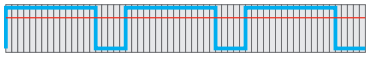
For the most part in computer language one means ON and zero means OFF. This keeps things nice and simple, but what if you want to turn something halfway ON so that it is not all the way ON and not all the way OFF? You can't just use a decimal because digital technology only understands ones and zeros. For this reason some of the pins on your Arduino are labeled PWM or Pulse Width Modulation pins. This means you can send a bunch of ones and zeros real quick and the Arduino board will read these ones and zeros as an average somewhere between one and zero. The red line in the diagrams represent the average. See tables to the right.

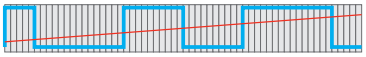
Luckily a lot of the work has been done for you so you don't have to figure out the actual patterns of ones and zeros. All you have to do is pick a number between 0 and 255 and type the command analogWrite. The number zero means the pin is set fully off, the number 255 means the pin is set fully on, and all other numbers set the pin to values between ON (100% or 255) and OFF (0% or 0). You can use PWM on any pin labeled PWM and do not need to set the pin mode before sending an analogWrite command.

PWM signal at 25%

PWM signal at 50%



PWM signal at 75%



PWM signal rising from 25% - 75%

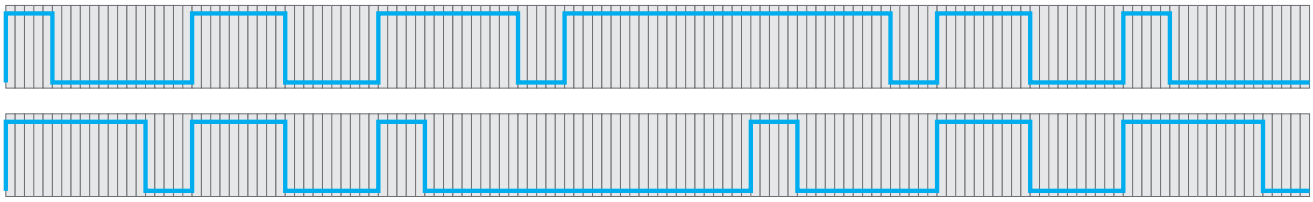
How do you think a PWM signal will affect each of these components compared to a 1 or a 0?

LED: _____

Motor: _____

Piezo: _____

Draw a line through these two charts to show where you believe the PWM value should be.



There are many concepts outside of electrical engineering that are similar to Pulse Width Modulation. Can you list at least three and explain what is being modulated?

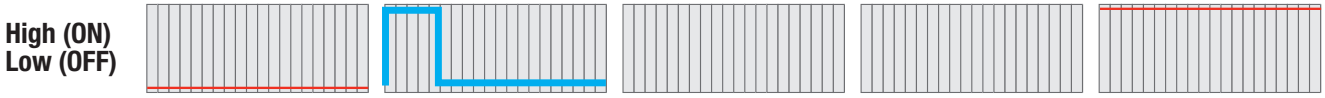
Name:
Date:

Computers and microprocessors only understand two things, ON and OFF. These are represented in a few different ways. There is ON and OFF, One and Zero, or HIGH and LOW. Ones and Zeros are used in the computer language Binary, HIGH and LOW are used with electricity, ON and OFF are plain old human speak.

But what if we want to turn something digital less than 100% ON? Then we use something called PWM, or Pulse Width Modulation. The way your Arduino microprocessor does this is by turning the electricity on a PWM pin ON and then OFF very quickly. The longer the electricity is ON the closer the PWM value is to 100%. This is very useful for controlling a bunch of stuff. For example: the brightness of a light bulb, volume of sound, or the speed of a motor. **These are very basic examples, what else might you need to control that is not only ON or OFF? Explain at least two examples.**

A microprocessor creates a PWM signal by using a built in clock. The microprocessor measures a certain amount of time (also called a window or a period) and turns the PWM pin ON (or HIGH) for the first part of this window and then OFF (or LOW) near the end of the window. The window is filled up with a different length ON (or HIGH) signal depending on the PWM value. If the PWM value is 50% then the PWM signal is ON (or HIGH) for half of the window. If the PWM value is 25% then the PWM signal is ON (or HIGH) for a quarter of the window. The only time the window will not have a LOW value is if the PWM signal is turned completely ON the whole time and therefore equal to 100% ON. The opposite is true as well, if the PWM signal is set to 0% or OFF, then there will not be any HIGH value at the beginning of the window. **Explain in your own words what a PWM window is.**

Below are five different PWM windows. A PWM signal is simply a bunch of PWM windows one after another. Some are missing labels and some are missing diagrams. Please fill in the blanks on the middle three.



PWM percent 0% (No wave)

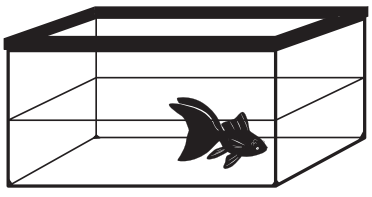
PWM percentage _____

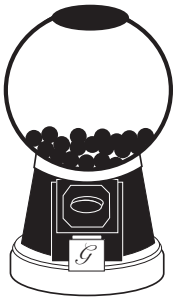
PWM percent 50% (Draw in window)

PWM percent 75% (Draw in window)

PWM percent 100% (No wave)

Below are three different metaphors for a PWM window and a PWM signal. Write the physical item that represents the window and the item or items that represents the signal. Then estimate the PWM percent.





Window: _____ Window: _____ Window: _____

Signal percentage: _____ Signal percentage: _____ Signal percentage: _____