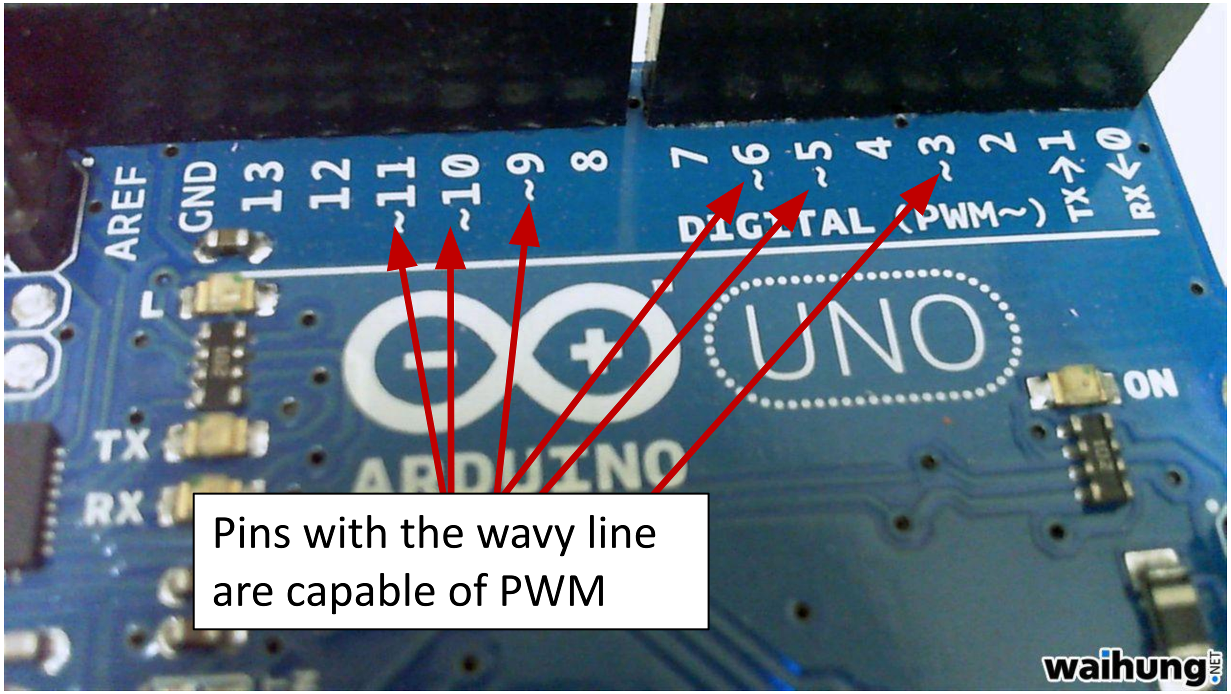
Arduino PWM

(Pulse Width Modulation)



