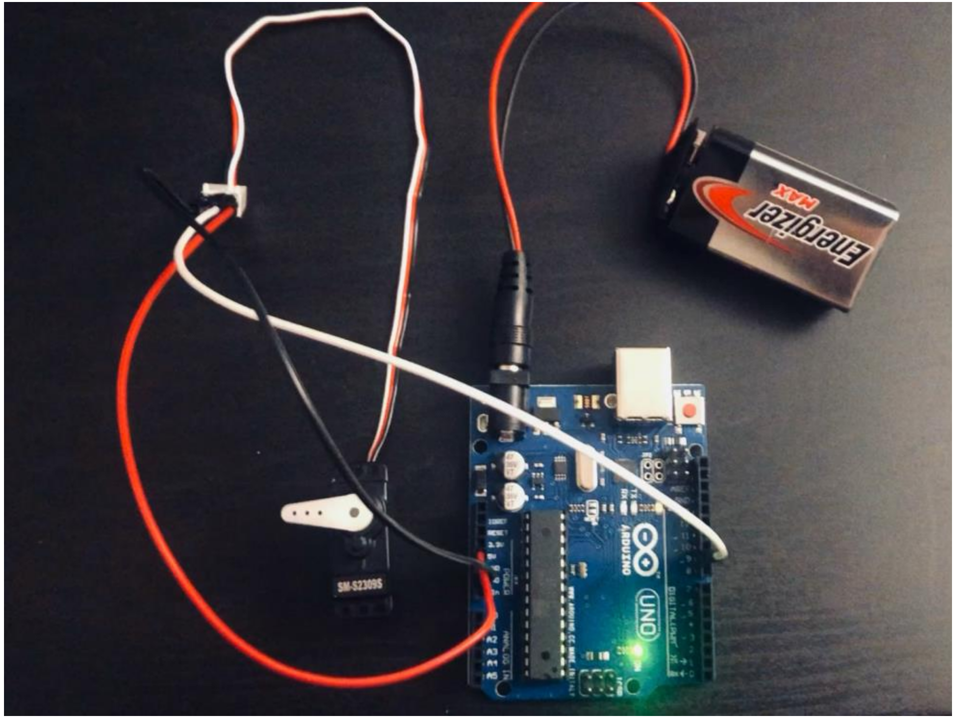
```
/*Code to rotate servo from 0 to 180°
and back to 0° in steps */

#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 to 180° in steps of 1°
    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° to 0°
    myservo.write(pos);
    // tell servo to go to position in variable 'pos'
    delay(15);
    // waits 15ms for the servo to reach the position
  }

}
```

Program

ACTIVITY - 3 (SERVO 0˚ TO 180˚)



