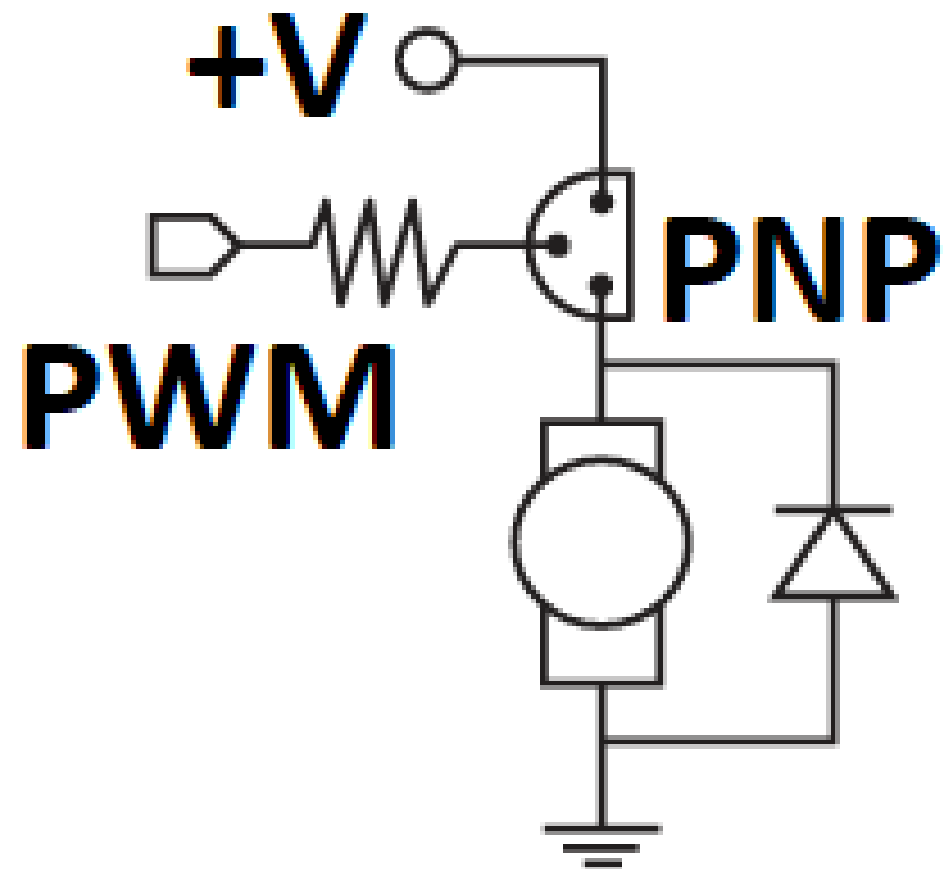
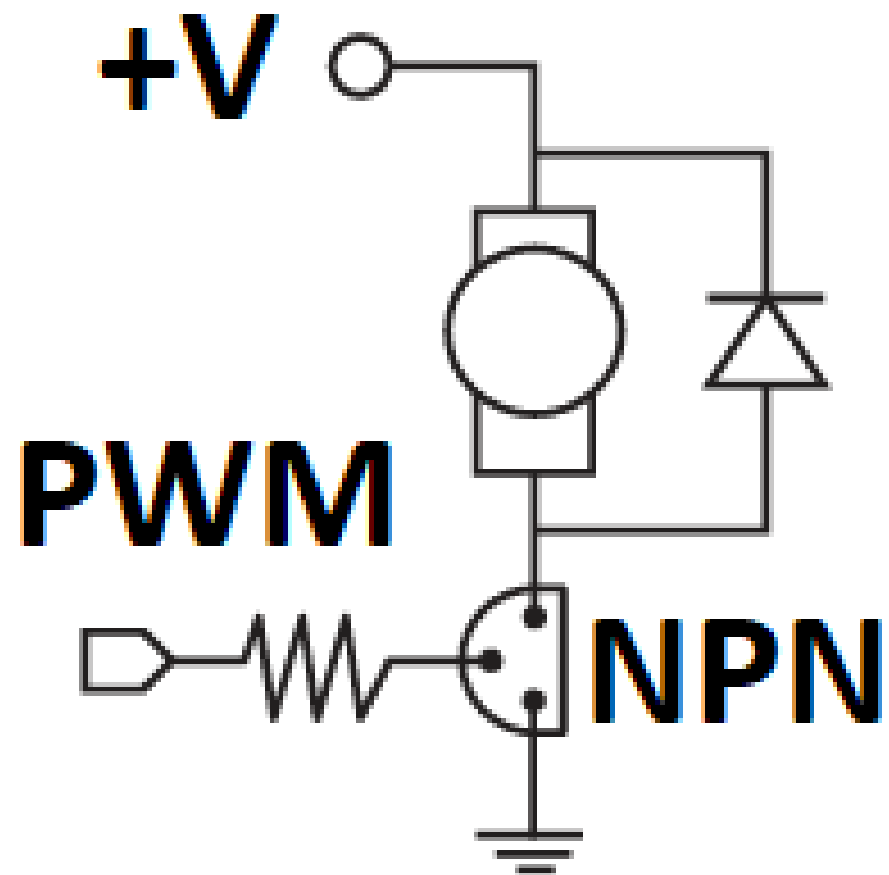