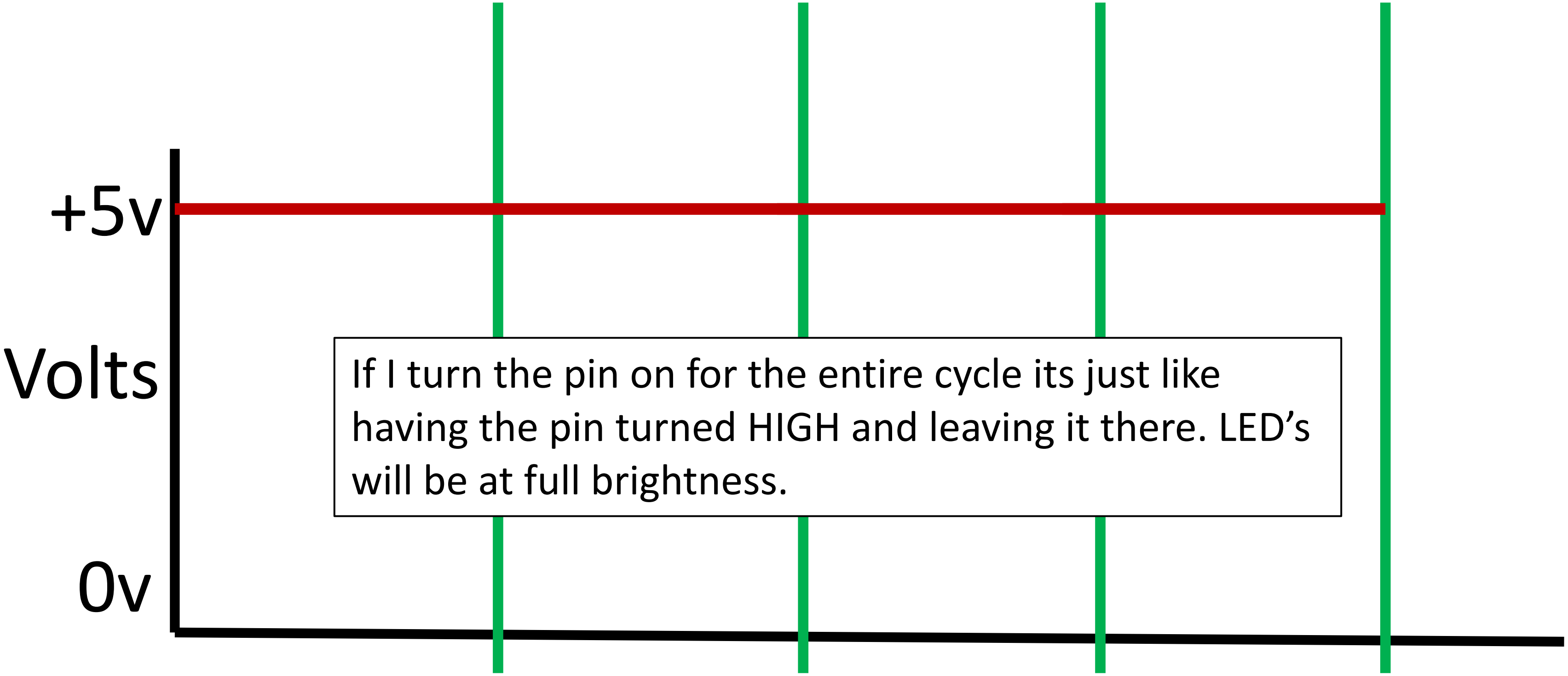
Pins with the wavy line are capable of PWM

When have we seen these measurements before?

KHz	1000 times per second
Mhz	1,000,000 times per second
Ghz	1,000,000,000 times per second

PMW pins on the Arduino run at 490Hz or 980hz

490Hz is equal to 490 times per second.



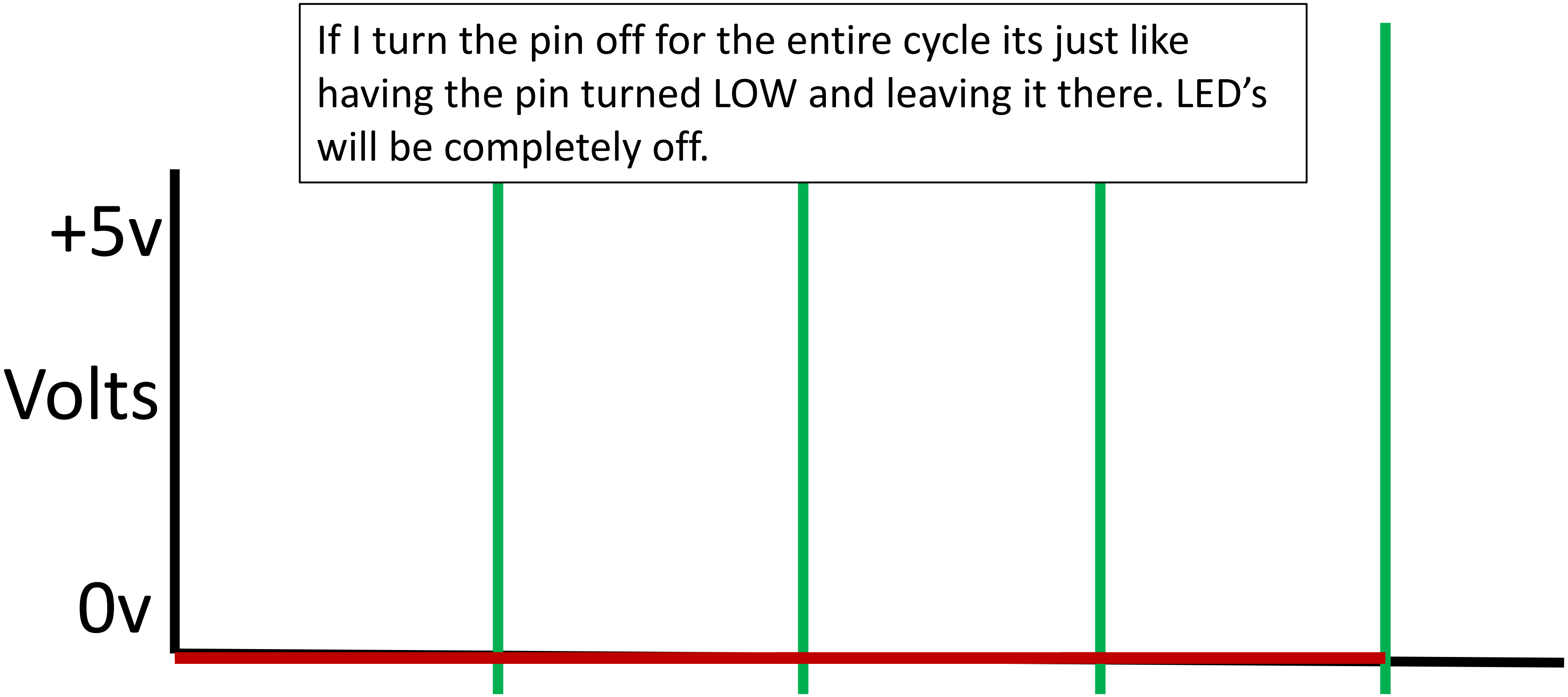
If I turn the pin on for the entire cycle its just like having the pin turned HIGH and leaving it there. LED’s will be at full brightness.