Video

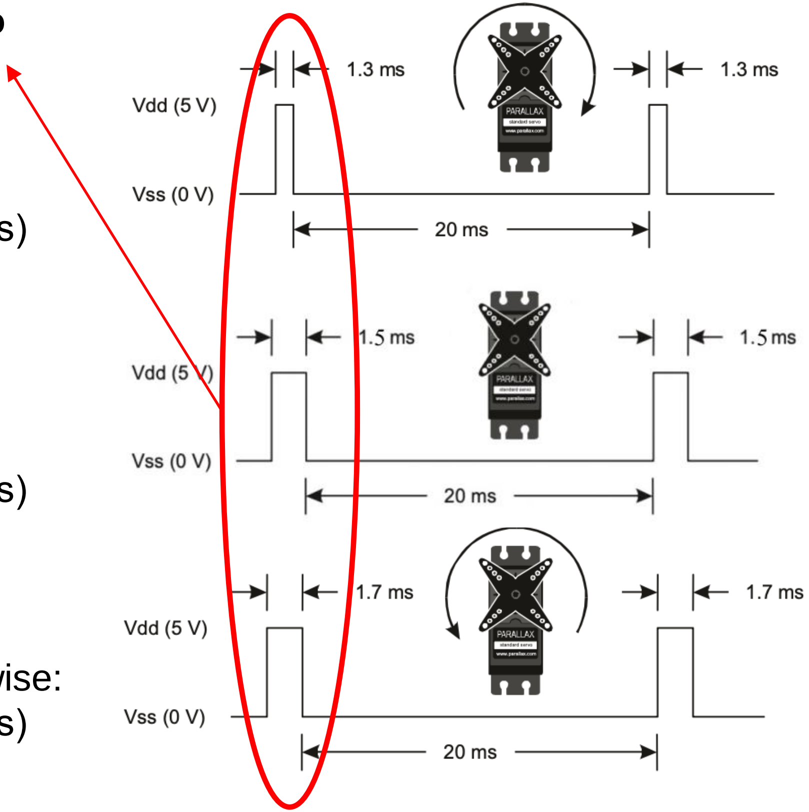
CONTINUOUS SERVO

Pulse length changes to control servo direction

Clockwise:
1300 µs (1.3 ms)

Stopped:
1500 µs (1.5 ms)

Counter-clockwise:
1700 µs (1.7 ms)



- Each pulse is from <u>1300 to 1700</u> µs (microseconds) in **duration**

- The pulses **repeat** about 50 times each second---once <u>every 20 milliseconds</u>

MOTOR CONSIDERATIONS: SPEED

- **More speed requires more voltage**

High speed motor	Ordinary motor
Voltage requirement: 6 - 8 VDC	Voltage requirement: 4 - 6 VDC
Speed: up to 180 RPM	Speed: up to 50 RPM

SPEED ↑ ⇒ VOLTAGE ↑

Speed: (For same power input)

Video [High speed servo v/s Ordrinary servo]

MOTOR CONSIDERATIONS: TORQUE

- **More torque requires more current**

High speed motor	*Ordinary motor*
Current requirement: 15 - 180 mA	Current requirement: 15 - 200 mA
Torque: 1.6 +/- 0.8 kg-cm @ 7.4 V	Torque: 2.74 kg-cm @ 6 V

Torque ⬆ ⇒ Current ⬆

Torque: (For same power input)

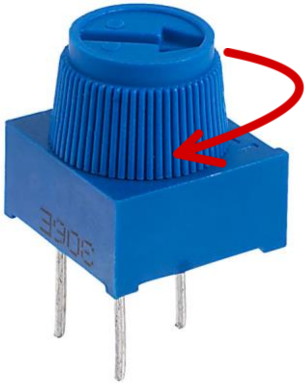
High Speed Servo v/s Ordinary Servo (Higher Torque)

Video

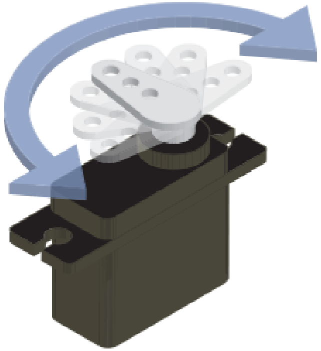
Video

CHALLENGE ACTIVITY

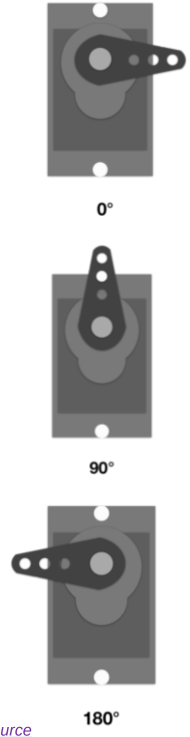
Control servo motor using potentiometer and 3 LEDs to glow at 0, 90 and 180