Microcontrollers & Applications

Lecture 6.1: DC & Vibration Motors

DC Motor: Structure



DC Motor Control: Hardware Connection

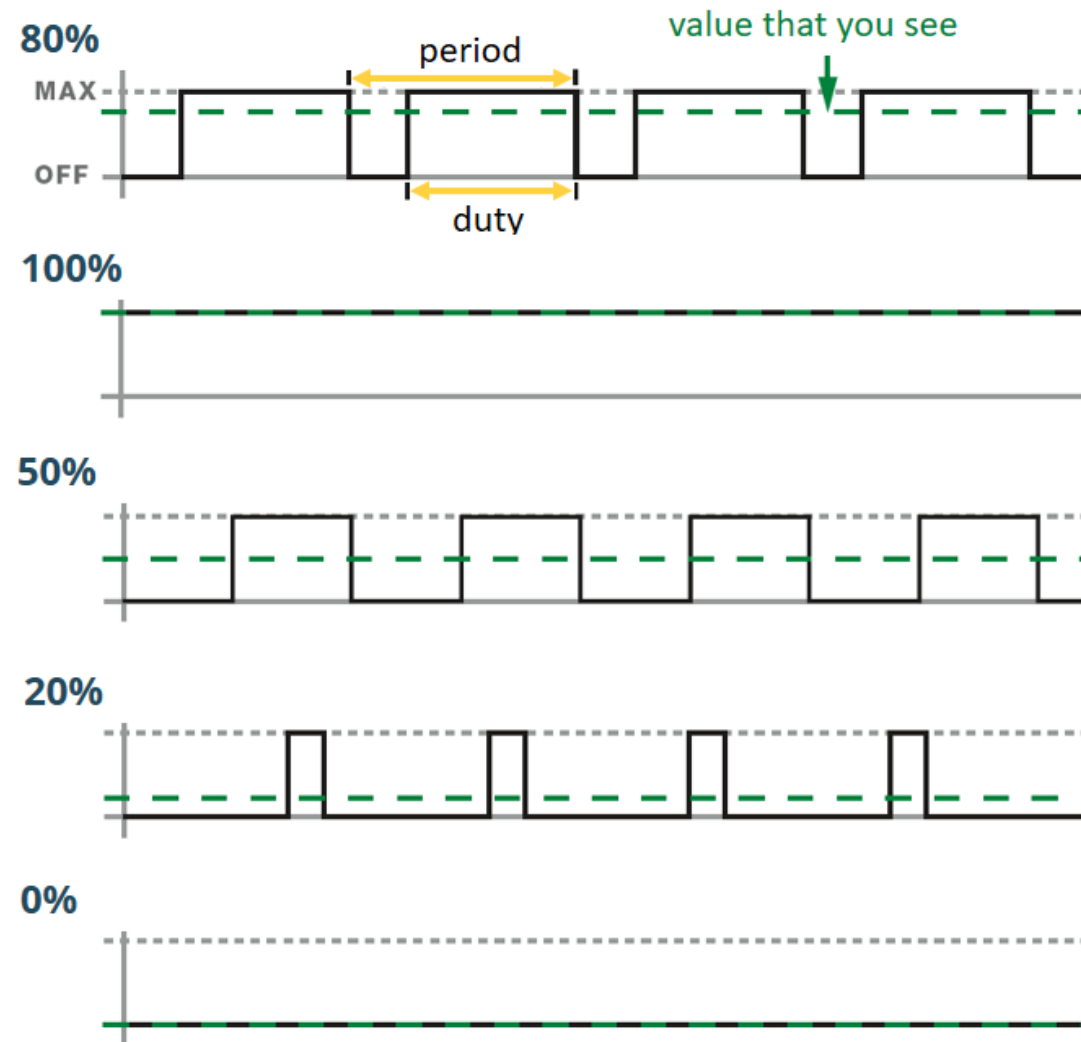
NPN Transistor

- +V connected to Motor->Positive terminal
- NPN->Collector connected to Motor->Negative terminal
- NPN->Base connected to PWM control pin via a resistor
- NPN->Emitter connected to GND
- Diode->Anode connected to Motor->Negative terminal
- Diode->Cathode connected to Motor->Positive terminal
- Diode is called as Flyback

PNP Transistor

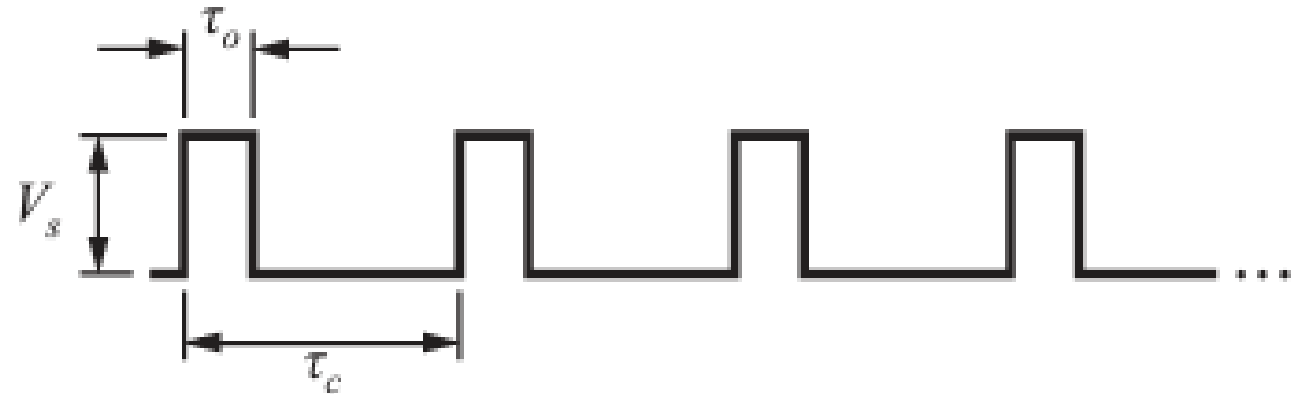
- +V connected to PNP->Emitter
- PNP->Collector connected to Motor->Positive terminal
- PNP->Base connected to PWM control pin via a resistor
- Motor->Negative terminal connected to GND
- Diode->Anode connected to Motor->Negative terminal
- Diode->Cathode connected to Motor->Positive terminal
- Diode is called as Flyback

