

PAN1080 PWM Sample Application Note

PAN-CLT-VER-B0, Rev 0.1

PANCHIP

PanchipMicroelectronics

www.panchip.com

修订历史

版本	修订日期	描述
V0.1	2021-10-17	初始版本创建

PANCHIP

目录

第 1 章 例程演示内容	4
1.1 测试内容	4
1.2 环境准备	4
1.2.1 软件环境	4
1.2.1.1 待测代码	4
1.2.1.2 软件工具	4
1.2.2 硬件环境	4
第 2 章 例程演示流程	6
2.1 环境配置	6
2.1.1 测试程序编译烧录	6
2.1.2 硬件接线	6
2.2 PWM 工作流程	6
2.3 测试程序初始化	6
2.4 基本功能验证	7
2.4.1 PWM 所有寄存器默认状态	7
2.4.2 计数模式	7
2.4.2.1 单次计数模式	7
2.4.2.2 带双缓存的自动加载模式	7
2.4.2.3 带中心加载的自动加载模式	8
2.4.3 计数形式与中断使用	9
2.4.3.1 边缘对齐（向下计数）形式	9
2.4.3.2 中心对齐（上下计数）形式	11
2.4.3.3 精确中心对齐（上下计数）形式	12
2.4.3.4 非对称计数的中心对齐形式	14
2.4.4 操作模式	16
2.4.4.1 独立通道模式	16
2.4.4.2 互补通道模式	17
2.4.4.3 带死区插入的互补通道模式	18
2.4.4.4 同步通道模式	19
2.4.4.5 分组通道模式	20
2.4.5 极性控制	21
2.4.6 简单 API 接口演示	23
2.4.6.1 演示 Demo 1	24
2.4.6.2 演示 Demo 2	26
第 3 章 使用注意事项	29

第1章 例程演示内容

1.1 测试内容

1. 寄存器默认值 (Register default value)
2. 计数模式 (PWM Counting Modes)
 - a) 单次计数模式 (One-Shot Mode)
 - b) 带双缓存的自动加载模式 (Auto-Reload Mode with Double Buffering)
 - c) 带中心加载的自动加载模式 (Auto-Reload Mode with Center Loading)
3. 计数形式与中断使用 (PWM Counting Types and Interrupts)
 - a) 边缘对齐 (向下计数) 形式 (Edge-Aligned PWM)
 - b) 中心对齐 (上下计数) 形式 (Center-Aligned PWM)
 - c) 精确中心对齐 (上下计数) 形式 (Precise Center-Aligned PWM)
 - d) 非对称计数的中心对齐形式 (Center-Aligned PWM with Asymmetric Mode)
4. 操作模式 (Operating Modes)
 - a) 独立通道模式 (Independent Mode)
 - b) 互补通道模式 (Complementary Mode)
 - c) 带死区插入的互补通道模式 (Complementary Mode with Dead-Time Insertion)
 - d) 同步通道模式 (Synchronized Mode)
 - e) 分组通道模式 (Grouping Mode)
5. 极性控制 (Polarity Control)
6. 简单 API 接口演示 (PWM Simple APIs Demo)

1.2 环境准备

1.2.1 软件环境

1.2.1.1 待测代码

测试工程文件:

<PAN1080-DK>\03_MCU\mcu_samples\PWM\keil\PWM.uvprojx

测试源文件目录:

<PAN1080-DK>\03_MCU\mcu_samples\PWM\src

1.2.1.2 软件工具

- 1、SecureCRT (用于显示 PC 与 EVB 的交互过程, 打印 log 等)
- 2、KingstVIS (逻辑分析仪 LA1010 配套软件)

1.2.2 硬件环境

- 1、PAN1080 EVB 1 块
 - a) UART0 (测试交互接口, TX: P00, RX: P01, 波特率: 921600)

- b) PWM0 (待测模块, 共 7 个通道)
 - i. 待测通道, 所有例程均用到, 默认 PWM1_CH0, Output Pin: P04
 - ii. 辅助测试通道, 在测试互补通道模式、同步通道模式、分组通道模式等例程的时候用到, 默认 PWM1_CH1, Output Pin: P05
 - iii. 所有通道 (Channel0~7), 在演示分组通道模式的时候用到, Output Pin: P04、P05、P06、P07、P22、P23、P30、P31
 - c) SWD (用来调试和烧录程序, SWDCLK: P46, SWDIO: P47)
- 2、逻辑分析仪 (波形抓取工具)
- 3、JLink (SWD 调试与烧录工具)

第2章 例程演示流程

2.1 环境配置

2.1.1 测试程序编译烧录

打开测试工程，确保可以编译通过。编译之前，可以在 `pwm_common.h` 中配置大部分例程的待测 PWM1 通道为 `PWM_CH0 ~ PWM_CH7`。本测试工程默认将 `PWM1_CH0` 作为待测 PWM Channel (Target PWM Channel)，将 `PWM1_CH1` 作为辅助测试 Channel (Auxiliary PWM Channel)。

2.1.2 硬件接线

接线方面，不同的例程连线稍有不同：

1. 将 EVB 板的 RX0 和 TX0 进行跳线，然后连接 USB->UART 到 PC。

2. 对于前三个例程（寄存器默认值、计数模式、计数形式与中断使用），需要将 P04 (Target PWM Channel)、P02 (CMPDAT/PERIOD Changing Indicator)、P03 (ZIF Signal Indicator)、P32 (CMPUIF Signal Indicator)、P33 (PIF Signal Indicator)、P10 (CMPDIF Signal Indicator) 等 6 个引脚分别接入逻辑分析仪的通道 0~通道 5。

3. 对于中间两个例程（操作模式、极性控制），需要将 P04、P05、P06、P07、P22、P23、P30、P31 (PWM1 Channel 0~7) 等 8 个引脚分别接入逻辑分析仪的通道 0~通道 7。

4. 对于最后一个例程（简单 API 接口演示），需要将 P22 (PWM1_CH4)、P23 (PWM1_CH5)、P30 (PWM1_CH6) 等 3 个引脚分别接入逻辑分析仪的通道 0~通道 2；将 P02（用于辅助观察调整波形占空比的效果）引脚接入逻辑分析仪的通道 3。

2.2 PWM 工作流程

参考 User Manual 文档。

2.3 测试程序初始化

硬件连线完成并烧录测试程序后，EVB 上电，观察串口是否正常打印测试主菜单。

CPU @ 64000000Hz

```

PN108D PWM Sample Code

Press key to start specific testcase:

Input '0'   Testcase 0: Register Default value check.
Input '1'   Testcase 1: PWM Counting Modes.
Input '2'   Testcase 2: PWM Counting Types and Interrupts.
Input '3'   Testcase 3: PWM Operating Modes.
Input '4'   Testcase 4: Polarity Control.
Input '5'   Testcase 5: Simple APIs Demo.
    
```

