Breathing LED (PWM)

I. Learning Objectives

1. Learn PWM breathing LED principle.
2. Control the LED indicator to change from dark to bright, then from bright to dark by outputting a PWM signal, similar to human breathing.

II. PWM Introduction

PWM is the abbreviation of Pulse Width Modulation, which means pulse width modulation in Chinese. It is a very effective technology that uses the digital output of microprocessors to control analog circuits. It has become the most widely used control method in power electronics technology due to its advantages of simple control, flexibility, and good dynamic response. Its application fields include measurement, communication, power control and conversion, motor control, servo control, dimming, switching power supplies, and even some audio amplifiers. Therefore, learning PWM has very important practical significance.

In fact, we can also understand it this way: PWM is a method of digitally encoding analog signal levels. Through the use of high-resolution counters, the duty cycle of square waves is modulated to encode the level of a specific analog signal. PWM signals are still digital because at any given moment, the full-scale DC power supply is either completely ON or completely OFF. Voltage or current sources are applied to analog loads as a repetitive pulse sequence of ON or OFF. When ON, it means DC power is applied to the load; when OFF, it means the power supply is disconnected. As long as the bandwidth is sufficient, any analog value can be encoded using PWM.

PWM Basic Parameters

PWM is pulse width modulation and has two very important parameters: frequency and duty cycle.

- **Frequency**: The frequency of PWM is the reciprocal of the entire period, that is, the number of repetitions of the PWM signal per unit time. It represents how many signal cycles are completed per second. Frequency determines the rate of change of the PWM signal. The higher the frequency, the shorter the signal period and the more changes occur.
- **Duty Cycle**: Duty cycle refers to the proportion of high level within one period. Usually expressed as a percentage. It reflects the "width" of the PWM signal, that is, the ratio of the duration of the high level part to the entire period time. The duty cycle determines the average output voltage of the signal. The larger the duty cycle, the higher the average output voltage.

III. Hardware Setup

Previously we introduced that the MSPM0G series has a total of 7 timers, which can be divided into 2 types: General Purpose Timers (TIMG) and Advanced Control Timers (TIMA). The PWM function is implemented based on timers. From the user manual, we can understand that MSPM0G3507 has 7 timers. Which timer has PWM function needs to refer to the pin description in the data sheet. Each PWM channel corresponds to a pin of the microcontroller. This pin is not uniquely fixed. One or two pins may correspond to the same channel. For example, TIMG_C0 corresponds to PA5 and PA12, which means both PA5 and PA12 pins can be configured as channel 0 of the timer. We can choose either one for configuration when using it.

10	PA5	HFXIN	TIMG8_C0 *[2]* / SPI0_PICO *[3]* / TIMA_FAL1 *[4]* / TIMG0_C0 *[5]* / TIMG6_C0 *[6]* / FCC_IN *[7]*	45	11	9	12	

34	PA12		UART3_CTS *[2]* / SPI0_SCK *[3]* / TIMG0_C0 *[4]* / CAN_TX *[5]* / TIMA0_C3 *[6]* / FCC_IN *[7]*	5	27	16	–	

This case uses the PB2 pin of LED1, configured as channel 0 of timer 6, to generate a PWM signal to control the light's on/off state.

15	PB2		UART3_TX *[2]* / UART2_CTS *[3]* / I2C1_SCL *[4]* / TIMA0_C3 *[5]* / UART1_CTS *[6]* / TIMG6_C0 *[7]* / TIMA1_C0 *[8]*	50	14	-	-	

PWM has two modes. One is edge-aligned PWM, which refers to the down-counting mode used by the timer. The output channel will output high level when the timer counting starts, **when the count value is the same as the comparison value of the output channel (CC0 and CCP0, CC1 and CCP1 in the figure),** the output channel will output low level, until the timer's count value counts to 0, completing one cycle of pulse output.