Time (Green lines are at pin Freq. to show 1 full second you would need 490 green lines on the graph)

PMW pins on the Arduino run at 490Hz or 980hz

490Hz is equal to 490 times per second.

If I turn the pin off for the entire cycle its just like having the pin turned LOW and leaving it there. LED's will be completely off.

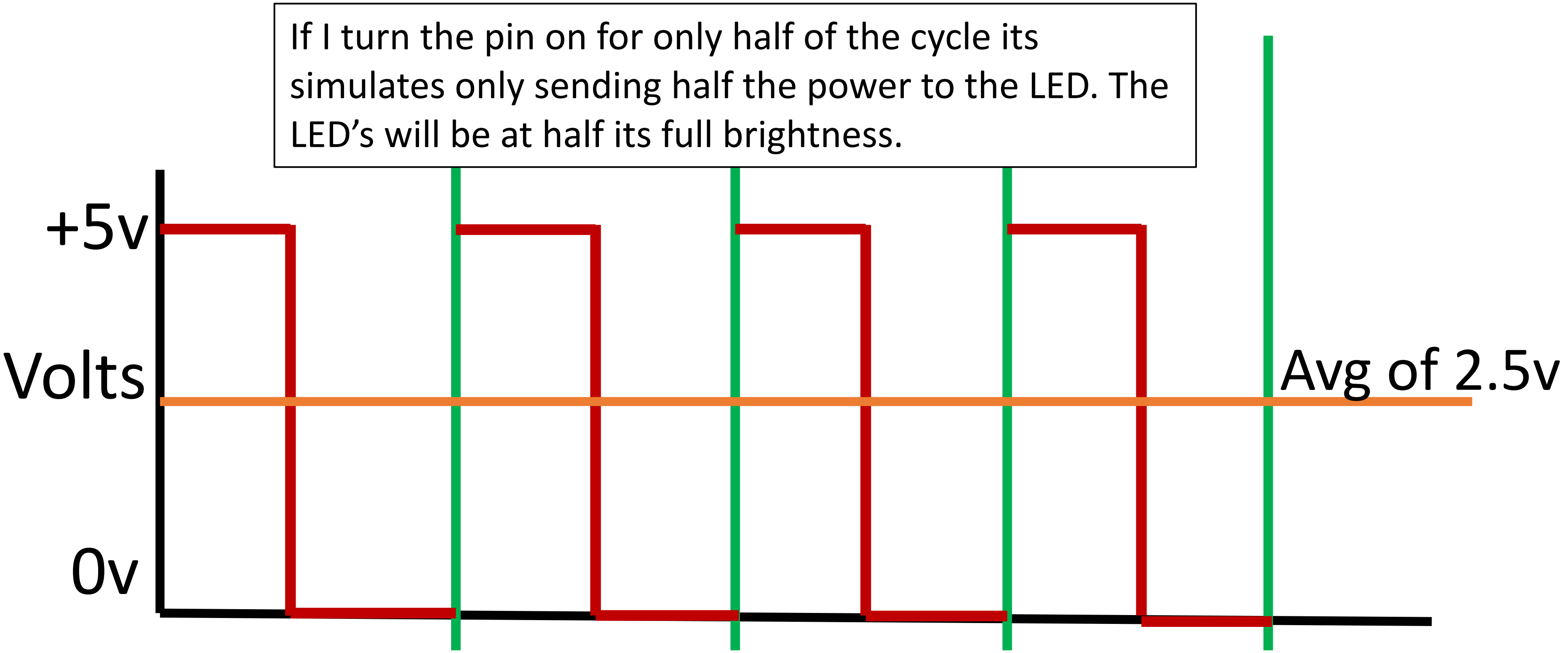


Time (Green lines are at pin Freq. to show 1 full second you would need 490 green lines on the graph)

PMW pins on the Arduino run at 490Hz or 980hz

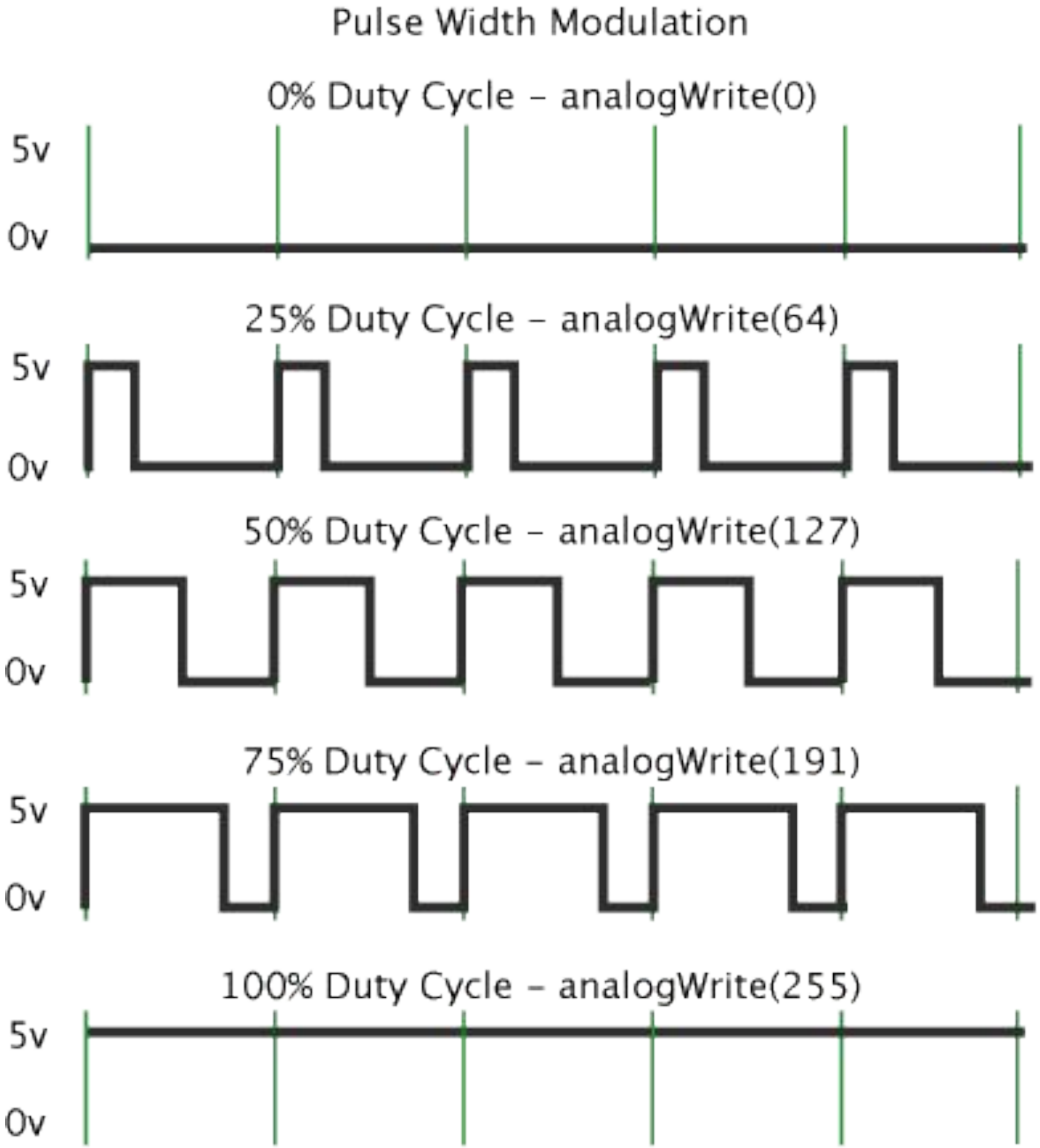
490Hz is equal to 490,000 times per second.

