Cyber-Physical Systems

Analog Output

ICEN 553/453– Fall 2018

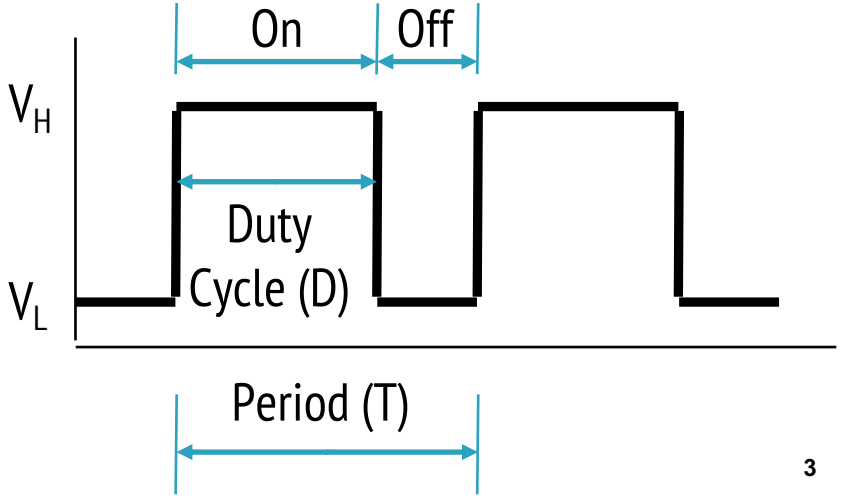
Prof. Dola Saha

Pulse Width Modulation (PWM)

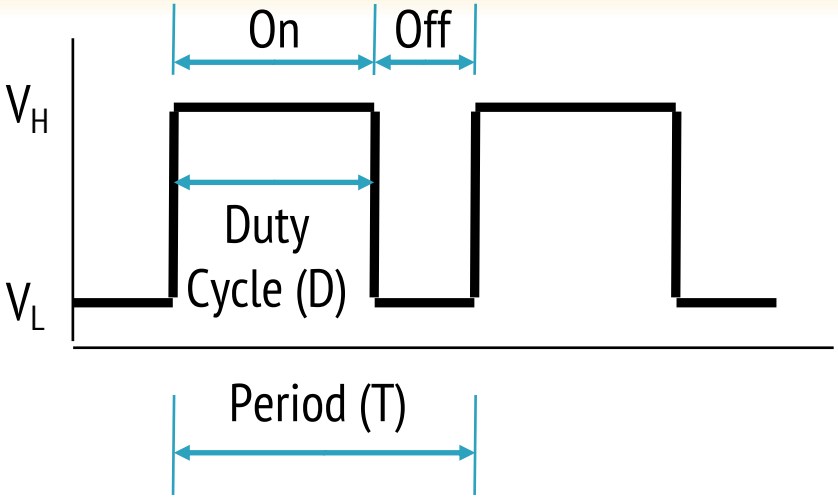
- Technique that conforms a signal width, generally pulses
- The general purpose is to control power delivery
- The on-off behavior changes the average power of signal
- Output signal alternates between on and off within a specified period.
- If signal toggles between on and off quicker than the load, then the load is not affected by the toggling

PWM – Duty Cycle

- A measure of the time the modulated signal is in its “high” state
- Generally recorded as the percentage of the signal period where the signal is considered on



Duty Cycle Formulation



Duty Cycle is determined by:

