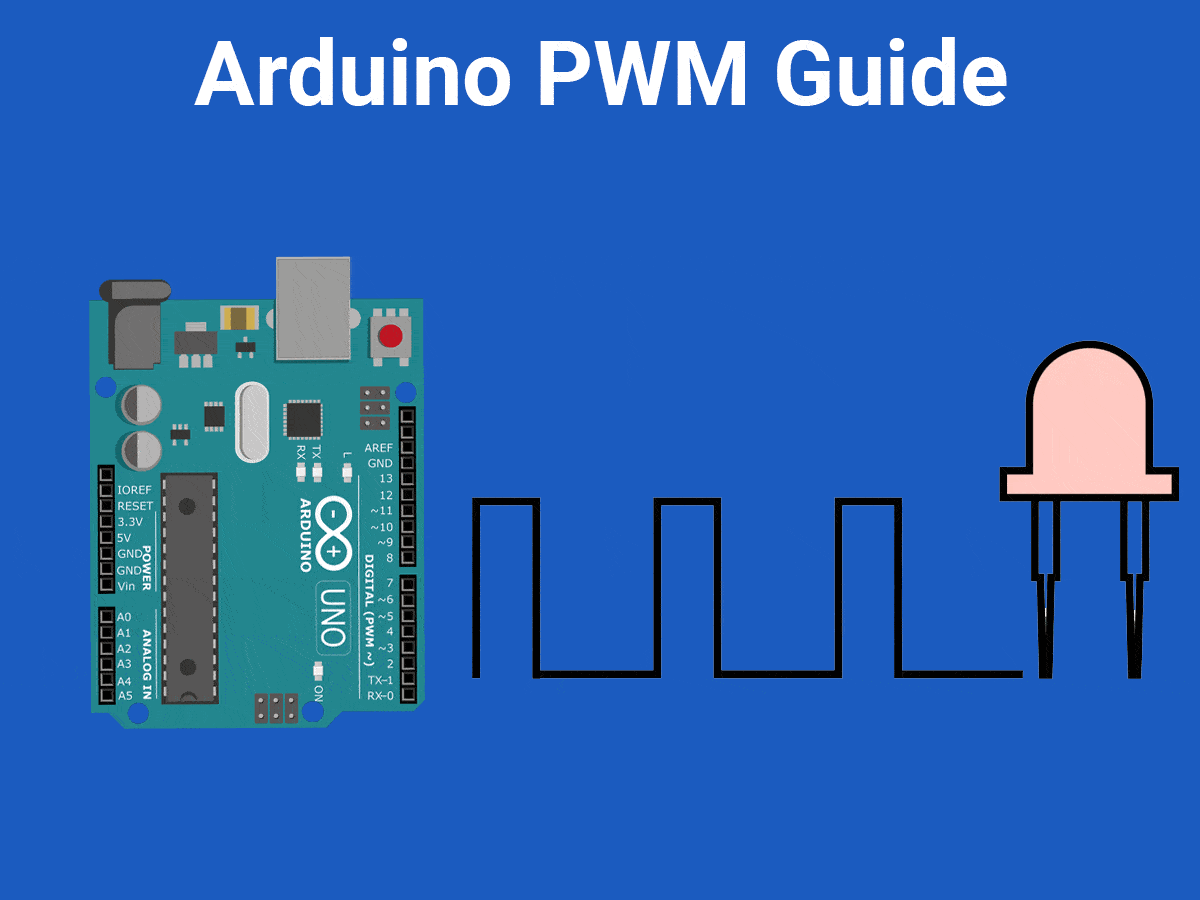
**Understanding Pulse Width Modulation (PWM) and the Arduino Analog Write Command:**

**PWM pins in Arduino:**

Arduino Uno R3 has 6 PWM pins that are 3, 5, 6, 9, 10, and 11. These pins are marked with the negation sign “ ~ “. These pins can generate a pulse as per the given inputs. Arduino supports an 8-bit wide pulse that can have 256 possible levels ( 0 to 255 ).

**Pulse Width Modulation ( PWM ):**

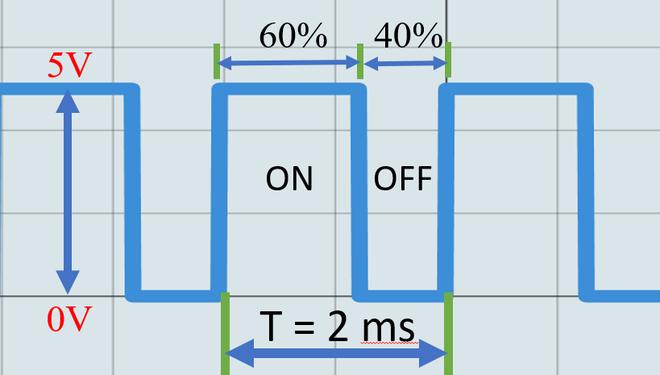
Pulse Width Modulation is a technique to get variable voltage in terms of Digital Input. PWM device generates ON and OFF pulses according to the control signal, which determines the desired voltage level. PWM is used to control the amount of power delivered to the load. It is commonly used for controlling the brightness of LED, the speed of motors, etc.