2.4 基本功能验证

2.4.1 PWM 所有寄存器默认状态

在主菜单下，输入 ‘0’ 命令 打印所有寄存器默认值：

测试目的：

检查所有 PWM 相关寄存器复位 Default 值状态。

测试预期：

寄存器默认值应和 PAN1080 Datasheet 上 PWM 模块默认值一致（全 0）。

测试现象：

```
0
pwm default value check ok
PWM Test OK, Success case: 0
```

测试分析：

参考芯片手册对比寄存器信息，发现是完全一致的，符合预期。

2.4.2 计数模式

在主菜单下，输入 ‘1’ 命令 进入 Subcase 菜单：

```
+-----+
| Press key to test specific function: |
|                                         |
| Input 'A'   One-shot Mode.             |
| Input 'B'   Auto-Reload Mode with PERIOD and CMPDAT value |
|              changing (Double Buffering Feature involved). |
| Input 'C'   Auto-Reload Mode with PERIOD and CMPDAT value |
|              changing (Center Loading Operation Enabled). |
| Press ESC key to back to the top level case list.         |
+-----+
```

2.4.2.1 单次计数模式

PAN1080 不支持此模式！

2.4.2.2 带双缓存的自动加载模式

测试目的：

验证带双缓存的自动加载模式（Auto-Reload Mode with Double Buffering）工作是否正常。

测试预期：

PWM 能够连续输出正确的方波，并且在此过程中修改 CMPDAT 和 PERIOD 值后，从下个周期开始，方波按照新设定的值改变周期和占空比后继续输出，直到软件停止输出。

测试现象：

先正确连接 EVB 与逻辑分析仪，然后输入 ‘B’ 命令，可以看到 Log 打印待测 PWM Channel 的初始和修改后的 PERIOD 和 CMPDAT 值。

```
PWM start, auto-reload mode, PERIOD: 6399, CMPDAT: 1919.
PERIOD and CMPDAT changed, new PERIOD: 12798, new CMPDAT: 7676.
TGT_PWM stopped.

+-----+
| Press key to test specific function: |
|                                     |
| Input 'A'   One-Shot Mode.         |
| Input 'B'   Auto-Reload Mode with  |
|              PERIOD and CMPDAT    |
|              value changing        |
|              (Double Buffering     |
|              Feature involved).    |
| Input 'C'   Auto-Reload Mode with  |
|              PERIOD and CMPDAT    |
|              value changing        |
|              (Center Loading      |
|              Operation Enabled).   |
| Press ESC key to back to the top  |
| level case list.                   |
+-----+
```

再看 LA 波形，发现 PWM 输出波形的周期和占空比在 500us 左右的时候有变化，并且在变化之前最后一个周期的高电平期间，P02 (CMPDAT/PERIOD) 引脚有一个脉冲出现。



测试分析:

Auto-Reload 模式下，波形可以连续输出，并且由于双缓存 (Double Buffering) 机制的存在，在输出波形的时候可以动态修改波形周期和占空比，生效时间是从修改完参数的下个周期开启。而看 LA 的波形可以发现，P04 的短脉冲 (指示 PERIOD 和 CMPDAT 被修改的时间) 正好发生在波形周期和占空比改变的前一个周期的时间内，符合预期。

2.4.2.3 带中心加载的自动加载模式

测试目的:

验证带中心加载的自动加载模式 (Auto-Reload Mode with Center Loading) 工作是否正常。

测试预期:

PWM 能够连续输出正确的方波，并且在此过程中修改 CMPDAT 值后，从当前周期 period 半周期开始，方波按照新设定的值改变占空比后继续输出，直到软件停止输出。

初始默认 PWM 周期 5000Hz，高电平 0.3ms，低电平 0.7ms，变化 $cmpdata = cmpdata * 2/3$ ，那么变化后的周期高电平时间为： $0.3/2 + (1 - 0.7 * 2/3) / 2 = 0.41ms$

测试现象:

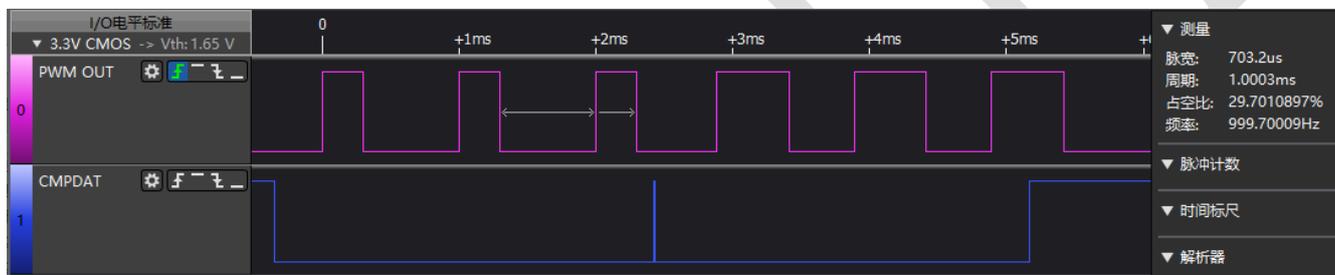
先正确连接 EVB 与逻辑分析仪，然后输入 ‘C’ 命令，可以看到 Log 打印待测 PWM Channel

的初始和修改后的 PERIOD 和 CMPDAT 值。

```
PWM start, auto-reload mode, PERIOD: 15999, CMPDAT: 11199.
PERIOD and CMPDAT changed, new PERIOD: 15999, new CMPDAT: 7466.
TGT_PWM stopped.

+-----+
| Press key to test specific function: |
|                                       |
| Input 'A'   One-Shot Mode.           |
| Input 'B'   Auto-Reload Mode with PERIOD and CMPDAT value |
|              changing (Double Buffering Feature involved). |
| Input 'C'   Auto-Reload Mode with PERIOD and CMPDAT value |
|              changing (Center Loading Operation Enabled). |
| Press ESC key to back to the top level case list.         |
+-----+
```

再看 LA 波形，发现 PWM 输出波形的周期和占空比在 2.5ms 左右的时候有变化，并且在变化之前最后一个周期前的高电平期间，P02（CMPDAT/PERIOD）引脚有一个脉冲出现。



测试分析:

Auto-Reload 模式下，波形可以连续输出，并且由于自动加载机制的存在，在输出波形的的时候可以动态修改波形占空比，生效时间是从修改完参数的当前周期一半后开启。而看 LA 的波形可以发现，P02（CMPDAT/PERIOD）引脚的短脉冲后高电平时间为 530us，大致符合预期。

备注：变化的周期 cmpdata 值设置必须在到达 period 值之前设置，否则写一个周期才生效

2.4.3 计数形式与中断使用

在主菜单下，输入 ‘2’ 命令 进入 Subcase 菜单：

```
+-----+
| Press key to test specific function: |
|                                       |
| Input 'A'   Edge-Aligned PWM.       |
| Input 'B'   Center-Aligned PWM.     |
| Input 'C'   Precise Center-Aligned  |
|              PWM.                   |
| Input 'D'   Center-Aligned PWM with |
|              Asymmetric Mode.       |
| Press ESC key to back to the top    |
|              level case list.       |
+-----+
```

2.4.3.1 边缘对齐（向下计数）形式

测试目的:

验证边缘对齐（向下计数）形式（Edge-Aligned PWM）及相关中断工作是否正常。

测试预期:

PWM 从 PERIOD 值开始递减计数，当计数值与 CMPDAT 相同时，输出电平翻转（由低电平变为高电平）同时触发 CMP Down 中断；计数值继续递减至 0，输出电平再次翻转（由高电平变为低电平）同时触发 Zero 中断，如此往复。

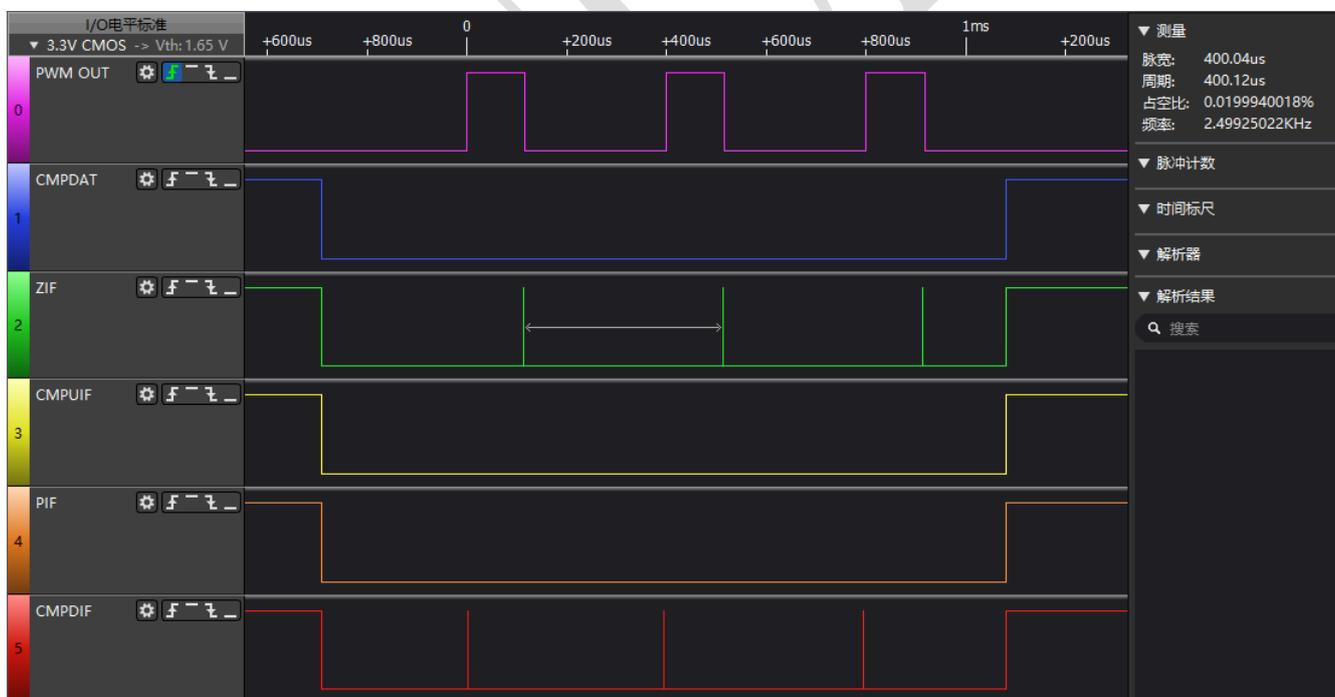
测试现象:

先正确连接 EVB 与逻辑分析仪，然后输入 ‘A’ 命令，可以看到 Log 打印待测 PWM Channel 的参数设定：

```
TGT_PWM Channel0 start...
APB CLK Freq: 64000000Hz, CLKPSC: 1, CLKDIV: 4, PERIOD: 12799, CMP: 3839
TGT_PWM stopped.

+-----+
| Press key to test specific function: |
|                                         |
| Input 'A'   Edge-Aligned PWM.         |
| Input 'B'   Center-Aligned PWM.       |
| Input 'C'   Precise Center-Aligned PWM. |
| Input 'D'   Center-Aligned PWM with Asymmetric Mode. |
| Press ESC key to back to the top level case list. |
+-----+
```

再看 LA 波形，发现 PWM 输出周期 400us（频率 2.5KHz）、占空比 30%的方波，并且在每个输出上升沿的时候 CMPDIF（CMP Down 中断）被触发，在每个输出下降沿的时候 ZIF（Zero 中断）被触发，而 CMPUIF 和 PIF 未被触发。



测试分析:

由 Log 可知，PWM 各参数设定为：APB_CLK = 64MHz, CLKPSC = 1, Divider = 1 (由 CLKDIV = 4 查 Spec 得出), PERIOD = 12799, CMPDAT = 3839，从而由边缘对齐的频率公式可得出 PWM

输出频率为:

$$\text{OutFreq} = \frac{APBCLK}{(CLKPSC + 1) * (Divider)} / (PERIOD + 1) = \frac{64MHz}{(1 + 1) * 1} / (12799 + 1) = 2.5KHz$$

再由边缘对齐的占空比公式可得出 PWM 输出占空比为:

$$\text{DutyRatio} = \frac{CMPDAT + 1}{PERIOD + 1} = \frac{3839 + 1}{12799 + 1} = 0.3 = 30\%$$

公式计算出的结果与 LA 观察到的波形一致，输出波形符合预期。

另外，从 LA 波形也可看出，PWM 的上升沿（PWM 内部递减计数值减至与 CMPDAT 相同的时刻）与 CMPDIF 中断同时出现，PWM 的下降沿（PWM 内部递减计数值减至 0 的时刻）与 ZIF 同时出现，而 CMPUIF 与 PIF 均未被触发，符合对边缘对齐中断的预期。

2.4.3.2 中心对齐（上下计数）形式

测试目的:

验证中心对齐（上下计数）形式（Center-Aligned PWM）及相关中断工作是否正常。

测试预期:

PWM 从 0 开始递增计数，当计数值大于 CMPDAT 时，输出电平翻转（由低电平变为高电平）同时触发 CMP Up 中断；计数值继续递增至 PERIOD，触发 Period 中断，但输出电平不变；随后开始递减，当计数值减至 CMPDAT 时，输出电平再次翻转（由高电平变为低电平）同时触发 CMP Down 中断；计数值继续递减至 0，触发 Zero 中断，如此往复。

测试现象:

先正确连接 EVB 与逻辑分析仪，然后输入 'B' 命令，可以看到 Log 打印待测 PWM Channel 的参数设定:

```
TGT_PWM channel0 start...
APB CLK Freq: 64000000Hz, CLKPSC: 1, CLKDIV: 4, PERIOD: 6399, CMP: 4479
TGT_PWM stopped.
```

```
Press key to test specific function:
```

```
Input 'A'   Edge-Aligned PWM.
Input 'B'   Center-Aligned PWM.
Input 'C'   Precise Center-Aligned PWM.
Input 'D'   Center-Aligned PWM with Asymmetric Mode.
Press ESC key to back to the top level case list.
```

再看 LA 波形，发现 PWM 输出周期 400us（频率 2.5KHz）、占空比 30% 的方波，并且在每个输出上升沿的时候 CMPUIF（CMP Up 中断）被触发，在每个输出高电平的中间时刻 PIF（Period 中断）被触发，在每个输出下降沿的时候 CMPDIF（CMP Down 中断）被触发，在每个输出低电平的中间时刻 ZIF（Zero 中断）被触发。