DC Motor: PWM Signal



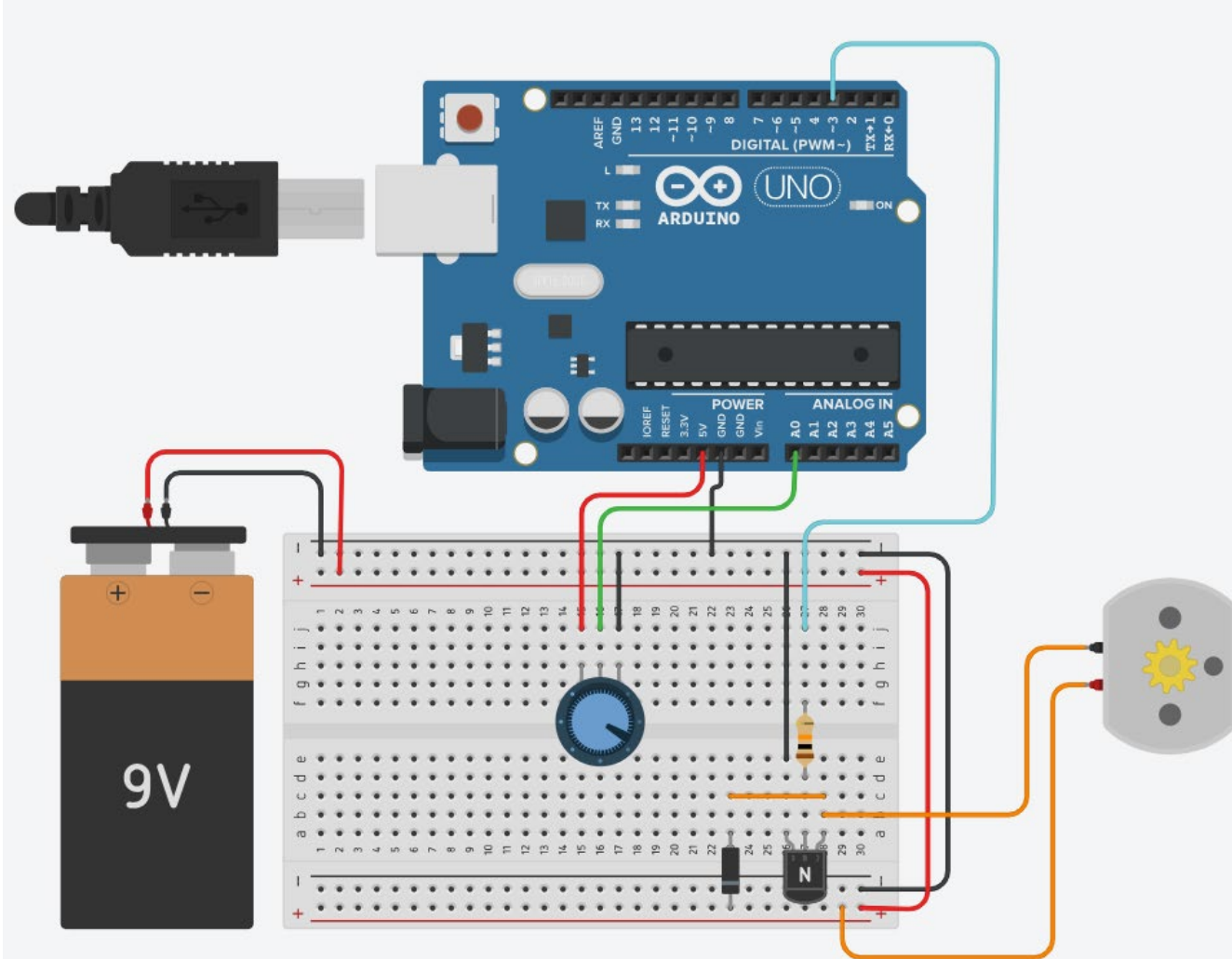
DC Motor: PWM Control

- PWM is a common technique for supplying variable levels to slowly-change electrical devices such as LEDs, heaters, DC motors, etc.
- Arduino cannot give an actual analog output.
- If the maximum voltage (V_s) transfers maximum power to the load, the efficient voltage (V_{eff}) transfers to efficient power to the load by the ratio of T_o to T_c .
- The efficient power never means that there is a linear relation between motor speed and the V_{eff} , be careful ! 😊



The effective voltage is $V_{\text{eff}} = V_s \frac{T_o}{T_c}$
 $\frac{T_o}{T_c}$ is called the duty cycle

DC Motor Control: Arduino Example (1)