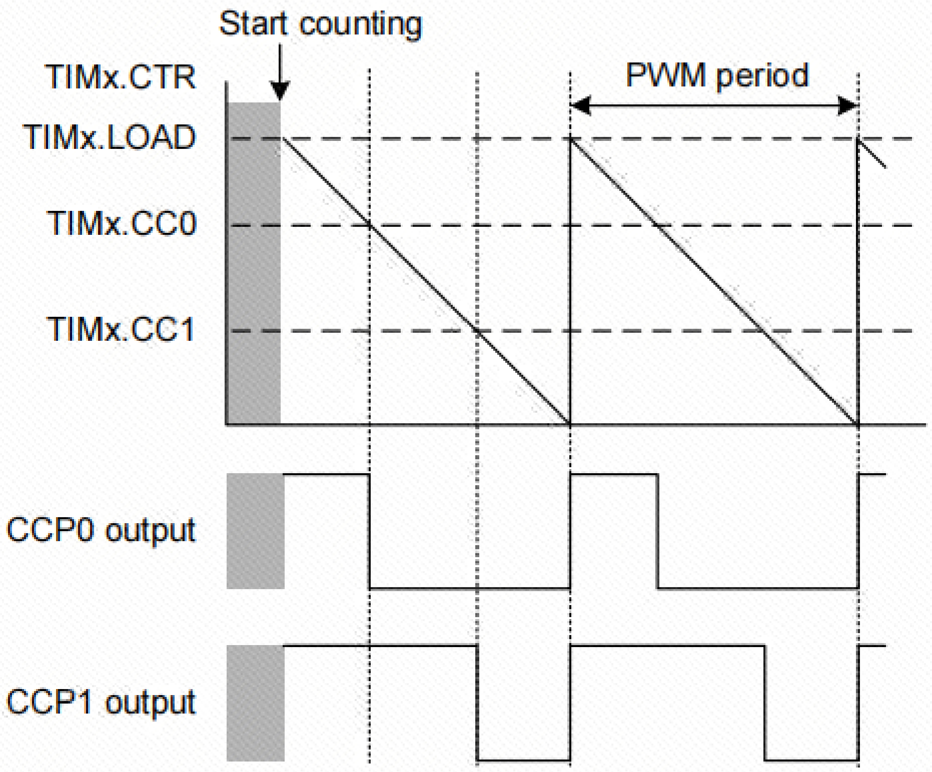


Figure 25-28. Edge-Aligned PWM Signals in Down-Counting Mode

Another mode is center-aligned mode, which refers to the up-down counting mode of the timer. The output channel will output low level when the timer starts, and will toggle the level once each time the timer counts up and down twice to reach the comparison value of the output channel.