$$Duty\ Cycle = \frac{On\ Time}{Period} \times 100\%$$

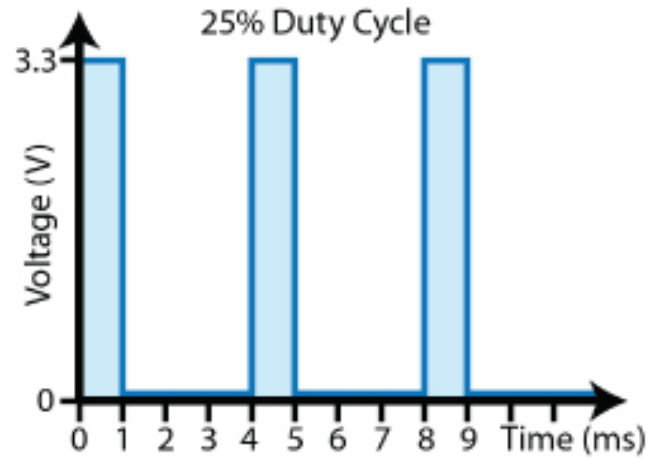
*Average value of a signal can be found as:

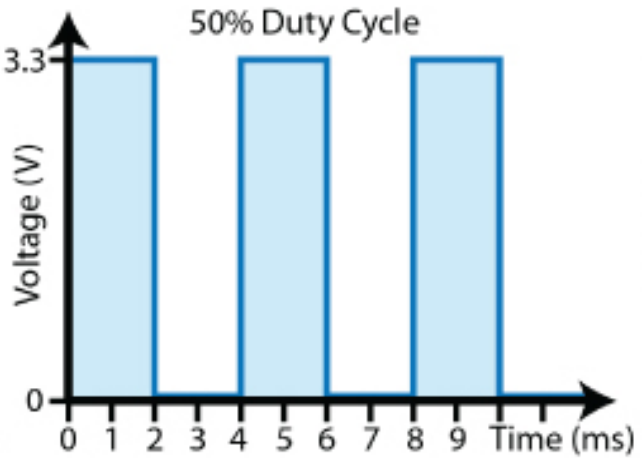
$$\bar{y} = \frac{1}{T}\int_0^T f(t)dt$$

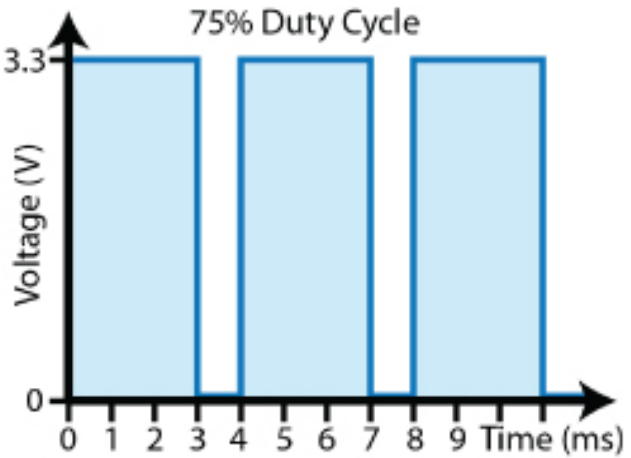
$$V_{avg} = D \cdot V_H + (1 - D) \cdot V_L$$

*In general analysis, V_L is taken as zero volts for simplicity.