测试分析:

由 Log 可知, PWM 各参数设定为: APB_CLK = 64MHz, CLKPSC = 1, Divider = 1 (由 CLKDIV = 4 查 Spec 得出), PERIOD = 6399, CMPDAT = 4479, 从而由中心对齐的频率公式可得出 PWM 输出频率为:

$$\text{OutFreq} = \frac{APBCLK}{(CLKPSC + 1) * (Divider)} / (2 * (PERIOD + 1)) = \frac{64MHz}{(1 + 1) * 1} / (2 * (6399 + 1)) = 2.5KHz$$

再由中心对齐的占空比公式可得出 PWM 输出占空比为:

$$\text{DutyRatio} = \frac{PERIOD - CMPDAT}{PERIOD + 1} = \frac{6399 - 4479}{4479 + 1} = 0.3 = 30\%$$

公式计算出的结果与 LA 观察到的波形一致, 输出波形符合预期。

另外, 从 LA 波形也可看出, PWM 的上升沿 (PWM 内部递增计数值增至大于 CMPDAT 的时刻) 与 CMPUIF 中断同时出现, PWM 高电平的中间时刻 (PWM 内部递增计数值增至 PERIOD 的时刻) 与 PIF 同时出现, PWM 的下降沿 (PWM 内部递减计数值减至 CMPDAT 的时刻) 与 CMPDIF 中断同时出现, PWM 低电平的中间时刻 (PWM 内部递减计数值减至 0 的时刻) 与 ZIF 同时出现, 符合对中心对齐中断的预期。

2.4.3.3 精确中心对齐 (上下计数) 形式

测试目的:

验证精确中心对齐 (上下计数) 形式 (Precise Center-Aligned PWM) 及相关中断工作是否正常。

测试预期:

PWM 从 0 开始递增计数，当计数值大于 CMPDAT 时，输出电平翻转（由低电平变为高电平）同时触发 CMP Up 中断；计数值继续递增至 PERIOD/2，触发 Period 中断，但输出电平不变；随后开始递减，当计数值减至 CMPDAT 时，输出电平再次翻转（由高电平变为低电平）同时触发 CMP Down 中断；计数值继续递减至 0，触发 Zero 中断，如此往复。

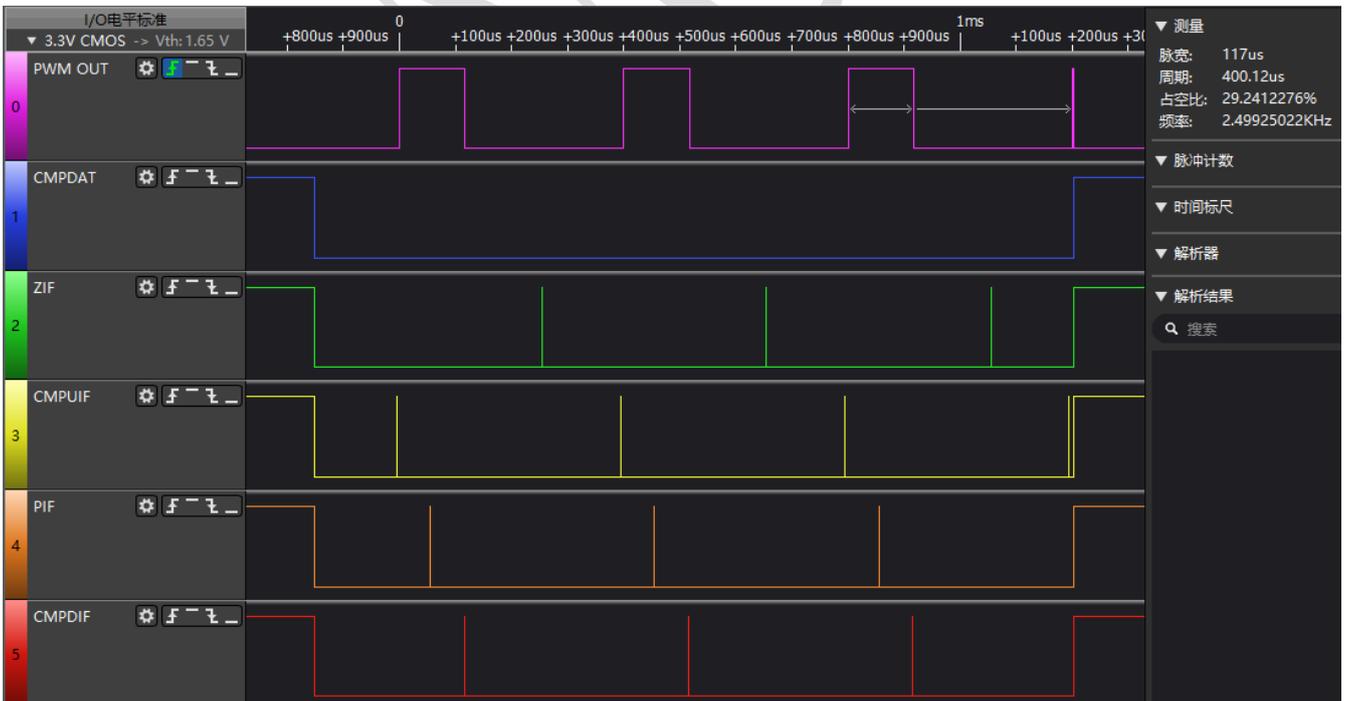
测试现象:

先正确连接 EVB 与逻辑分析仪，然后输入 ‘C’ 命令，可以看到 Log 打印待测 PWM Channel 的参数设定：

```
TGT_PWM Channel0 start...
APB CLK Freq: 64000000Hz, CLKPSC: 1, CLKDIV: 4, PERIOD: 12800, CMP: 4479
TGT_PWM stopped.
```

```
Press key to test specific function:
Input 'A'   Edge-Aligned PWM.
Input 'B'   Center-Aligned PWM.
Input 'C'   Precise Center-Aligned PWM.
Input 'D'   Center-Aligned PWM with Asymmetric Mode.
Press ESC key to back to the top level case list.
```

再看 LA 波形，发现 PWM 输出周期 400us（频率 2.5KHz）、占空比 30%的方波，并且在每个输出上升沿的时候 CMPUIF（CMP Up 中断）被触发，在每个输出高电平的中间时刻 PIF（Period 中断）被触发，在每个输出下降沿的时候 CMPDIF（CMP Down 中断）被触发，在每个输出低电平的中间时刻 ZIF（Zero 中断）被触发。



测试分析:

由 Log 可知, PWM 各参数设定为: APB_CLK = 64MHz, CLKPSC = 1, Divider = 1 (由 CLKDIV = 4 查 Spec 得出), PERIOD = 12800, CMPDAT = 4479, 从而由精确中心对齐的频率公式可得出 PWM 输出频率为:

$$\text{OutFreq} = \frac{APBCLK}{(\text{CLKPSC} + 1) * (\text{Divider})} / \text{PERIOD} = \frac{64\text{MHz}}{(1 + 1) * 1} / 12800 = 2.5\text{KHz}$$

