

# Driving Different Motors with Arduino



Ben Cheng

RISD ID

April 20, 2024

# Outline I

## 1 Basic AC/DC Motor

Basics of Motors

Analog Control

Digital Control

## 2 Stepper Motor

## 3 Servo Motor

## 1 Basic AC/DC Motor

Basics of Motors

Analog Control

Digital Control

## 2 Stepper Motor

## 3 Servo Motor

# Basic Principle of Motors

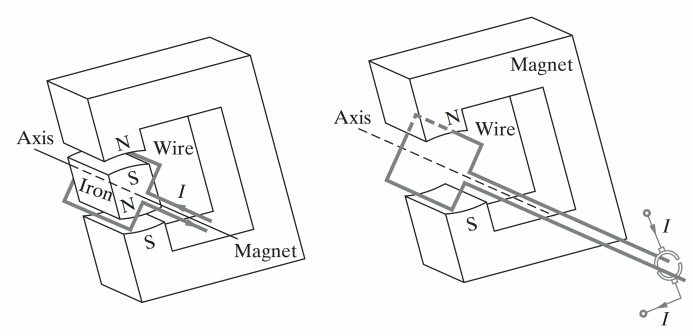
## Theorem (Biot-Savart Law)

$$\mathbf{B} = \frac{\mu_0}{4\pi} \oint_{C_1} \frac{I d\mathbf{l}' \times \hat{\mathbf{R}}}{R^2}$$

## Theorem (Lorentz's equation)

$$\mathbf{F} = q\mathbf{v} \times \mathbf{B}$$

# AC vs DC Motors



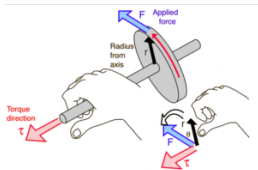
**Figure:** Simple dc motor. (Inan 2015)

# Torque and Power

How much torque is needed to lift this?

## Theorem (Torque)

$$\tau = F \times r$$



(Linear Motion Tips)

How much power do I need to drive this motor?

## Theorem (Power)

$$P = I \times V$$

(Energy per unit time)

# Analog Control

## Theorem (Ohm's Law)

$$V = A \times R$$

, or

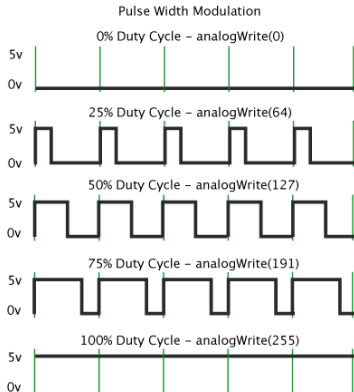
$$A = \frac{V}{R}$$

- Simple, accurate, predictable
- Power source = control unit
- Hard to implement on microcontroller

# Digital Control: PWM

## Pulse Width Modulation

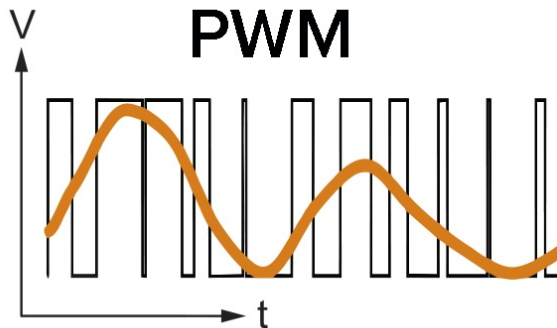
- Easy to implement on microcontroller
- Easy manipulation
- Inaccurate approximation
- Power and control unit saperated



(Arduino)



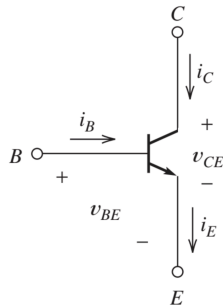
# PWM Approximation



(Thomson Linear)

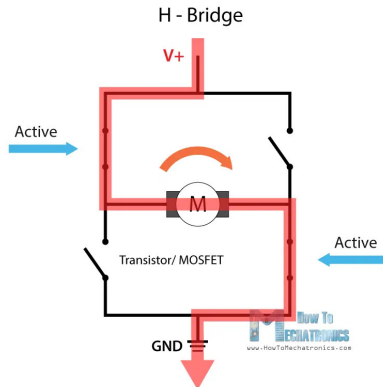
# Saperated Power and Control Unit

- Scale of current:
  - 2A: Fry a human
  - 0.35A: DC Motor
  - 20mA: Arduino Pinout
- Saperate power and signal circuit
  - MOSFET
  - Bipolar Junction Transistor
  - Relay
  - H-Bridge



BJT. (Hambly 2018)

# L298N Motor Driver



(HowToMechatronics.com)

- ENA to pin 9 (PWM)
- IN1, IN2 to pin 5, 6
- GND and 12V to pwr supply
- OUT1, OUT2 to motor

# PWM Implementation

---

```
1  int speed = 255;
2  String inputStr = "";
3  bool clean = false;
4
5  void setup() {
6      inputStr.reserve(200);
7      Serial.begin(9600);
8      // put your setup code here, to run once:
9      pinMode(5, OUTPUT);
10     pinMode(6, OUTPUT);
11     pinMode(9, OUTPUT);
12
13     digitalWrite(5, 1);
14     digitalWrite(6, 0);
15 }
```