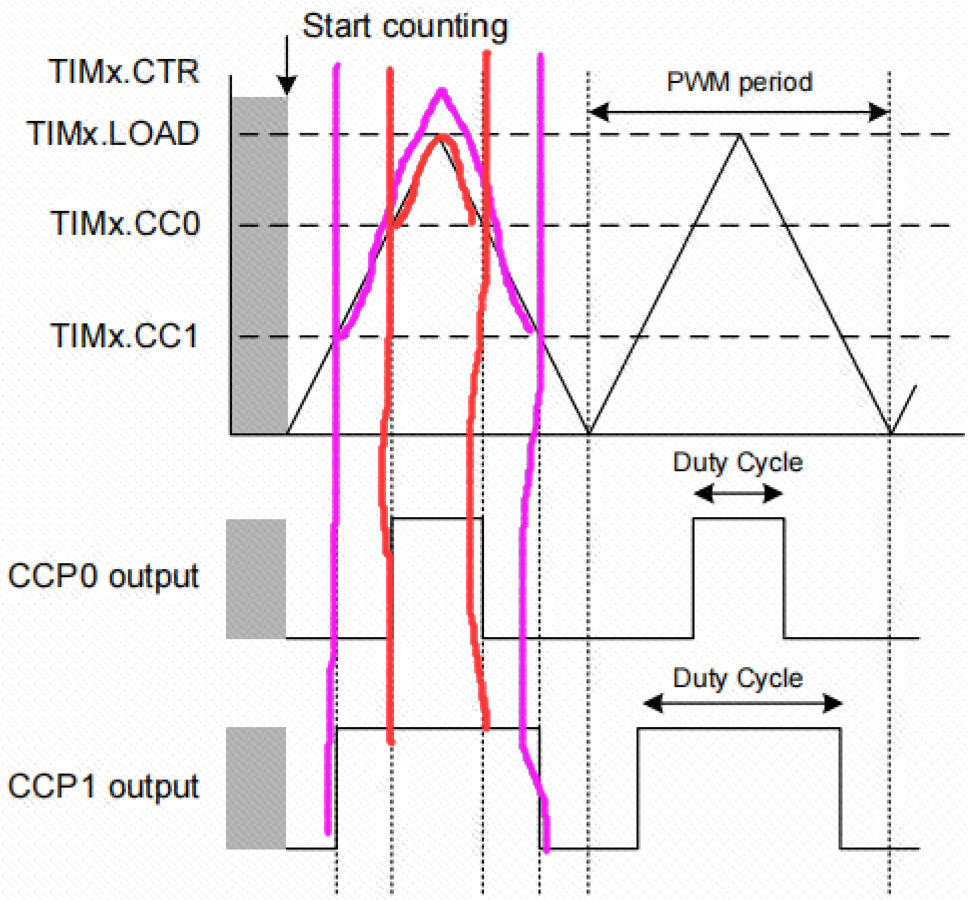


Figure 25-30. Center-Aligned PWM

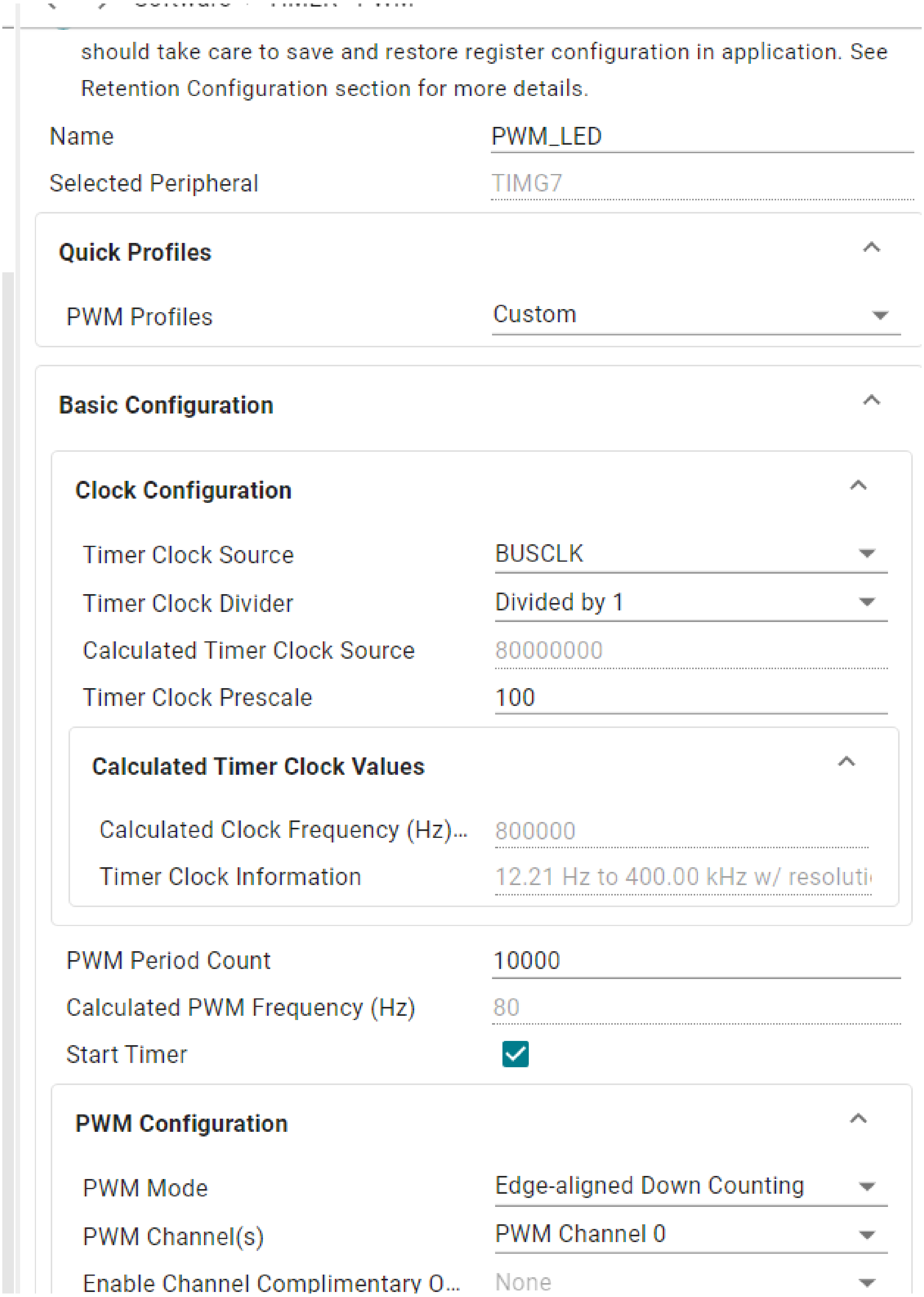
IV. Experiment Steps

By driving PWM signals to achieve breathing LED effect, by changing the duty cycle of PWM signals, the current flowing through the LED can be adjusted, thereby achieving gradual changes in LED brightness. Generally, the human eye does not have obvious flicker perception for refresh frequencies above 80Hz, because flicker cannot be perceived