再由精确中心对齐的占空比公式可得出 PWM 输出占空比为:

$$\text{DutyRatio} = \frac{\text{PERIOD} - 2 * (\text{CMPDAT} + 1)}{\text{PERIOD}} = \frac{12800 - 2 * (4479 + 1)}{12800} = 0.3 = 30\%$$

公式计算出的结果与 LA 观察到的波形一致, 输出波形符合预期。

另外, 从 LA 波形也可看出, PWM 的上升沿 (PWM 内部递增计数值增至大于 CMPDAT 的时刻) 与 CMPUIF 中断同时出现, PWM 高电平的中间时刻 (PWM 内部递增计数值增至 PERIOD/2 的时刻) 与 PIF 同时出现, PWM 的下降沿 (PWM 内部递减计数值减至 CMPDAT 的时刻) 与 CMPDIF 中断同时出现, PWM 低电平的中间时刻 (PWM 内部递减计数值减至 0 的时刻) 与 ZIF 同时出现, 符合对中心对齐中断的预期。

2.4.3.4 非对称计数的中心对齐形式

测试目的:

验证非对称计数的中心对齐形式 (Center-Aligned PWM with Asymmetric Mode) 及相关中断工作是否正常。

测试预期:

PWM 从 0 开始递增计数, 当计数值大于 CMPn 时, 输出电平翻转 (由低电平变为高电平) 同时触发 CMP Up 中断; 计数值继续递增至 PERIOD, 触发 Period 中断, 但输出电平不变; 随后开始递减, 当计数值减至 CMPDn 时, 输出电平再次翻转 (由高电平变为低电平) 同时触发 CMP Down 中断; 计数值继续递减至 0, 触发 Zero 中断, 如此往复。

测试现象:

先正确连接 EVB 与逻辑分析仪, 然后输入 'D' 命令, 可以看到 Log 打印待测 PWM Channel 的参数设定:

```
Asymmetric Enabled, CMPD: 1119
TGT_PWM Channel0 start...
APB CLK Freq: 64000000Hz, CLKPSC: 1, CLKDIV: 4, PERIOD: 6399, CMP: 4479
TGT_PWM stopped.
```

```
+-----+
| Press key to test specific function: |
| Input 'A'   Edge-Aligned PWM.      |
| Input 'B'   Center-Aligned PWM.    |
| Input 'C'   Precise Center-Aligned |
| Input 'D'   Center-Aligned PWM with |
|              Asymmetric Mode.      |
| Press ESC key to back to the top   |
| level case list.                    |
+-----+
```

再看 LA 波形, 发现 PWM 输出周期 400us (频率 2.5KHz)、占空比 56.2% 的方波, 并且在

每个输出上升沿的时候 CMPUIF (CMP Up 中断) 被触发, 在每个输出高电平的某一时刻 PIF (Period 中断) 被触发, 在每个输出下降沿的时候 CMPDIF (CMP Down 中断) 被触发, 在每个输出低电平的某一时刻 (非中间) ZIF (Zero 中断) 被触发。



测试分析:

由 Log 可知, PWM 各参数设定为: APB_CLK = 64MHz, CLKPSC = 1, Divider = 1 (由 CLKDIV = 4 查 Spec 得出), PERIOD = 6399, CMPn = 4479, CMPDn = 1119, 从而由中心对齐的频率公式可得出 PWM 输出频率为:

$$\begin{aligned} \text{OutFreq} &= \frac{APBCLK}{(CLKPSC + 1) * (Divider) * (2 * (PERIOD + 1))} \\ &= \frac{64MHz}{(1 + 1) * 1 * (2 * (4479 + 1))} = 2.5KHz \end{aligned}$$

再由非对称中心对齐的占空比公式可得出 PWM 输出占空比为:

$$\begin{aligned} \text{DutyRatio} &= \frac{2 * PERIOD - CMPn - CMPDn}{2 * (PERIOD + 1)} \\ &= \frac{2 * 6399 - 4479 - 1119}{2 * (6399 + 1)} = 0.562 = 56.2\% \end{aligned}$$

公式计算出的结果与 LA 观察到的波形一致, 输出波形符合预期。

另外, 从 LA 波形也可看出, PWM 的上升沿 (PWM 内部递增计数值增至大于 CMPDAT 的时刻) 与 CMPUIF 中断同时出现, PWM 高电平的某一时刻 (PWM 内部递增计数值增至 PERIOD 的时刻) 与 PIF 同时出现, PWM 的下降沿 (PWM 内部递减计数值减至 CMPDAT 的时刻) 与

CMPDIF 中断同时出现，PWM 低电平的某一时刻（PWM 内部递减计数值减至 0 的时刻）与 ZIF 同时出现，符合对中心对齐中断的预期。

2.4.4 操作模式

在主菜单下，输入 ‘3’ 命令 进入 Subcase 菜单：

```

+-----+
| Press key to test specific function:                |
| Input 'A'   Independent Mode.                      |
| Input 'B'   Complementary Mode.                   |
| Input 'C'   Complementary Mode with Dead-Time      |
|             Insertion.                             |
| Input 'D'   Synchronized Mode.                   |
| Input 'E'   Grouping Mode.                        |
| Press ESC key to back to the top level case list. |
+-----+
    
```

2.4.4.1 独立通道模式

测试目的：

验证独立通道模式（Independent Mode）工作是否正常。

测试预期：

PWM 每个通道对（Channel Pair）可以独立输出周期和占空比不同的方波。

测试现象：

先正确连接 EVB 与逻辑分析仪，然后输入 ‘A’ 命令，可以看到 Log 打印 PWM 输出开始和结束的标记：

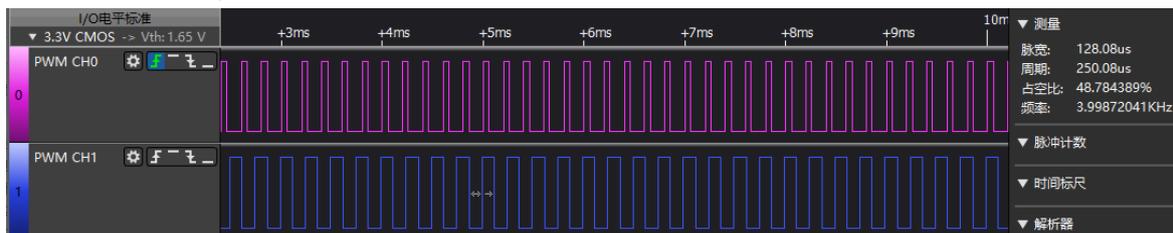
```

a
PWM start...
PWM stopped.
    
```

```

+-----+
| Press key to test specific function:                |
| Input 'A'   Independent Mode.                      |
| Input 'B'   Complementary Mode.                   |
| Input 'C'   Complementary Mode with Dead-Time      |
|             Insertion.                             |
| Input 'D'   Synchronized Mode.                   |
| Input 'E'   Grouping Mode.                        |
| Press ESC key to back to the top level case list. |
+-----+
    
```

再看 LA 波形，发现 PWM 的 Channel 0（Target Channel）与 Channel 1（Auxiliary Channel）同时输出了周期和占空比各不相同的方波（5KHz, 30%; 4KHz, 50%）：



测试分析:

从 LA 波形可看出，虽然同一个通道对（Channel Pair）共享 CLKPSC（Prescaler）寄存器，但由于 CLKDIV、PERIOD、CMPDAT 等其他寄存器仍是每个通道独立的，因此每个通道仍然可以独立输出不同周期和占空比的波形。

2.4.4.2 互补通道模式

测试目的:

验证互补通道模式（Complementary Mode）工作是否正常。

测试预期:

PWM 每个通道对（Channel Pair）可以输出互补的方波。

测试现象:

先正确连接 EVB 与逻辑分析仪，然后输入 ‘B’ 命令，可以看到 Log 打印 PWM 输出开始和结束的标记：

```
b
PWM start...
PWM stopped.
```

```

+-----+
| Press key to test specific function: |
|                                         |
| Input 'A'   Independent Mode.         |
| Input 'B'   Complementary Mode.       |
| Input 'C'   Complementary Mode with   |
|             Dead-Time Insertion.      |
| Input 'D'   Synchronized Mode.       |
| Input 'E'   Grouping Mode.           |
| Press ESC key to back to the top     |
| level case list.                     |
+-----+
    
```

再看 LA 波形，发现 PWM 的 Channel 0（Target Channel）与 Channel 1（Auxiliary Channel）同时输出了互补的方波：



测试分析:

从 LA 波形可看出，互补模式下，只需配置每个通道对（Channel Pair）的第一个 Channel（此处为 Target Channel CH0），即可分别从两个 Output 口输出互补的波形。

2.4.4.3 带死区插入的互补通道模式

测试目的:

验证带死区插入的互补通道模式（Complementary Mode with Dead-Time Insertion）工作是否正常。

测试预期:

PWM 每个通道对(Channel Pair)可以输出互补的方波,且第二个 Channel(Auxiliary Channel)有插入死区(Dead-Zone/Dead-Time)。

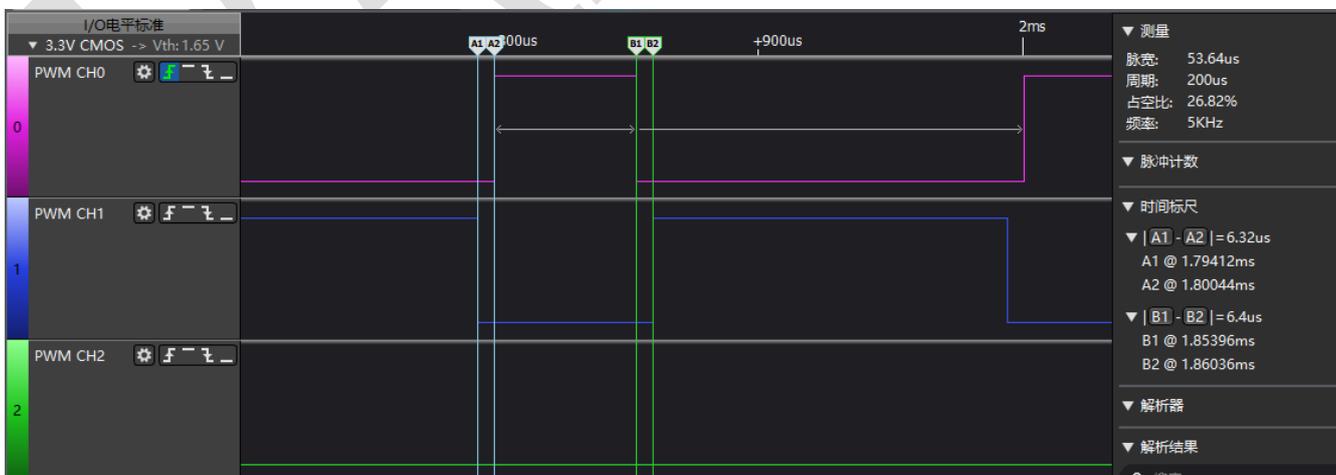
测试现象:

先正确连接 EVB 与逻辑分析仪,然后输入 'C' 命令,可以看到 Log 打印 PWM 输出开始和结束的标记,以及 PWM 配置参数、死区(Dead-Zone)持续时间等:

```
PWM Channel0 Channel1 start...
APB CLK Freq: 64000000Hz, CLKPSC: 1, CLKDIV: 4, PERIOD: 6399, CMPDAT: 1919
Dead Zone Duration: 100
TGT_PWM stopped.
```

```
-----+-----
|                                     |
| Press key to test specific function: |
|                                     |
| Input 'A'   Independent Mode.      |
| Input 'B'   Complementary Mode.    |
| Input 'C'   Complementary Mode with|
|             Dead-Time Insertion.   |
| Input 'D'   Synchronized Mode.    |
| Input 'E'   Grouping Mode.        |
| Press ESC key to back to the top  |
| level case list.                  |
|                                     |
|-----+-----
```

再看 LA 波形,发现 PWM 的 Channel 0 (Target Channel) 与 Channel 1 (Auxiliary Channel) 同时输出互补的方波(频率 5KHz,周期 200us,占空比分别为 26.85%与 66.85%),并且方波表现出如下特点,即某个波形的上升沿,一定滞后于另一个波形的下降沿一段时间,从图中也可看出滞后时间为 6.3us 左右:



测试分析:

由 Log 可知, PWM 各参数设定为: APB_CLK = 64MHz, CLKPSC = 1, Divider = 1 (由 CLKDIV = 4 查 Spec 得出), PERIOD = 6399, CMPDAT = 1919, dzDuration(Dead-Zone Duration) = 100, 又由于波形配置成边缘对齐, 由此根据公式可得出:

PWM 输出频率为:

$$\begin{aligned} \text{OutFreq} &= \frac{APBCLK}{(CLKPSC + 1) * (Divider)} / (PERIOD + 1) \\ &= \frac{64MHz}{(1 + 1) * 1} / (6399 + 1) = 5KHz \end{aligned}$$

PWM 死区 (Dead Zone) 持续时间为: [P04 和 P05 为按键复用, 出来的结果可能不准]

$$\begin{aligned} \text{DeadTime} &= (dzDuration + 1) / \frac{APBCLK}{(CLKPSC + 1) * (Divider)} \\ &= (100 + 1) / \frac{64MHz}{(1 + 1) * 1} = 3.16us \end{aligned}$$

PWM 第一个 Channel (Target Channel) 输出占空比为:

$$\begin{aligned} \text{DutyRatio} &= \frac{CMPDAT + 1 - (dzDuration + 1)}{PERIOD + 1} \\ &= \frac{1919 + 1 - (100 + 1)}{6399 + 1} = 0.2842 = 26.84\% \end{aligned}$$

PWM 第二个 Channel (Auxiliary Channel) 输出占空比为:

$$\begin{aligned} \text{DutyRatio} &= \frac{PERIOD - CMPDAT - (dzDuration + 1)}{PERIOD + 1} \\ &= \frac{6399 - 1919 - (100 + 1)}{6399 + 1} = 0.6842 = 66.84\% \end{aligned}$$

公式计算出的结果与 LA 观察到的波形一致, 输出波形符合预期。

2.4.4.4 同步通道模式

测试目的:

验证同步通道模式 (Synchronized Mode) 工作是否正常。

测试预期:

PWM 每个通道对 (Channel Pair) 只需配置第一个 Channel 的参数, 即可在两个 Channel 中输出完全同步的方波。

测试现象:

先正确连接 EVB 与逻辑分析仪, 然后输入 'D' 命令, 可以看到 Log 打印 PWM 输出开始和结束的标记:

```
d
PWM start...
PWM stopped.
```

```

+-----+
| Press key to test specific function:
|
| Input 'A'   Independent Mode.
| Input 'B'   Complementary Mode.
| Input 'C'   Complementary Mode with Dead-Time Insertion.
| Input 'D'   Synchronized Mode.
| Input 'E'   Grouping Mode.
| Press ESC key to back to the top level case list.
+-----+

```

再看 LA 波形，发现 PWM 的 Channel 0（Target Channel）与 Channel 1（Auxiliary Channel）同时输出了周期和占空比完全相同的方波：



测试分析：

从 LA 波形可看出，同步模式下，只需配置每个通道对（Channel Pair）的第一个 Channel（此处为 Target Channel CH0），即可分别从两个 Output 口输出完全相同的波形。

2.4.4.5 分组通道模式

测试目的：

验证分组通道模式（Grouping Mode）工作是否正常。

测试预期：

PWM 共有两个通道组（Channel Group），其中 Group A 包括 Channel0/2/4/6，Group B 包括 Channel1/3/5/7，同组所有 Channel 输出完全相同的方波，两个组可以分别配置输出不同的方波。

测试现象：

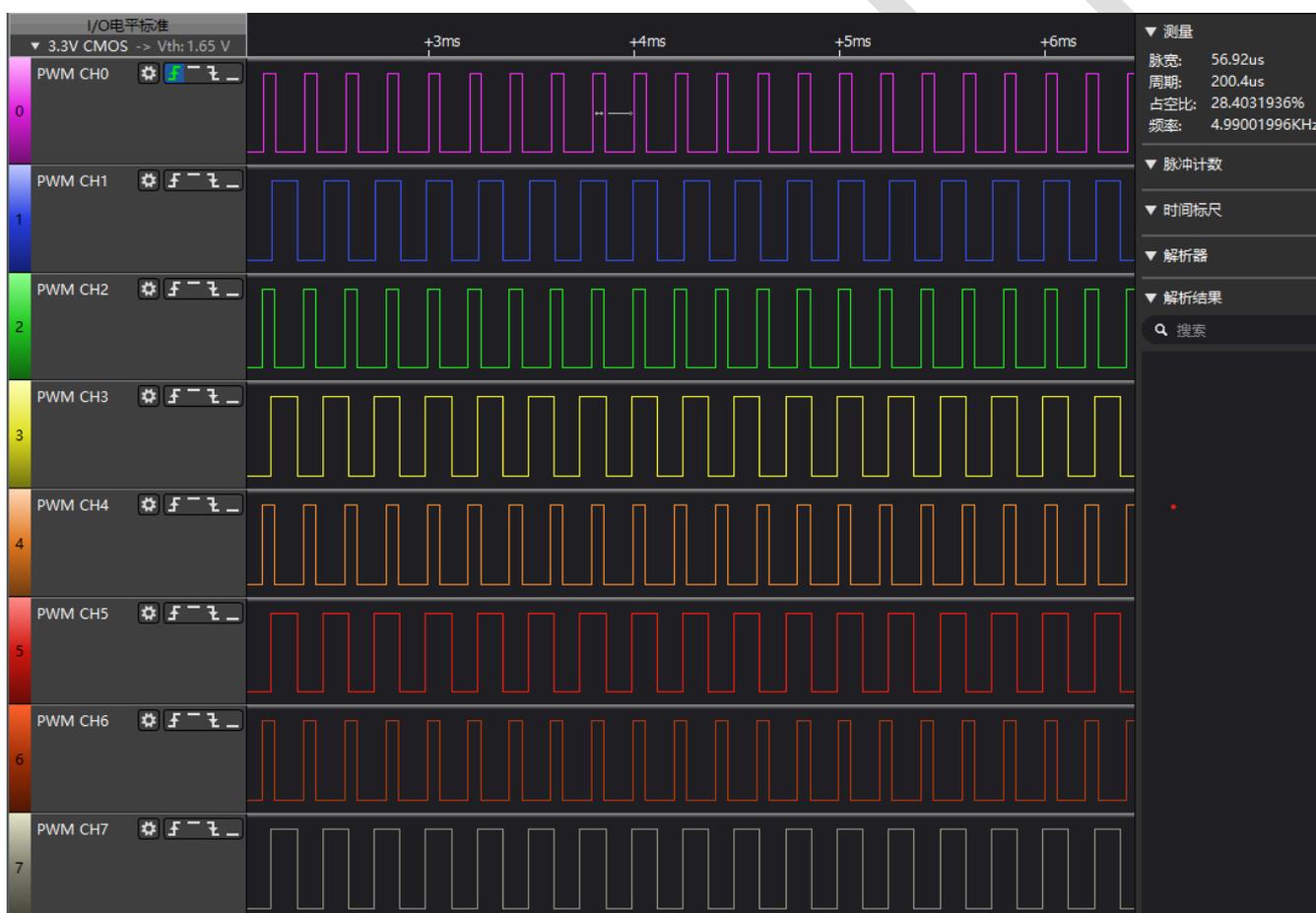
先正确连接 EVB 与逻辑分析仪（注意将所有 Channel 的 Output 都连上 LA），然后输入 ‘E’ 命令，可以看到 Log 打印 PWM 输出开始和结束的标记：

```
e
PWM start...
PWM stopped.
```

```

+-----+
| Press key to test specific function:
|
| Input 'A'   Independent Mode.
| Input 'B'   Complementary Mode.
| Input 'C'   Complementary Mode with Dead-Time Insertion.
| Input 'D'   Synchronized Mode.
| Input 'E'   Grouping Mode.
| Press ESC key to back to the top level case list.
+-----+
```

再看 LA 波形，发现 PWM 的 Channel 0/2/4/6 输出完全相同的方波，而 Channel 1/3/5/7 则输出了完全相同的且区别于偶数通道的方波：



测试分析：

从 LA 波形可看出，Grouping Mode 下，同组的所有 Channel 输出相同方波，不同组可以输出不同的方波，符合预期

2.4.5 极性控制

在主菜单下，输入 ‘4’ 命令 进入 Subcase 菜单：

```
Press key to test specific function:  
Input 'A'    output waveform with Polarity Control.  
Press ESC key to back to the top level case list.
```

测试目的:

验证极性控制 (Polarity Control) 是否正常。

测试预期:

PWM 在不同操作模式下 (独立通道模式、带死区插入的互补通道模式、同步通道模式) 极性翻转均正常。

测试现象:

先正确连接 EVB 与逻辑分析仪, 然后输入 'A' 命令, 可以看到 Log 打印 PWM 依次输出 3 种类型的波形 (独立通道模式、带死区插入的互补通道模式、同步通道模式):