If I turn the pin on for only half of the cycle its simulates only sending half the power to the LED. The LED's will be at half its full brightness.



Time (Green lines are at pin Freq. to show 1 full second you would need 490 green lines on the graph)

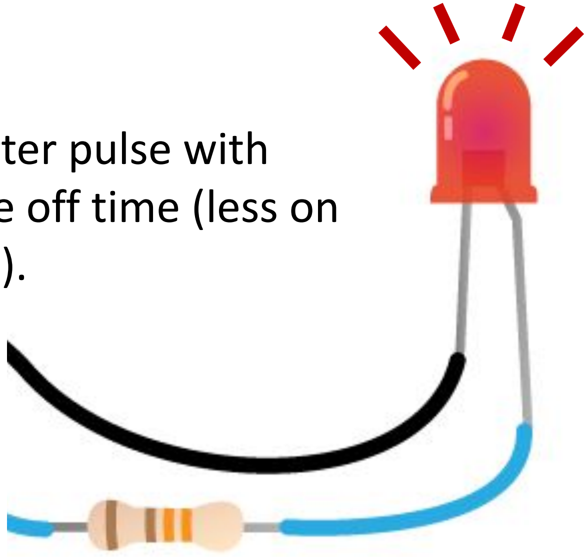
The time that the pin spends turned on is called the duty cycle.


Pulse Width Modulation
0% Duty Cycle - analogWrite(0)
5v
0v
25% Duty Cycle - analogWrite(64)
5v
0v
50% Duty Cycle - analogWrite(127)
5v
0v
75% Duty Cycle - analogWrite(191)
5v
0v
100% Duty Cycle - analogWrite(255)
5v
0v

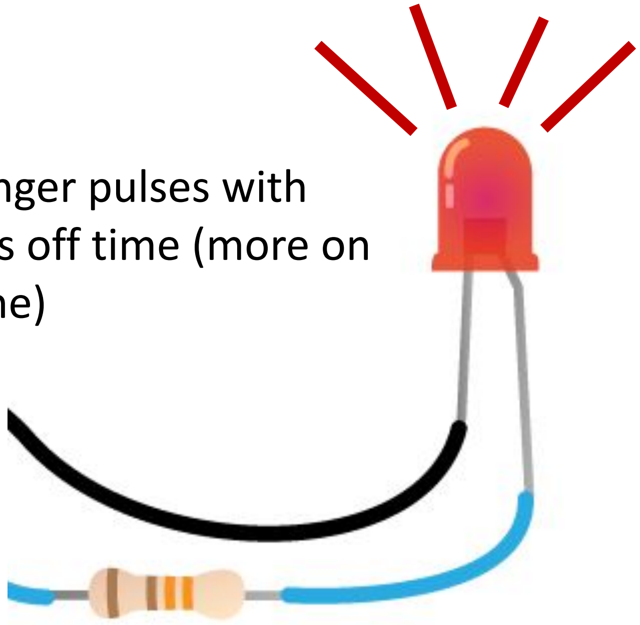
What can you do with Pulse Width Modulation?

-Digitally control the brightness of the LED using only 5 volts.

Shorter pulse with more off time (less on time).

