Line Tracer 06

- PWM & DC Motor -

This lecture is based on

- DC Motors

1. PWM

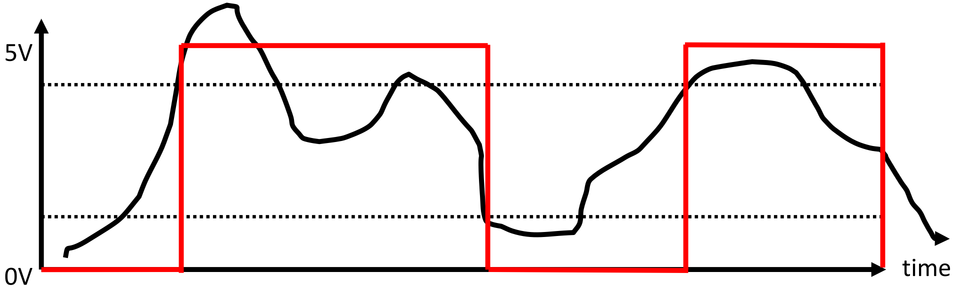
PWM Principle

We want to adjust the brightness of the LED

- 0V means 0% brightness
- 5V means 100% brightness
- 0.05V means 1% brightness?

-> No. Circuit would consider 0.05V as 0V

We need a way to convert a digital signal into an analog signal



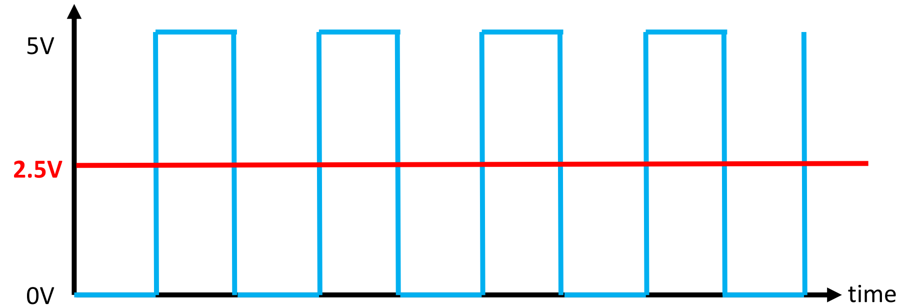
PWM Principle

What is PWM

- Pulse Width Modulation
- Digital to Analog Converter

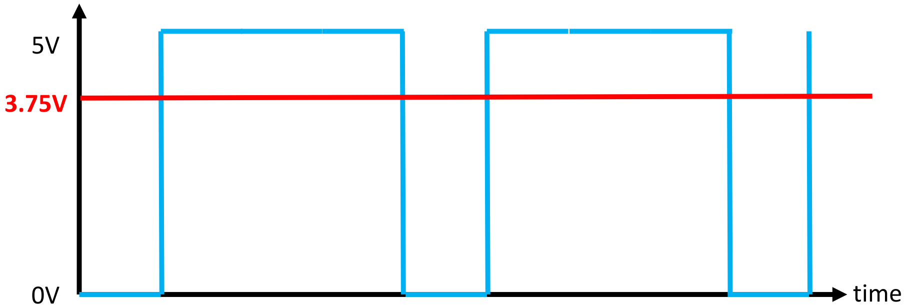
PWM Principle

50% duty cycle



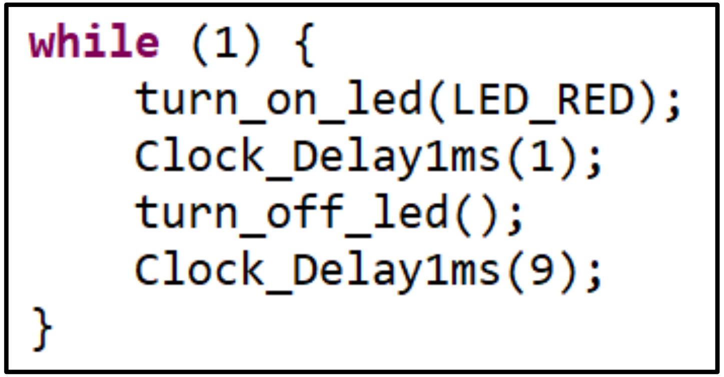
It behaves like 2.5v

75% duty cycle



It behaves like 3.75v

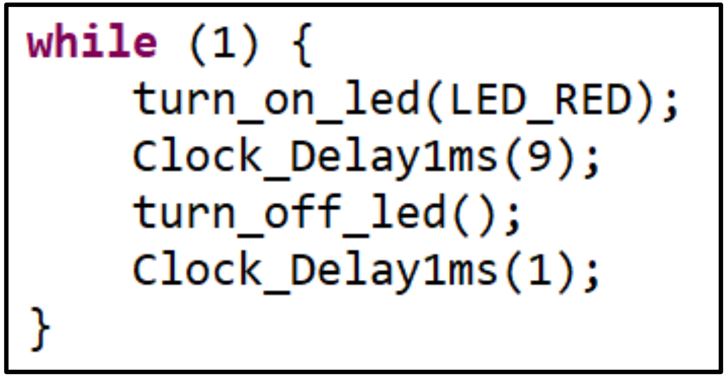
PWM Example 1



```
while (1) {
    turn_on_led(LED_RED);
    Clock_Delay1ms(1);
    turn_off_led();
    Clock_Delay1ms(9);
}
```

PWM Freq : 100Hz

Duty Cycle : 10%

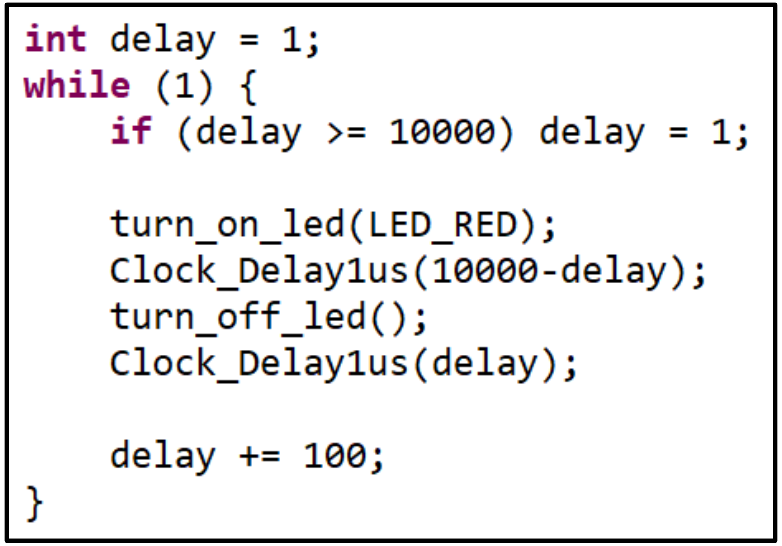


```
while (1) {
    turn_on_led(LED_RED);
    Clock_Delay1ms(9);
    turn_off_led();
    Clock_Delay1ms(1);
}
```