Source

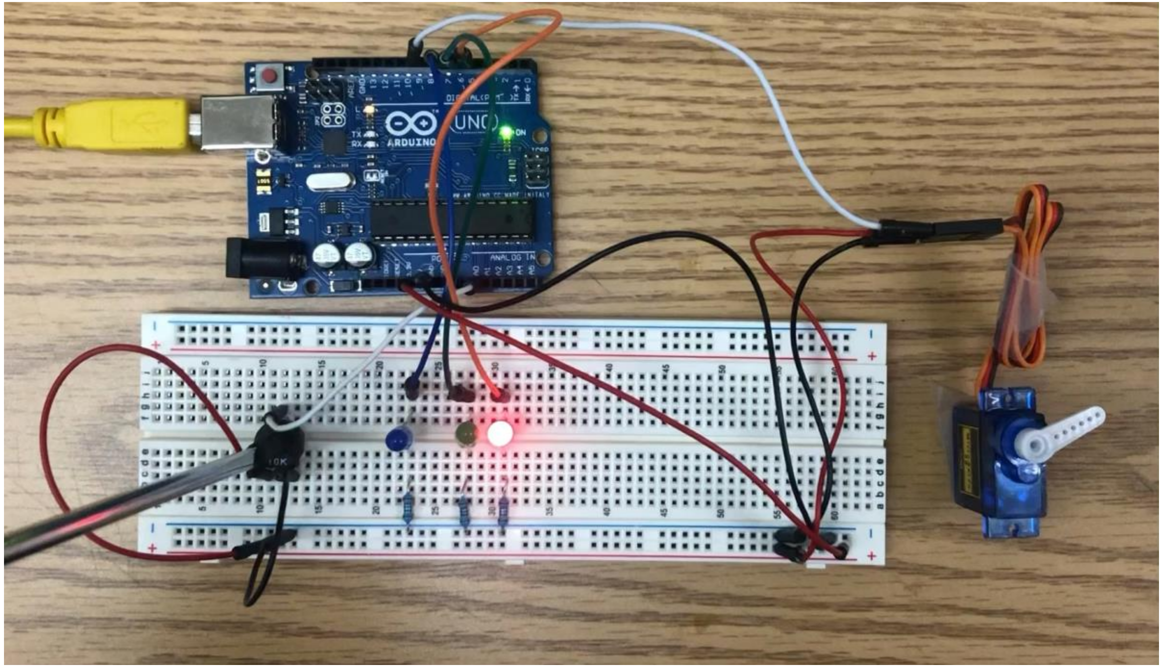


Source



Source

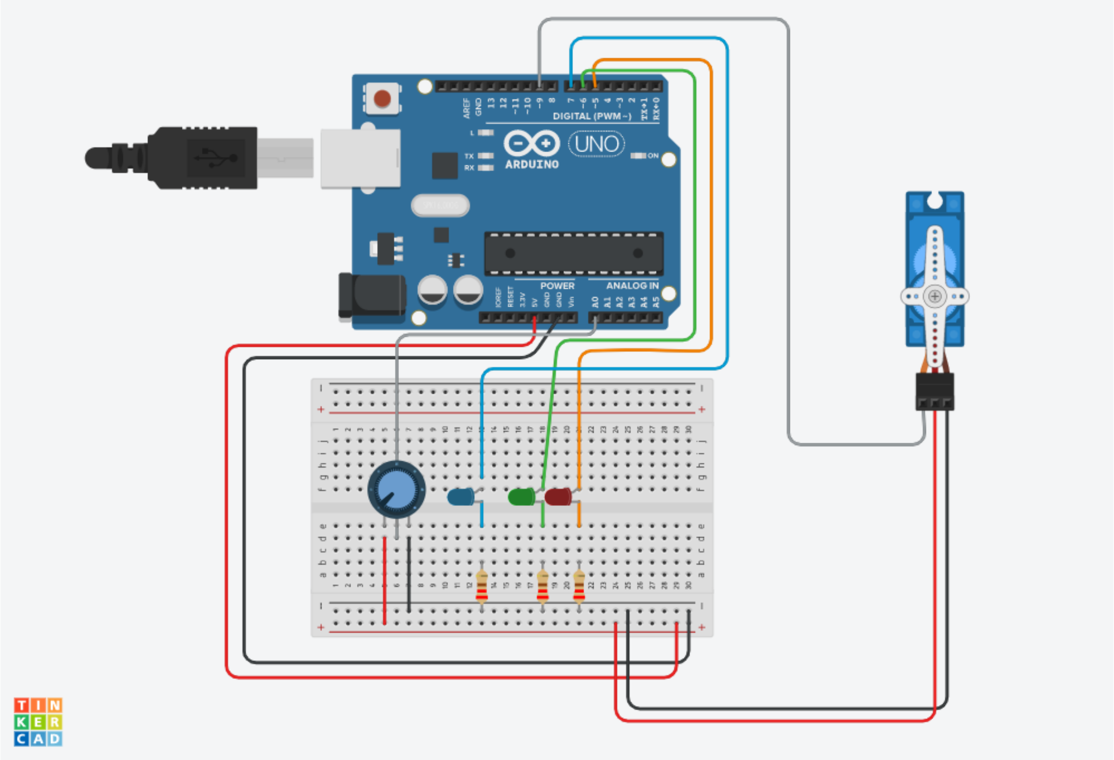
CHALLENGE ACTIVITY - SOLUTION



Video

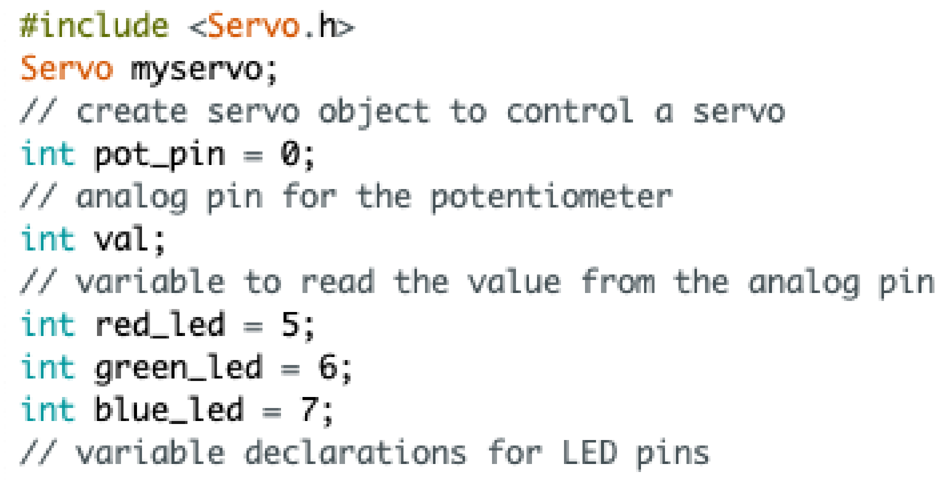
CHALLENGE ACTIVITY - CIRCUIT

CIRCUIT

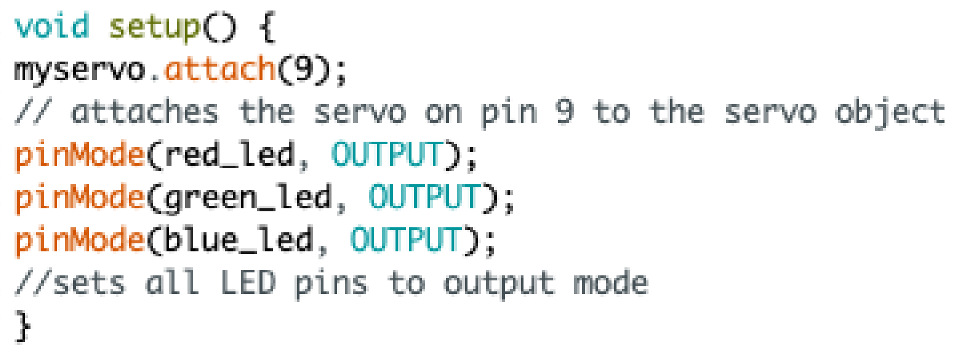


CHALLENGE ACTIVITY - CODE

CODE



```
#include <Servo.h>
Servo myservo;
// create servo object to control a servo
int pot_pin = 0;
// analog pin for the potentiometer
int val;
// variable to read the value from the analog pin
int red_led = 5;
int green_led = 6;
int blue_led = 7;
// variable declarations for LED pins
```