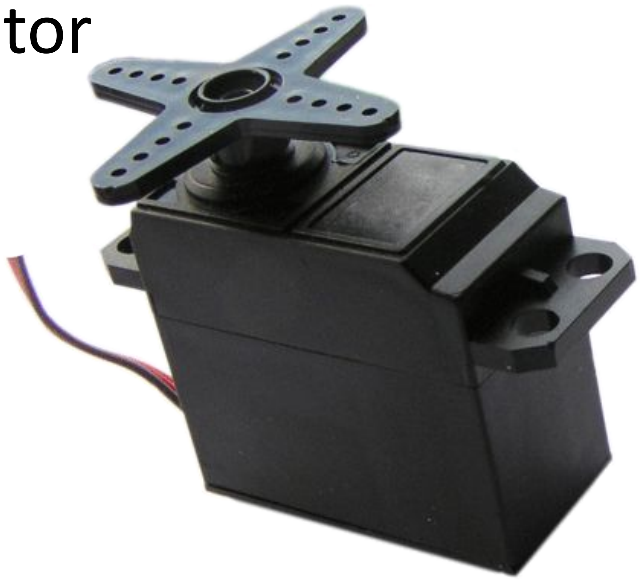


Longer pulses with less off time (more on time)



More uses for PWM

Control the position of an actuator



Control the speed of a motor

Generate sounds

```
int led1 = 3;

void setup() {
    pinMode(led1, OUTPUT);
}

void loop() {
    analogWrite(led1, 10);
    delay(1000);
    analogWrite(led1, 127);
    delay(1000);
    analogWrite(led1, 255);
    delay(1000);
}
```

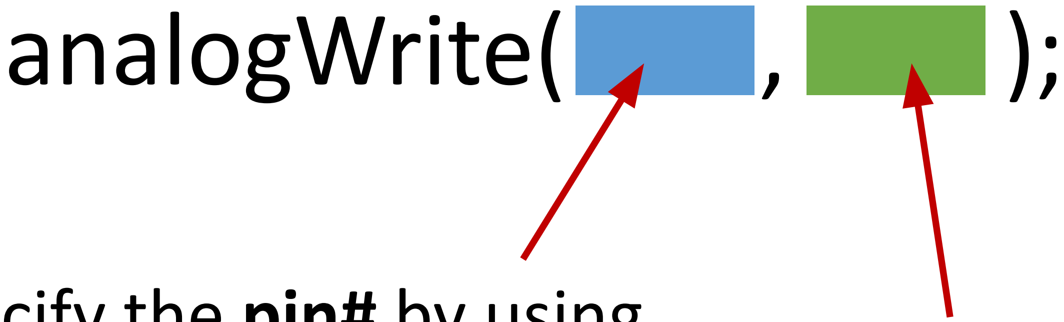
Programming a PMW pin in the Arduino Language

The command (function) we use to tell a pin to pulse at a certain rate is

analogWrite();

analogWrite();
Needs two pieces of information to do its job.
1. What I/O pin.
2. What rate to pulse at.

The command (function) we use to tell a pin to pulse at a certain rate is



analogWrite(🟦, 🟩);

Specify the **pin#** by using the actual integer of the pin or a variable name containing that integer.

Specify the **duty cycle** you would like the pin to operate at. This can be any integer value from 0 to 255 or a variable name that contain an integer from 0 to 255

The command (function) we use to tell a pin to pulse at a certain rate is

```
analogWrite(🟦, 🟩);
```

```
analogWrite(3, 255);
analogWrite(led, 127);
analogWrite(motor2, speed);
```

These are variables that contain integers.

```
int led1 = 3;
int brightness = 0;

void setup() {

  pinMode(led1, OUTPUT);
}



void loop() {
  analogWrite(led1, brightness);
  delay(15);
  brightness = brightness + 5;
  if (brightness >= 255){
    brightness = 0;
  }

}
```