PWM Freq : 100Hz

Duty Cycle : 90%

PWM Example 2

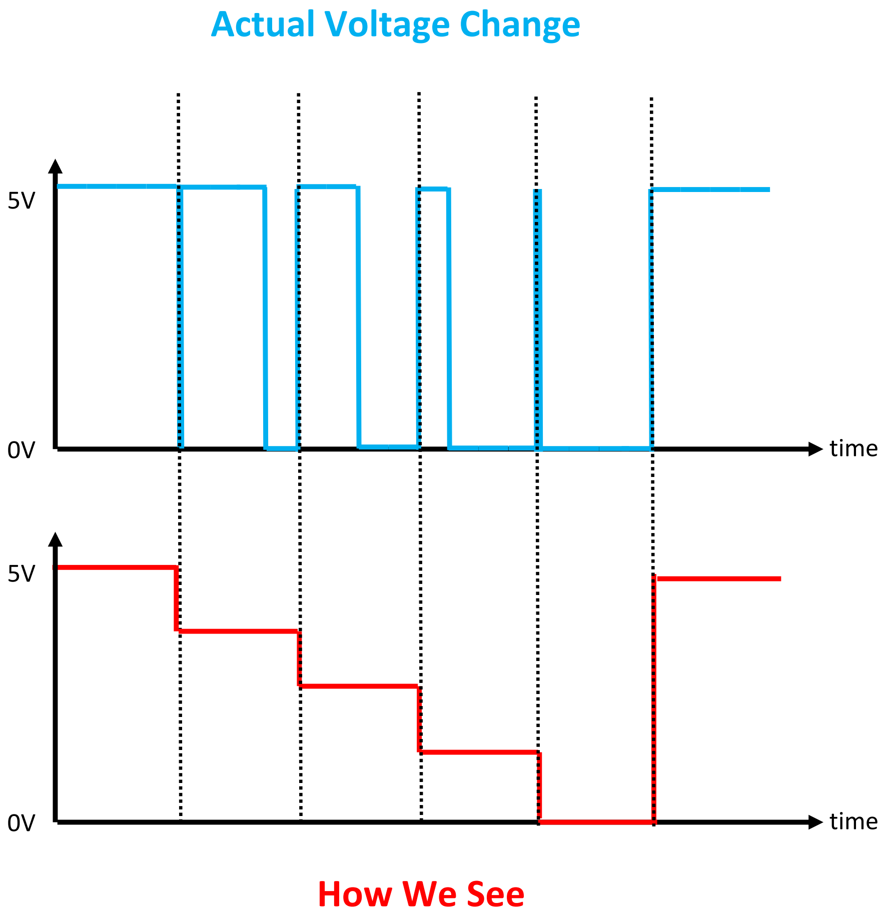


```
int delay = 1;
while (1) {
    if (delay >= 10000) delay = 1;

    turn_on_led(LED_RED);
    Clock_Delay1us(10000-delay);
    turn_off_led();
    Clock_Delay1us(delay);

    delay += 100;
}
```