DC Motor Control: Arduino Example (2)

```
#define POT A0
#define MOTOR 3

void setup() {
  pinMode(MOTOR,OUTPUT);
}

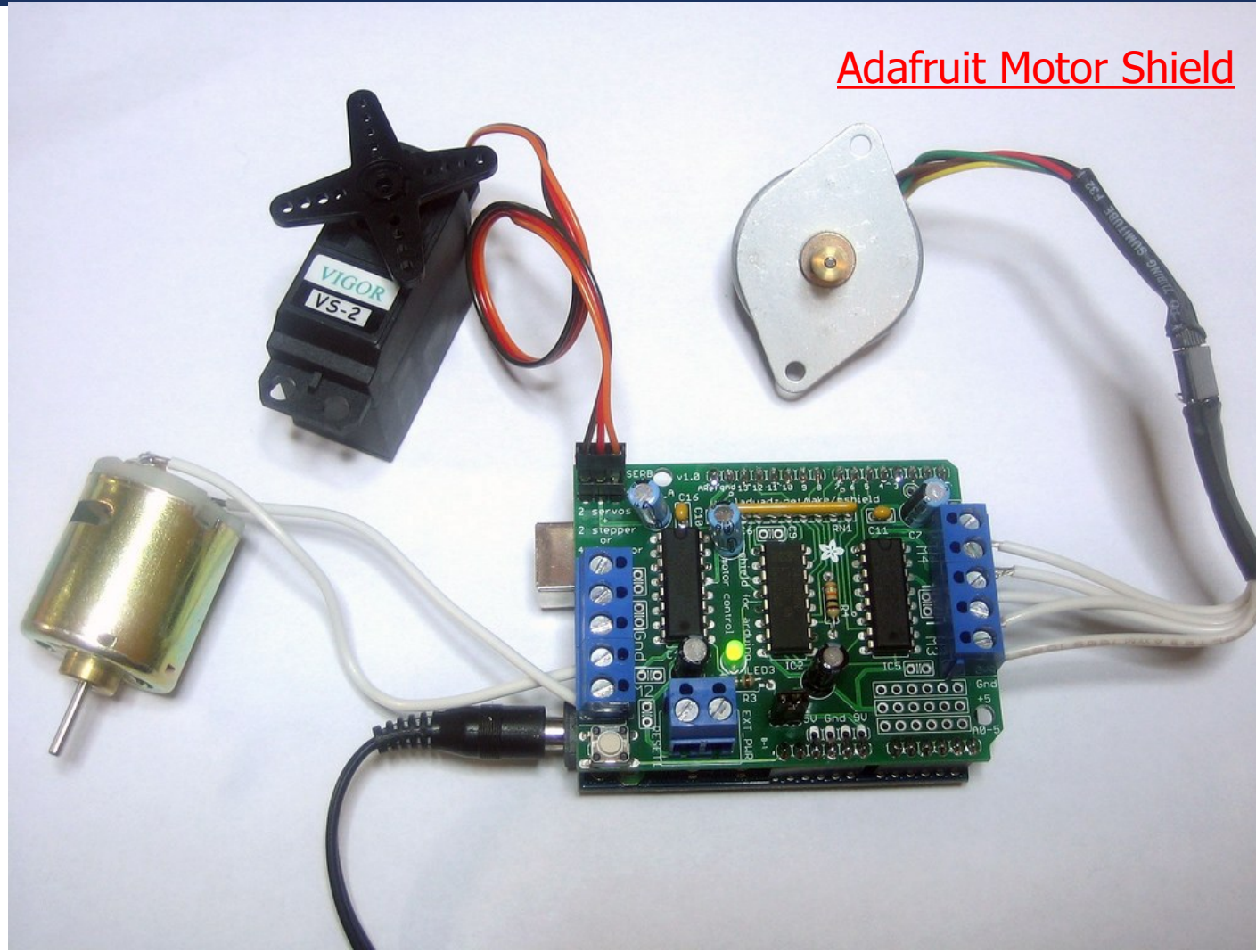
void loop() {
  unsigned int potValue = analogRead(POT);