---

# analogWrite()

---

```
1 void loop() {  
2     // put your main code here, to run repeatedly:  
3     if (clean) {  
4         analogWrite(9, inputStr.toInt());  
5         Serial.println(inputStr.toInt());  
6         // clear the string:  
7         inputStr = "";  
8         clean = false;  
9     }  
10  
11 }
```

---

## 1 Basic AC/DC Motor

Basics of Motors

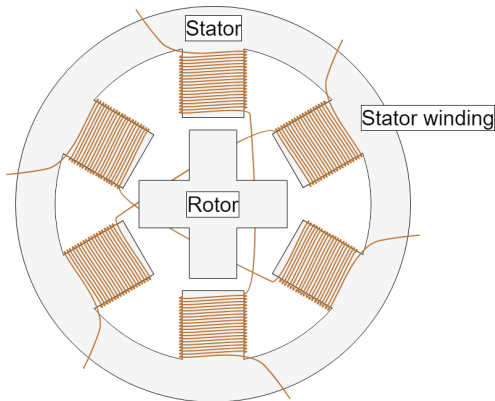
Analog Control

Digital Control

## 2 Stepper Motor

## 3 Servo Motor

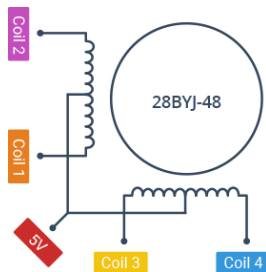
# Stepper Motor



(monolithicpower.com)

## 28BYJ-48 Stepper Motor

- 4 coils
- 32 steps per revolution
- plus 64:1 gear ratio
- 2048 steps per revolution
- 5 pins



(lastminuteengineering.com)

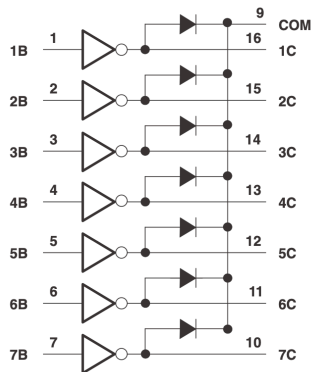


## Stepper vs DC Motor

- Stepper is very accurate
- More difficult to control
- More power draw, more heat generation
- Stepper is suitable for short period, precise application.
- DC motor is for continuous, powerful application.

# ULN2003

- 7 Darlington pair (BJT)
- 500mA emitter current



(Texas Instrument)

# Stepper Library

```
1 #include <Stepper.h>
2
3 // initialize the stepper library on pins 8 through 11:
4 Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);
5
6 void setup() {
7     // set the speed at 6 rpm:
8     myStepper.setSpeed(6);
9     // initialize the serial port:
10    Serial.begin(9600);
11 }
12
13 void loop() {
14     // step one revolution in one direction:
15     Serial.println("clockwise");
16     myStepper.step(2048);
17     delay(500);
18 }
```

## 1 Basic AC/DC Motor

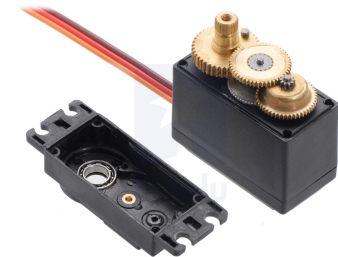
Basics of Motors  
Analog Control  
Digital Control

## 2 Stepper Motor

## 3 Servo Motor

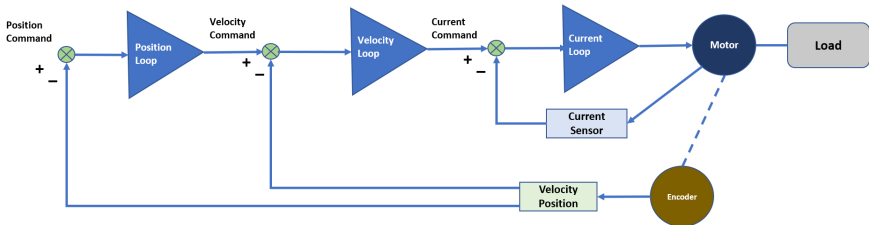
# Inside Servo Motor

- DC Motor
- Gearbox
- Encoder
- Controller IC



(popolu.com)

# Increasing Precision of DC Motor



(Kollmorgan)

# Sweep

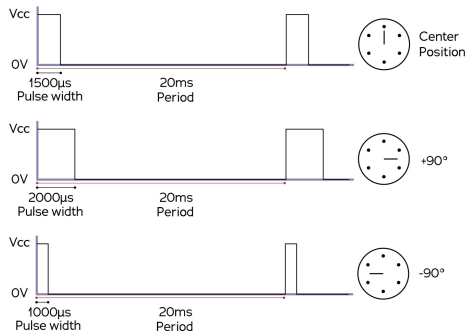
- Orange to pin 9, red to 3V3, brown to GND

---

```
1 #include <Servo.h>
2 Servo myservo;
3 void setup() {
4     myservo.attach(9);
5 }
6 void loop() {
7     int pos = 0;
8     for (pos = 0; pos <= 180; pos += 1) {
9         myservo.write(pos);
10        delay(30);
11    }
12    for (pos = 180; pos >= 0; pos -= 1) {
13        myservo.write(pos);
14        delay(30);
15    }
16 }
```

---

# Servo Pulse



(Wikipedia user Hforesti)



# Servo vs Stepper vs DC Motor

## Servo

- precise
- digital
- for less than 1 rev

## Stepper

- precise
- digital
- for short period

## Servo

- not for angular precision
- digital/analog
- for continuous rev