100% brightness -> 0% brightness for every second



Actual Voltage Change

How We See

2. Motor

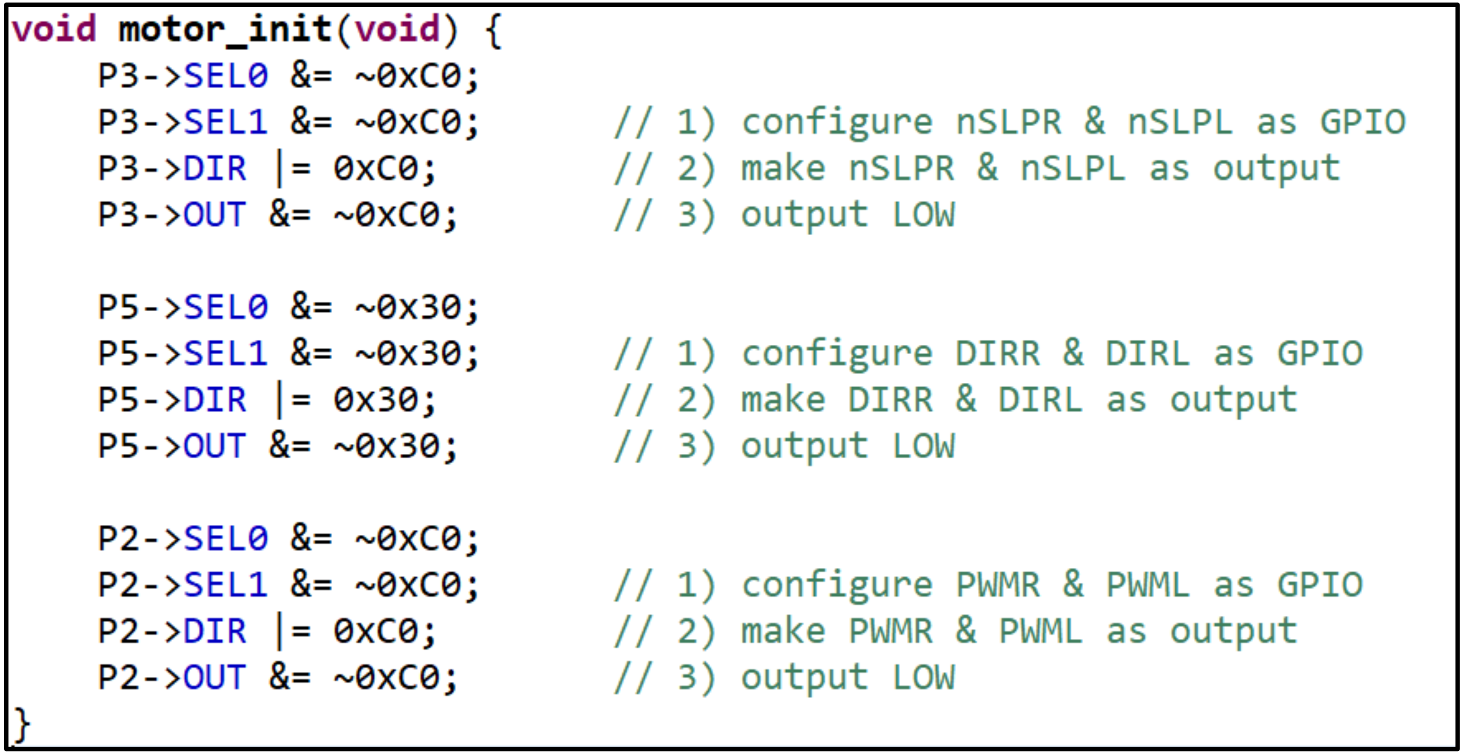
Motor Port Map

LaunchPad	TI-RSLK chassis board	DRV8838	Description
P5.5	DIRR	PH	Right Motor Direction
P3.6	nSLPR	nSLEEP	Right Motor Sleep
P2.6	PWMR	EN	Right Motor PWM
P5.4	DIRL	PH	Left Motor Direction
P3.7	nSLPL	nSLEEP	Left Motor Sleep
P2.7	PWML	EN	Left Motor PWM

PH	EN	nSleep	Motor
0	0	1	Stop
1	0	1	Stop
0	1	1	Forward
1	1	1	Back

To go forward, set nSleep=1, PH=0, and activate EN

Motor Initialization

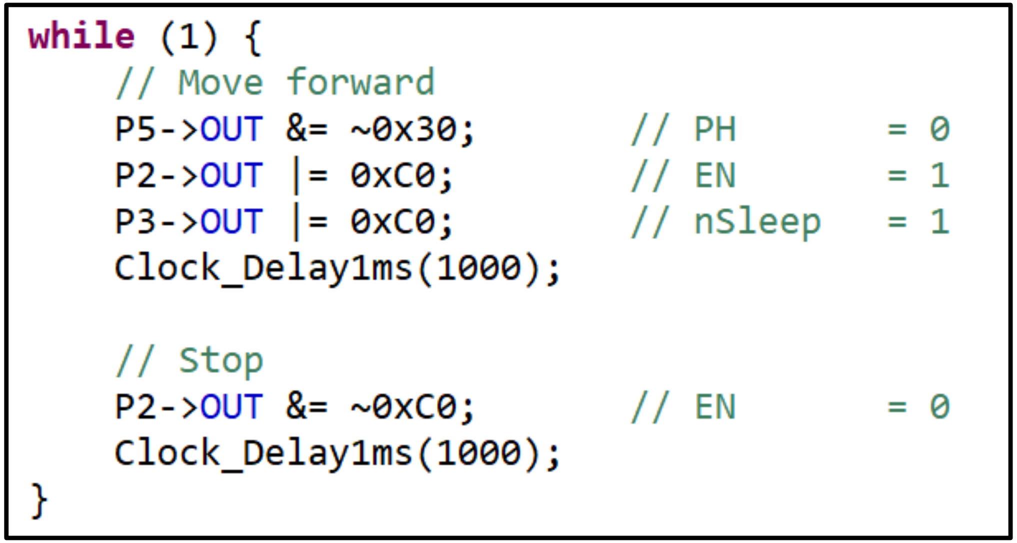


```c
void motor_init(void) {
    P3->SEL0 &= ~0xC0;
    P3->SEL1 &= ~0xC0;      // 1) configure nSLPR & nSLPL as GPIO
    P3->DIR |= 0xC0;        // 2) make nSLPR & nSLPL as output
    P3->OUT &= ~0xC0;       // 3) output LOW

    P5->SEL0 &= ~0x30;
    P5->SEL1 &= ~0x30;      // 1) configure DIRR & DIRL as GPIO
    P5->DIR |= 0x30;        // 2) make DIRR & DIRL as output
    P5->OUT &= ~0x30;       // 3) output LOW

    P2->SEL0 &= ~0xC0;
    P2->SEL1 &= ~0xC0;      // 1) configure PWMR & PWML as GPIO
    P2->DIR |= 0xC0;        // 2) make PWMR & PWML as output
    P2->OUT &= ~0xC0;       // 3) output LOW
}
```