PWM Duty Cycle

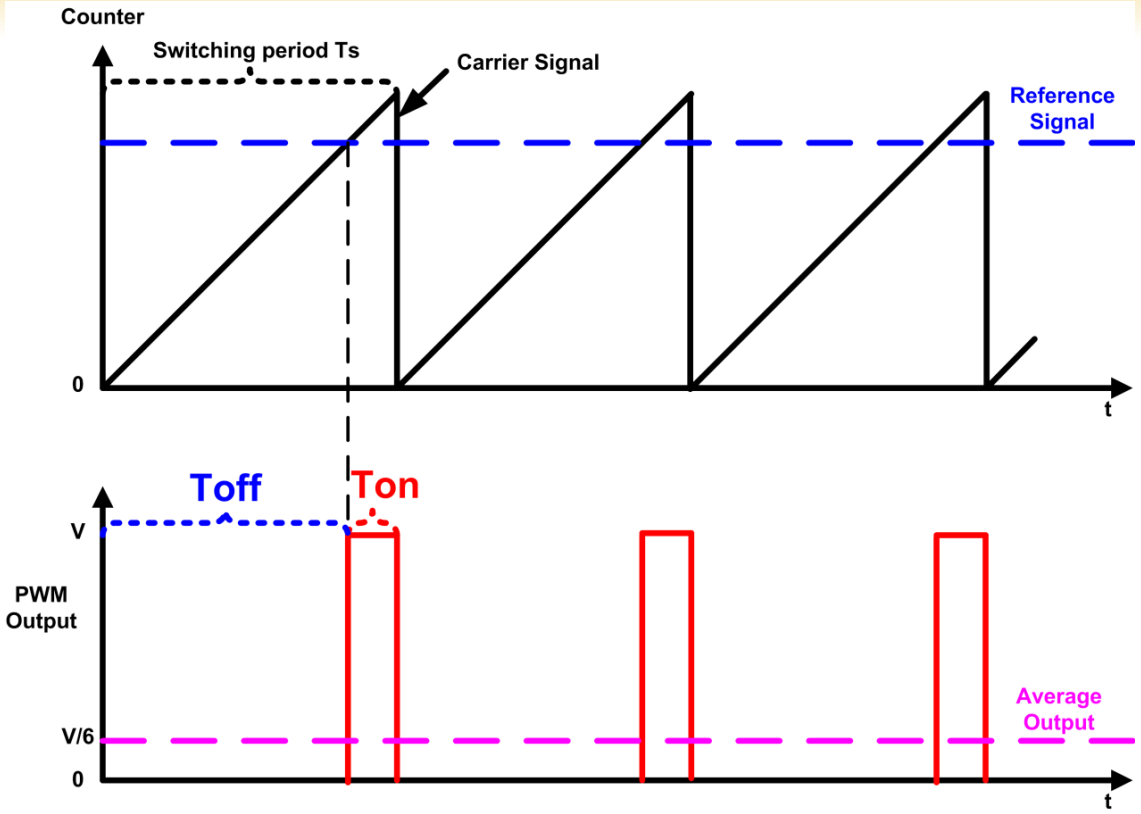






PWM Mode

- Counter counts up to the range provided
- When the counter value is higher than set value, output is high



PWM Duty Cycle Calculation

- The PWM device on the RPi is clocked at a fixed base-clock frequency of 19.2 MHz
- Integer divisor and range values are used to tailor the PWM frequency according to application requirements
- $f_{PWM} = 19.2MHz/(divisor \times range)$
- If f_{PWM} is 10KHz (0.01MHz), and range is 128,
  - $divisor = \frac{19.2MHz}{f_{PWM} \times range} = 15$

PWM0 and PWM1 Map

	PWM0	PWM1
GPIO 12	Alt Fun 0	-
GPIO 13	-	Alt Fun 0
GPIO 18	Alt Fun 5	-
GPIO 19	-	Alt Fun 5
GPIO 40	Alt Fun 0	-
GPIO 41	-	Alt Fun 0
GPIO 45	-	Alt Fun 0
GPIO 52	Alt Fun 1	-
GPIO 53	-	Alt Fun 1

9.6 Control and Status Registers

PWM Address Map			
Address Offset	**Register Name**	**Description**	**Size**
0x0	CTL	PWM Control	32
0x4	STA	PWM Status	32
0x8	DMAC	PWM DMA Configuration	32
0x10	RNG1	PWM Channel 1 Range	32
0x14	DAT1	PWM Channel 1 Data	32
0x18	FIF1	PWM FIFO Input	32
0x20	RNG2	PWM Channel 2 Range	32
0x24	DAT2	PWM Channel 2 Data	32