  unsigned int motorValue;

  // motorValue = potValue >> 2; // OR
  motorValue = map(potValue, 0, 1023, 0, 255);
  analogWrite(MOTOR, motorValue);

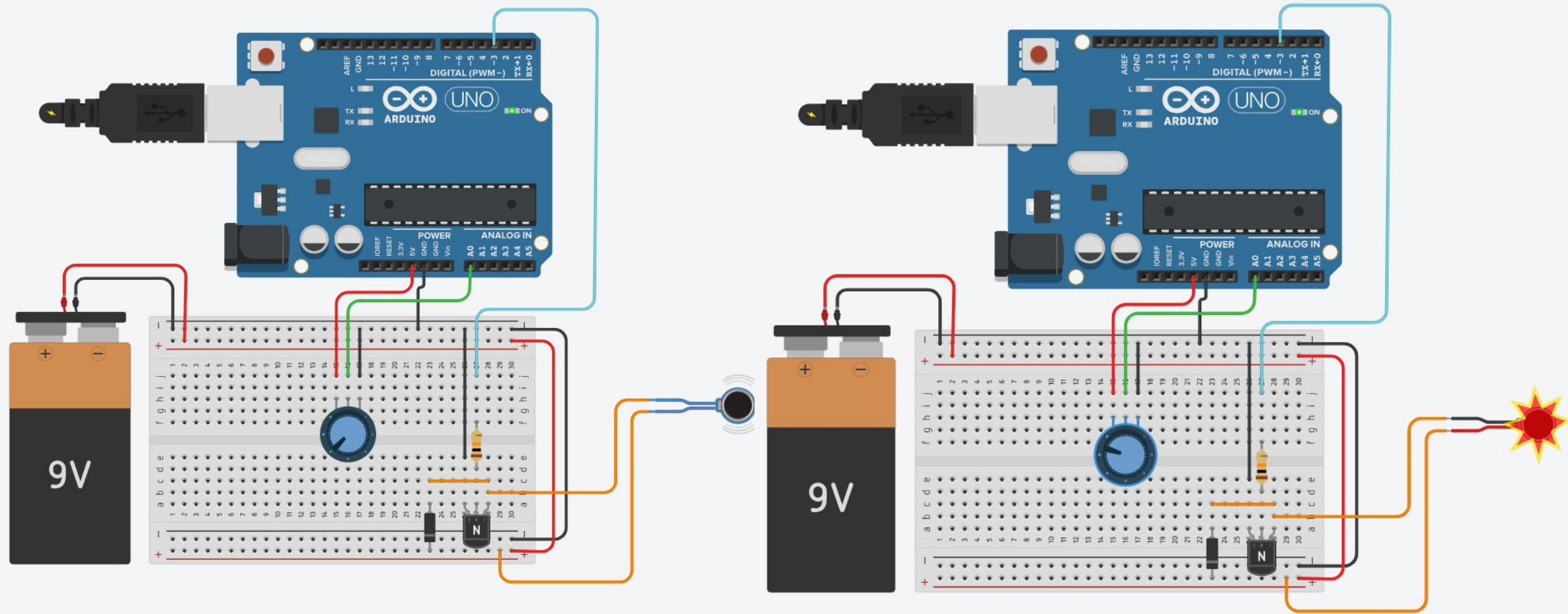
  delay(100);
}
```