Motor Example

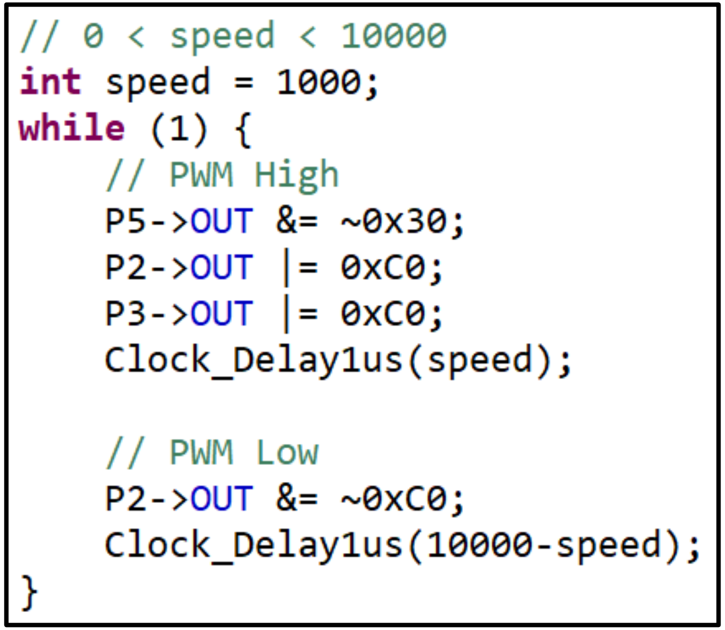


```c
while (1) {
    // Move forward
    P5->OUT &= ~0x30;        // PH       = 0
    P2->OUT |= 0xC0;         // EN       = 1
    P3->OUT |= 0xC0;         // nSleep   = 1
    Clock_Delay1ms(1000);

    // Stop
    P2->OUT &= ~0xC0;        // EN       = 0
    Clock_Delay1ms(1000);
}
```

You should turn on the power!

Motor Speed Control Example

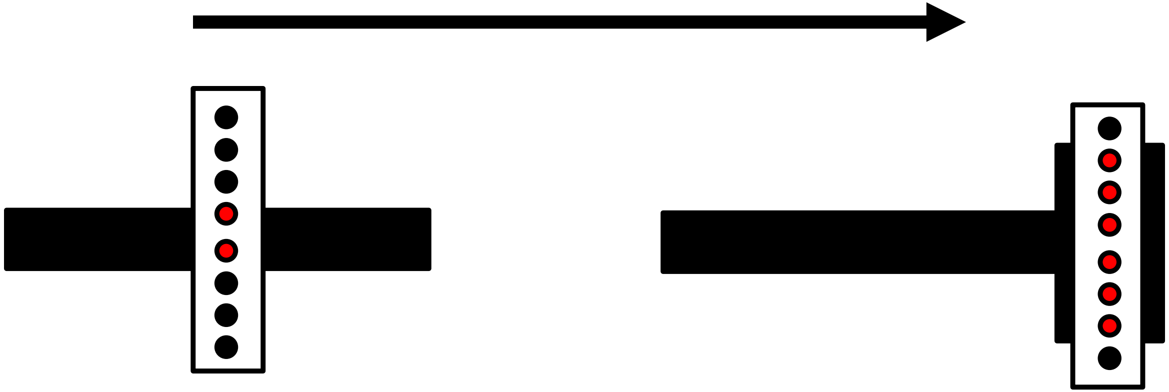


```c
// 0 < speed < 10000
int speed = 1000;
while (1) {
    // PWM High
    P5->OUT &= ~0x30;
    P2->OUT |= 0xC0;
    P3->OUT |= 0xC0;
    Clock_Delay1us(speed);

    // PWM Low
    P2->OUT &= ~0xC0;
    Clock_Delay1us(10000-speed);
}
```

3. Motor Activity

Stop at finish line(Not Assignment)



Move!

Don’t Move!