when the frequency is high. The larger the duty cycle, the brighter the LED; the smaller the duty cycle, the darker the LED. Therefore, with fixed frequency, LED brightness can be controlled by adjusting the duty cycle, thereby achieving breathing LED effect. This case will use the onboard LED (red) to achieve breathing LED effect, gradually brightening and then gradually dimming.

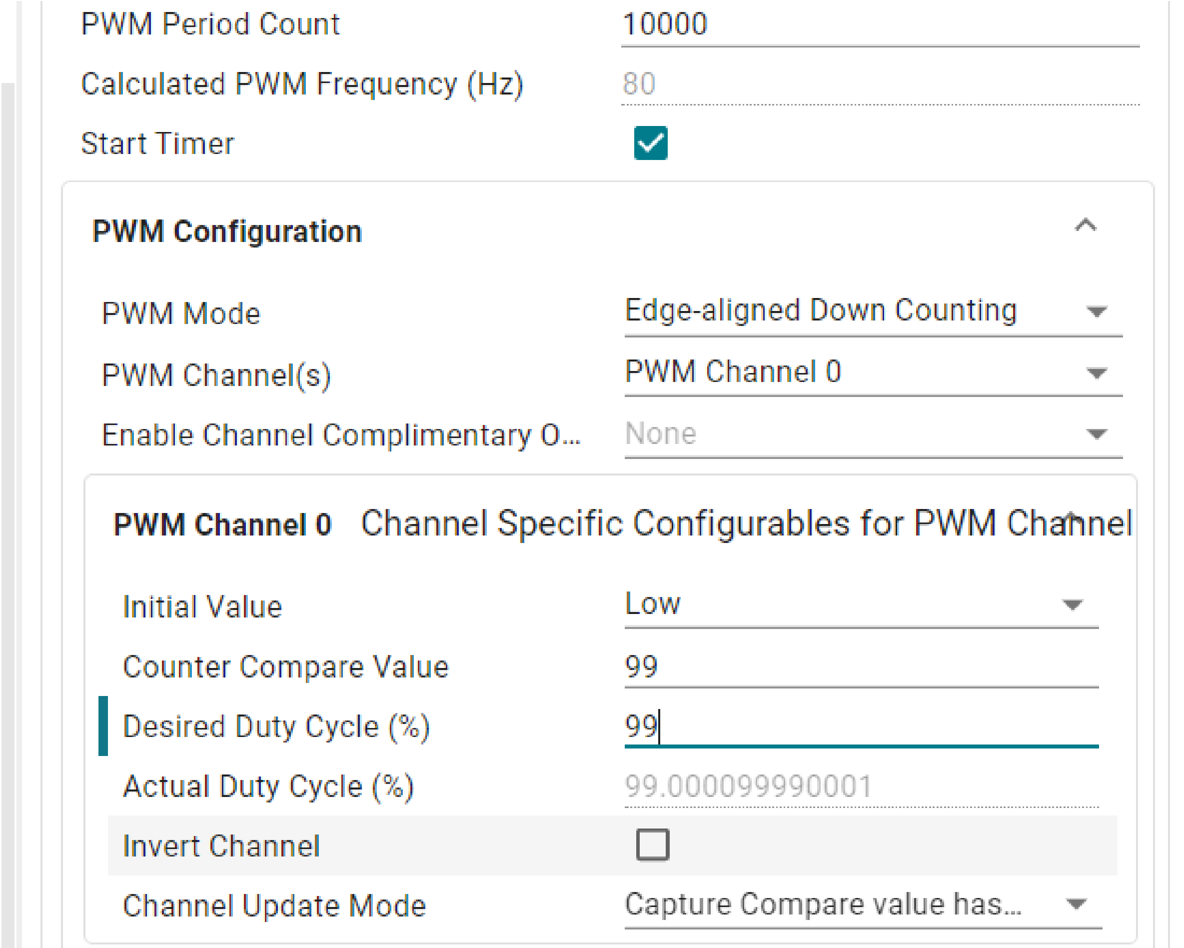
This project can be built based on our LED control