exploringPi/chp06/wiringPi/pwm.cpp

```
#include <iostream>
#include <wiringPi.h>
using namespace std;
#define PWM0        12                  // this is physical Pin 12
#define PWM1        33                  // only on the RPi B+/A+/2/3
int main() {                            // must be run as root
   wiringPiSetupPhys();                 // use the physical pin numbers
   pinMode(PWM0, PWM_OUTPUT);           // use the RPi PWM output
   pinMode(PWM1, PWM_OUTPUT);           // only on recent RPis
   // Setting PWM frequency to be 10kHz with a full range of 128 steps
   // PWM frequency = 19.2 MHz / (divisor * range)
   // 10000 = 19200000 / (divisor * 128) => divisor = 15.0 = 15
   pwmSetMode(PWM_MODE_MS);             // use a fixed frequency
   pwmSetRange(128);                    // range is 0-128
   pwmSetClock(15);                     // gives a precise 10kHz signal
   cout << "The PWM Output is enabled" << endl;
   pwmWrite(PWM0, 32);                  // duty cycle of 25% (32/128)
   pwmWrite(PWM1, 64);                  // duty cycle of 50% (64/128)
   return 0;                            // PWM output stays on after exit
}
```

Implement the Circuit

How do we fade an LED?

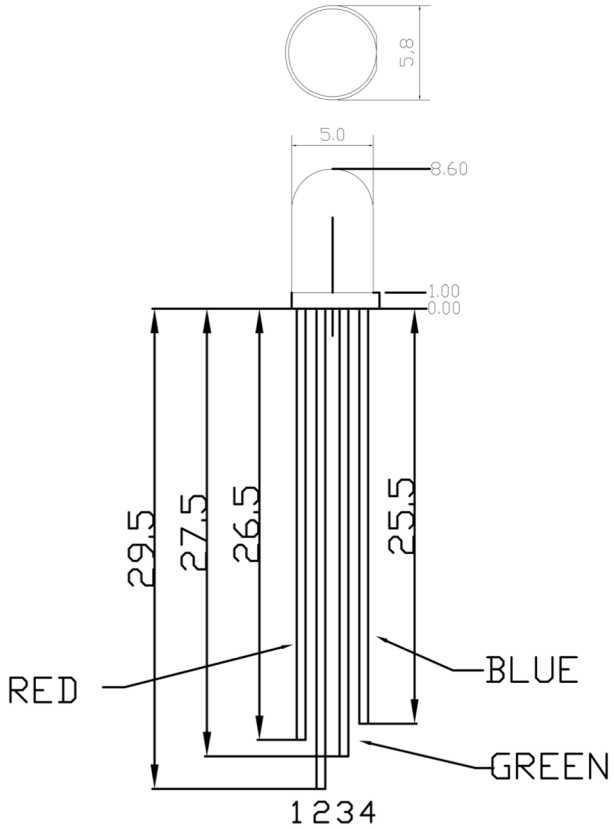
gpiozero Library

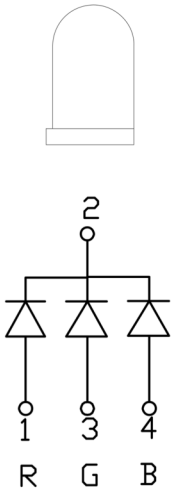
- PWM in effect

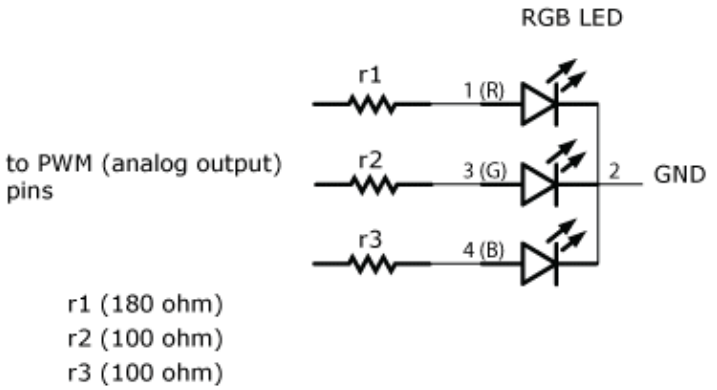
```
from gpiozero import PWMLED
from time import sleep

led = PWMLED(17)

while True:
    led.value = 0  # off
    sleep(1)
    led.value = 0.5  # half brightness
    sleep(1)
    led.value = 1  # full brightness
    sleep(1)
```

```
from gpiozero import PWMLED
from signal import pause

led = PWMLED(17)

led.pulse()

pause()
```

