

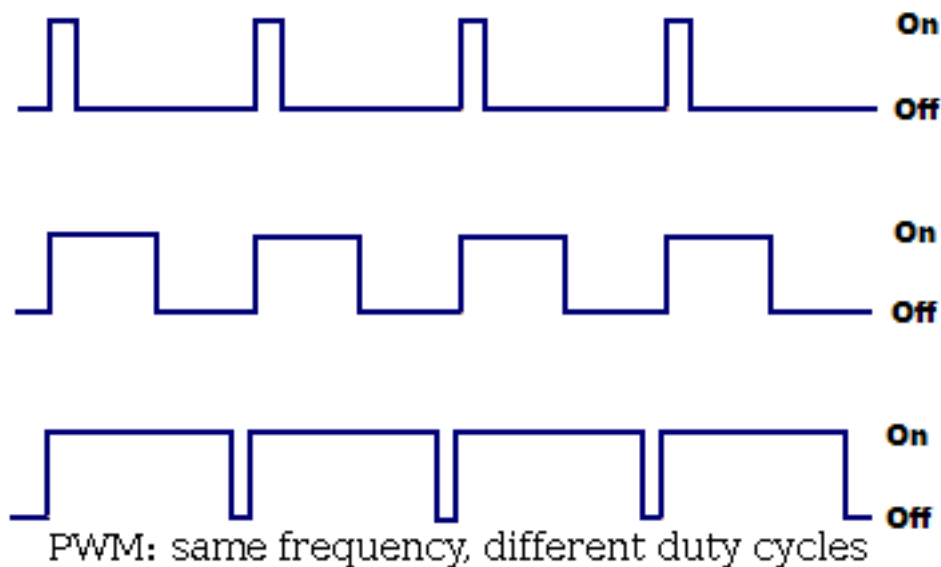
## PWM and its implementation by R-pi

Hi again. Today we use the PWM facility that R-Pi provides us.

For the sake of absolute beginners I will start by explaining what is PWM(Pulse Width Modulation).

Any digital device, like a microcontroller can only generate two levels on its output lines, HIGH=5v and LOW=0V. But what if we want to generate 2.5v or 3.1v or any voltage between 0-5 volt output ? For these kinds of requirement, instead of generating a constant DC voltage output we generate a square wave, which has high = 5V and Low = 0v

Now instead of varying the frequency of the square wave, we vary its duty cycle. Duty cycle is defined as the ratio of time wave remains on to the time period.



So you can see it is just the percentage of the total time the output was high. If the frequency of a wave is sufficiently high (say 500 Hz) and the duty cycle is 50% the output you get is half of 5v i.e. 2.5 volts. Thus if this output is connected to a motor(by means of suitable drivers) it will run at 50% of its full speed at 5v. The PWM technique utilizes this fact to generate any voltage between two extremes (for example between 0-12volts). The trick is to vary the duty cycle between 0-100% and get same percentage of input voltage to output.

Now, lets get to R-pi.

Unfortunately R-pi just provides us with one PWM pin, that is GPIO 1 corresponding to the 12<sup>th</sup> phsy pin.

The code I have written is very simple, and will slowly light up the LED increasing its brightness. There are two things I would like to mention, the PinMode here is set to PWM\_OUTPUT which will enable the GPIO 1 to give PWM output. Secondly, The pwmWrite function takes integer values from 0-1024 with 0 implying 0% duty cycle and 1024 giving 100% duty cycle. I hope the rest of the code is pretty clear.

```
#include<wiringPi.h>

main(){
    wiringPiSetup();
    pinMode(1,PWM_OUTPUT);
    while(1){
        for(int i=0;i<1024;i=i+10){
            pwmWrite(1,i);
            delay(100);}
        }
    return 0;
}
```

Finally, compile and execute the code and when you connect the LED to the 12<sup>th</sup> pin you should see the desired output.

Cheers!

Riddhish Bhalodia (sophomore EE, at IITB)