First, we open the sysconfig tool to add timer peripheral configuration as follows

Here to connect with the effect, our PWM peripheral name is named PWM_LED, you can name it yourself



We configure the clock source as bus clock, no division, prescaler coefficient as 100, PWM reload value set to 10000. With this configuration, we get an 80Hz square wave signal



Name: Custom name. Set to `PWM_LED` here.

PWM Profiles: Configure PWM template, select the default one.

Timer Clock Source: Timer clock source, select `BUSCLK` (bus clock), default frequency is 32MHz, we are at 80MHz here.

Timer Clock Divider: Timer clock divider set to **80 division**.

Calculated Timer Clock Source: Display the clock frequency after division.

Timer Clock Prescale: Timer prescale, we configured TIMG6, a 16-bit timer that supports prescaling. No prescale is set here.

Calculated Clock Frequency (Hz): Display the frequency after division and prescaling.

PWM Period Count: PWM period count value, we set it to 10000 here.

Calculated PWM Frequency(Hz): Software automatically calculates PWM frequency, displayed as 80.

Start Timer: Whether to start the timer. Checked here, power on to start timer PWM output.

PWM Mode: PWM mode. Configured as `Edge-aligned Down Counting` edge-aligned mode and down-counting mode.

PWM Channel(s): PWM channels. Configured to only enable channel 0.

Enable Channel Complimentary Output: Whether to enable channel complementary output. Not enabled here.

Initial Value: Default output state of PWM channel. Configured as default low level output.

Counter Compare Value: PWM comparison value. Configured as 1000, that is, PWM duty cycle setting range is 0-999.

Desired Duty cycle (%): PWM duty cycle. Default duty cycle 0 is used here.

Actual Duty cycle (%): Display the actual duty cycle automatically calculated by software.

Invert Channel: Whether to enable inverted output. Not enabled here.

Channel Update Mode: Channel update mode. Configured as `Capture Compare value has immediate effect` immediate update when comparison value is reached.

V. Program Analysis

empty.c

```
#include "ti_msp_dl_config.h"
#include "uart0.h"
#include "delay.h"

int main(void)
{
    //System initialization
     SYSCFG_DL_init();

    while (1)
    {

                for(int i=10000; i>=0;i-=10)
                {
                    //Function to set comparison value

DL_Timer_setCaptureCompareValue(PWM_LED_INST,i,GPIO_PWM_LED_C0_IDX);
                    delay_ms(1);
                }

                for(int i=0; i<=10000;i+=10)
                {
                    //Function to set comparison value

DL_Timer_setCaptureCompareValue(PWM_LED_INST,i,GPIO_PWM_LED_C0_IDX);
                    delay_ms(1);
                }
    }

}
```

VI. Experiment Phenomenon

After the program download is completed, you can see the LED gradually brighten and dim, showing a breathing LED effect.