- Arduino's analog inputs have 10-bit resolution.
 - From 0 to 1023.
- Arduino's PWM outputs have 8-bit resolution.
 - From 0 to 255.
- Raspberry Pi Pico's analog inputs have 12-bit resolution.
 - From 0 to 4095.
- Raspberry Pi Pico's PWM outputs have 16-bit resolution.
 - From 0 to 65535.

Step Motor Control: Arduino Example (3)



Vibration Motor Control: Arduino Example (1)



Vibration Motor Control: Arduino Example (2)

```
#define POT A0
#define MOTOR 3

void setup() {
  pinMode(MOTOR,OUTPUT);
}

void loop() {
  unsigned int potValue = analogRead(POT);
  unsigned int motorValue;

  // motorValue = potValue >> 2; // OR
  motorValue = map(potValue, 0, 1023, 0, 255);
  analogWrite(MOTOR, motorValue);

  delay(100);
}
```

- Vibration motor is more naive than the DC motors.
- Control is almost same with DC motors.
- Higher voltages can damage the vibration motors easily !
 - See the simulation on the right of the previous circuit schematics 😊

Left to Students: Morse Code Generator

- Develop an «Arduino SMS Vibrating System» with circuit and code to
 - Read message from the serial port,
 - Translate the message to the Morse code,
 - Vibrate it to a blind user ☹️.
- You may want to investigate «<https://www.arduino.cc/education/morse-code-project/>»

A	· —	N	— ·	1	· — — — —	?	· · — — — —
B	— · · ·	O	— — — —	2	· · — — — —	!	— · · — — — —
C	— · · · ·	P	· — — — ·	3	· · · — — —	,	· — — — — ·
D	— · ·	Q	— — — · —	4	· · · · —	,	— — — — · — — —
E	·	R	· — ·	5	· · · · ·	;	— · — — — ·
F	· · — ·	S	· · ·	6	— · · · ·	:	— — — — · · ·
G	— — — ·	T	—	7	— — — · · ·	+	· — — — ·
H	· · · ·	U	· · —	8	— — — — · ·	-	— · · · · —
I	· ·	V	· · · —	9	— — — — — ·	/	— · · — ·
J	· — — — —	W	· — — —	0	— — — — —	=	— · · · —
K	— · —	X	— · · —				
L	· — · ·	Y	— · — — —				
M	— — —	Z	— — — · ·				



Thanks for
listening 😊

YALÇIN İŞLER

Assoc. Prof.