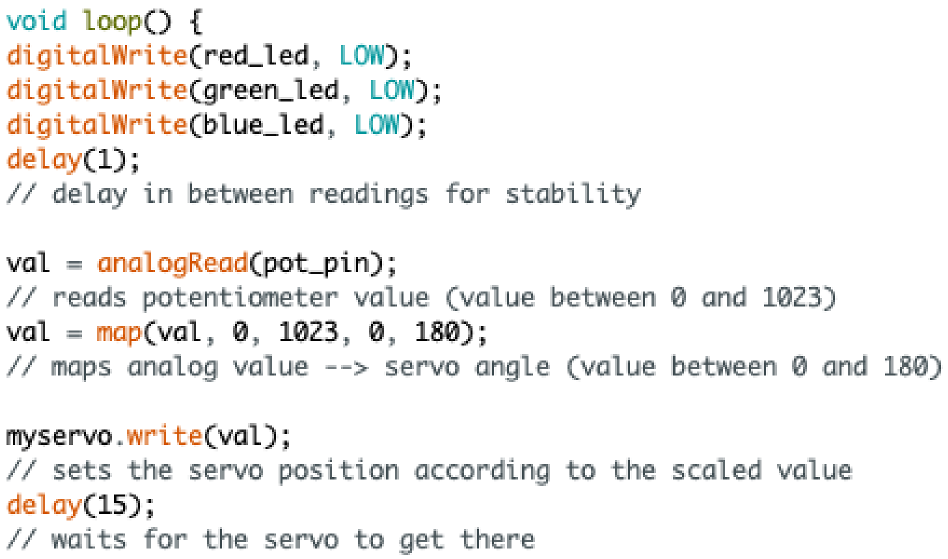
```
void setup() {
myservo.attach(9);
// attaches the servo on pin 9 to the servo object
pinMode(red_led, OUTPUT);
pinMode(green_led, OUTPUT);
pinMode(blue_led, OUTPUT);
//sets all LED pins to output mode
}
```

Program

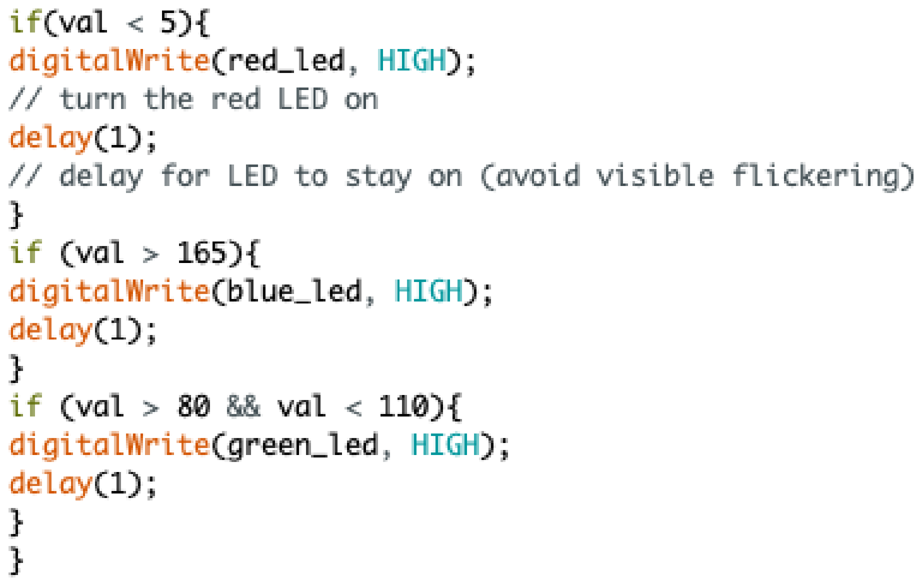
(Contd.)

CHALLENGE ACTIVITY - CODE

CODE



```
void loop() {
digitalWrite(red_led, LOW);
digitalWrite(green_led, LOW);
digitalWrite(blue_led, LOW);
delay(1);
// delay in between readings for stability

val = analogRead(pot_pin);
// reads potentiometer value (value between 0 and 1023)
val = map(val, 0, 1023, 0, 180);
// maps analog value --> servo angle (value between 0 and 180)

myservo.write(val);
// sets the servo position according to the scaled value
delay(15);
// waits for the servo to get there
```



```
if(val < 5){
digitalWrite(red_led, HIGH);
// turn the red LED on
delay(1);
// delay for LED to stay on (avoid visible flickering)
}
if (val > 165){
digitalWrite(blue_led, HIGH);
delay(1);
}
if (val > 80 && val < 110){
digitalWrite(green_led, HIGH);
delay(1);
}
}
```

Program

TASK / ACTIVITY

- Find duty cycle and power, when PWM on time is given
- DC motor direction control for two motors
- DC motor speed control for two motors
- DC motor speed and direction control for two motors
- Servo motor angle control (user input)
- Rotational servo motor Calibration
- Rotational servo speed control
- Rotational servo motor direction control

Thank You!

Questions and Feedback?