**Digital to Analog Conversion in Arduino:**

Arduino does not have a dedicated Digital to Analog converter. But It can emulate analog signals using the PWM technique. In PWM, the digital input is converted into a Digital pulse.

It may cause little confusion between analog voltage and digital pulse. Yes, Arduino cannot produce pure analog voltage. The analog output voltage is the average voltage of the “ON” time width of the digital pulse.



From the above fig,Consider the time period of one cycle is 2ms. The “ON” cycle of the pulse is called as the **duty cycle**.

**Duty cycle = ( ON Time / Time period ) \* 100**

**Duty cycle = ( 1.2 / 2 ) \* 100 = 60 %**

This frequent change in on and off mode with a varied length of pulse produces variable analog voltage.

**Duty Cycle to analog Conversion:**

**Analog Output voltage = ( Duty cycle / 100 ) \* Amplitude of the pulse**

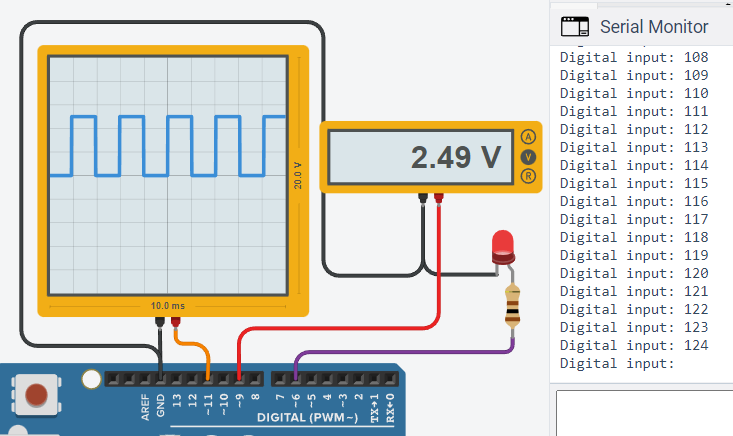
**Analog output voltage = ( 60 / 100 ) \* 5 = 3 Volts**

Of course, the digital pulse did not come to 3 volts. But the “ON” time of the pulse determines the average voltage per time period.

**Components Required:**

1. Arduino Uno R3
2. LED
3. Oscilloscope (optional)
4. Jumper Wires

**Circuit Diagram:**



**Setup:**

1. Connect the positive terminal of the LED to the PWM pin.
2. Connect the negative terminal of the LED to the GND pin.
3. Connect any other output device to PWM as same as above steps.
4. Upload the code to Arduino.

**Arduino code:**

**void** setup()

{

pinMode(11,OUTPUT);

pinMode(9,OUTPUT);

pinMode(6,OUTPUT);

**void** loop()

{

**for**(**int** digitalInput=0; digitalInput<255 ; digitalInput++)

{

analogWrite(11, digitalInput);

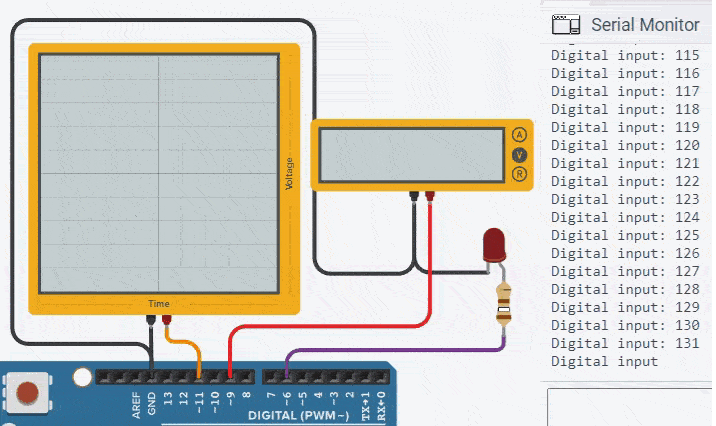
analogWrite(9, digitalInput);

analogWrite(6, digitalInput);

}

}

**OUTPUT:**

****