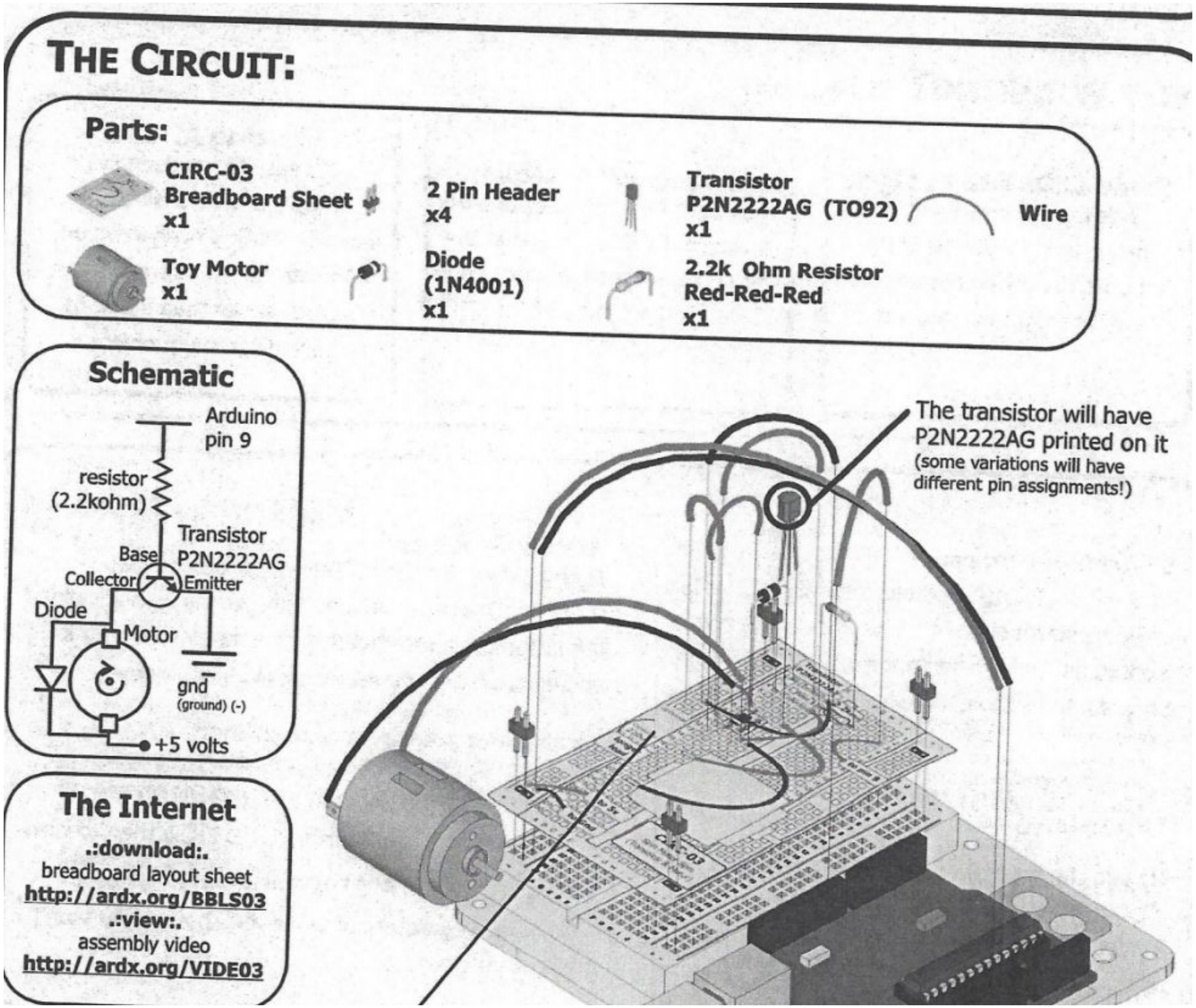
The Arduino is digital right?

The analogWrite() is not really a true analog thing. Its is an approximation of the real analog value.

What is real analog?

A tutorial video on arduino PWM here

Arduino Lab



THE CIRCUIT:

Parts:

- CIRC-03 Breadboard Sheet x1
- 2 Pin Header x4
- Transistor P2N2222AG (TO92) x1
- Wire
- Toy Motor x1
- Diode (1N4001) x1
- 2.2k Ohm Resistor Red-Red-Red x1

Schematic

The transistor will have P2N2222AG printed on it (some variations will have different pin assignments!)

The Internet

.:download:.
breadboard layout sheet
http://ardx.org/BBLS03
.:view:.
assembly video
http://ardx.org/VIDE03