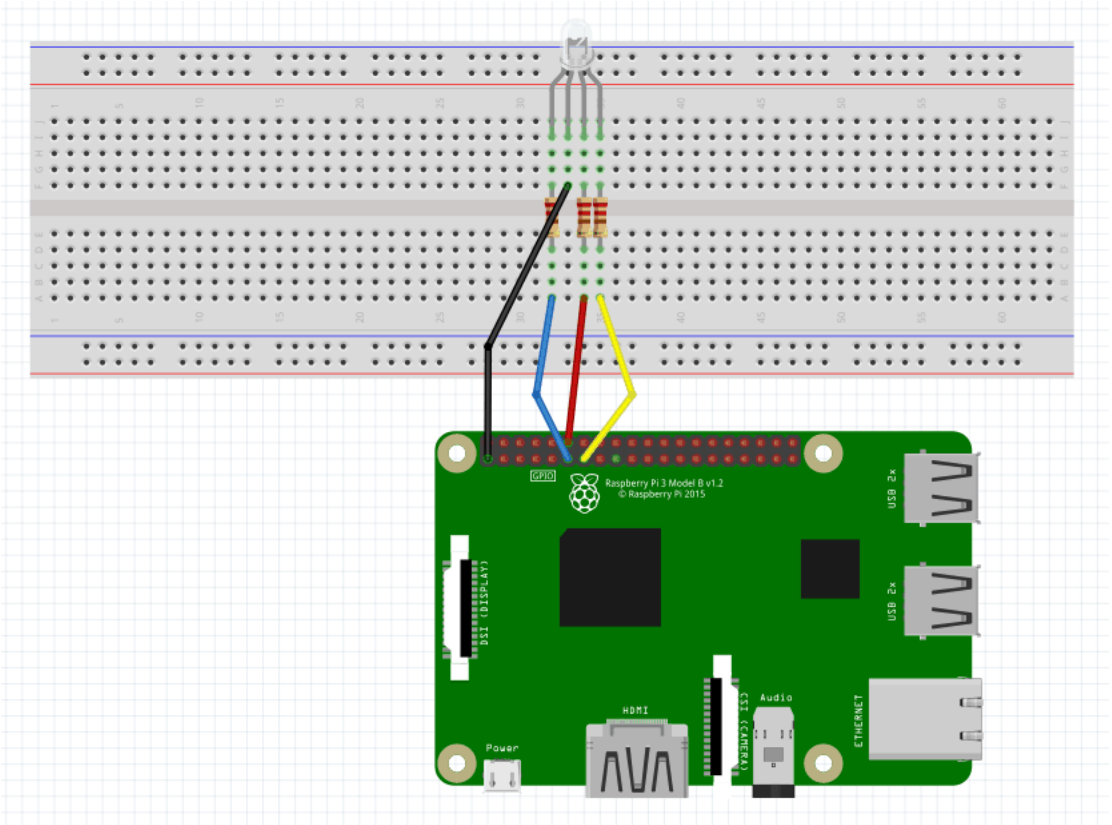
Use RGB LED for showing your own colors







Circuit


GPIO
Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015
USB 2x
USB 2x
DSI (DISPLAY)
HDMI
CSI (CAMERA)
Audio
ETHERNET
Power

Soft PWM Library in WiringPi (C/C++)

- https://projects.drogon.net/raspberry-pi/wiringpi/software-pwm-library/

- https://github.com/WiringPi/WiringPi/blob/master/wiringPi/softPwm.c

C Code for Soft PWM

Soft PWM

- PWM implemented in software
- https://sourceforge.net/p/raspberry-gpio-python/code/ci/default/tree/source/soft_pwm.c

Python Code for Soft PWM