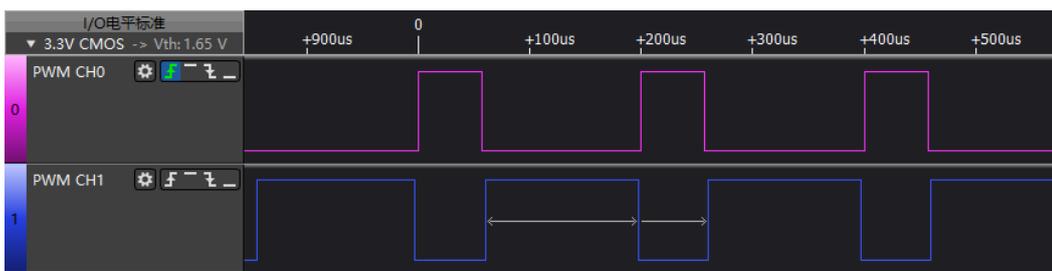
```
a  
Independent PWM wave start...  
PWM wave stopped.  
  
Complementary PWM wave with Dead-Time start...  
PWM wave stopped.  
  
Synchronized PWM wave start...  
PWM wave stopped.
```

```
Press key to test specific function:  
Input 'A'    output waveform with Polarity Control.  
Press ESC key to back to the top level case list.
```

再看 LA 波形, 发现 PWM 的确依次输出了三段不同的波形:



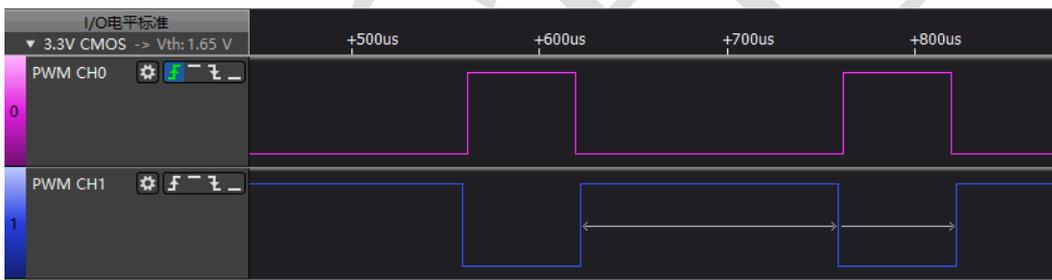
分别放大观察, 第一段是完全互补的波形:



第二段是同步的波形，但是带有死区插入：



第三段也是完全互补的波形：



测试分析：

从测试代码和 Log 可知，第一段波形的两个 channel 是配置成独立通道模式，且周期、占空比、计数形式等参数也完全相同，唯一不同的是第二个通道的极性翻转功能被打开，因此 Output 真正输出的波形应该为互补的，而从 LA 抓到的波形来看的确如此，符合预期。

第二段波形的两个 channel 是配置成带死区插入的互补模式，并且第二个通道的极性翻转功能被打开，因此 Output 真正输出的波形应该为同步的，但由于死区插入的影响，两个波形不会完全一致，从 LA 抓到的波形来看的确如此，符合预期。

第三段波形的两个 channel 是配置成同步模式，且第二个通道的极性翻转功能被打开，因此 Output 真正输出的波形应该是互补的，从 LA 抓到的波形来看的确如此，符合预期。

2.4.6 简单 API 接口演示

在主菜单下，输入 ‘5’ 命令 进入 Subcase 菜单：

```

Press key to test specific function:

Input 'A'    PWM Proc 1 Init.
Input 'B'    PWM Proc 1 Start/Stop.
Input 'C'    PWM Proc 2 Demo.
Press ESC key to back to the top level case list.
    
```

2.4.6.1 演示 Demo 1

测试目的:

使用简单 API 接口，演示同时输出两路不同的 PWM 波形。

测试预期:

Demo 例程行为正常。

测试现象:

先正确连接 EVB 与逻辑分析仪，然后输入 ‘A’ 命令，可以看到如下的 Log 打印，表示 Demo Proc1 成功初始化：

```

a
Initialize PWM in simple way procl.
    
```

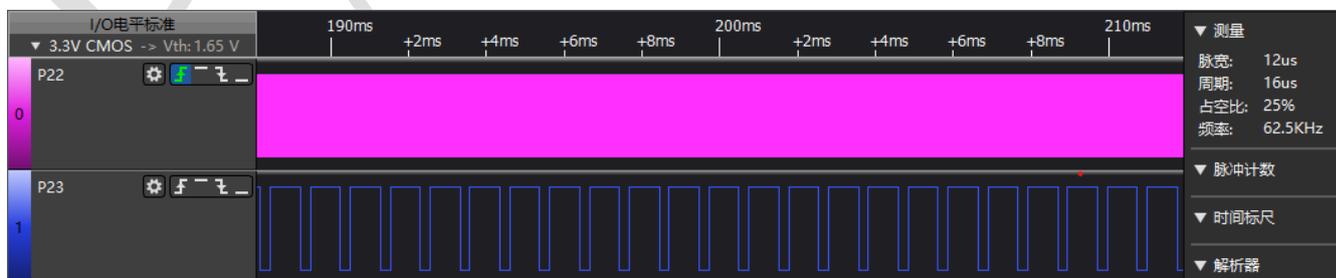
```

Press key to test specific function:

Input 'A'    PWM Proc 1 Init.
Input 'B'    PWM Proc 1 Start/Stop.
Input 'C'    PWM Proc 2 Demo.
Press ESC key to back to the top level case list.
    
```

接着输入 ‘B’ 命令，开启 PWM 波形翻转，此时观察 LA 波形，发现有 2 路 PWM 开始波形输出：

然后再输入一遍 ‘B’ 命令，停止 PWM 波形翻转，此时观察 LA 波形，发现 PWM 波形输出停止：



分别放大观察，P22 上是一个周期 16us，其中高电平持续 4us 的周期波形：



P23 上是一个周期 1024us，其中高电平持续 768 的波形：



测试分析：

从测试代码可知，系统时钟为 64MHz，AHB 与 APB 均不分频，而 PWM 模块挂在 APB1 上，因此 PWM 模块的输入时钟与 APB1 相同，为 64MHz。

第一段波形（P22）为 PWM1 Channel 4，其预分频因子 Prescaler（CLKPSC）被配置为 1，分频因子 Divider 被配置为 2，由计数频率公式：

$$\text{CntFreq} = \frac{APBCLK}{(\text{CLKPSC} + 1) * (\text{Divider})} = \frac{64\text{MHz}}{(1 + 1) * 2} = 16\text{MHz}$$

可知此 PWM 通道的内部计数频率为 16MHz，即内部计数一个 count 的时间为 $1/16 = 0.0625$ us。

又由于此例程未特别配置计数形式，因此使用默认的边沿对齐（向下计数）的方式，于是根据公式有：

输出周期：

$$t_{\text{OutputPeriod}} = t_{\text{percnt}} * (\text{PERIOD} + 1) = 0.0625 * (255 + 1) = 16 \text{ (us)}$$

输出高电平脉宽：

$$t_{\text{OutputPulse}} = t_{\text{percnt}} * (\text{CMPDAT} + 1) = 0.0625 * (63 + 1) = 4 \text{ (us)}$$

即预期输出波形的周期为 16us，其中高电平 4us，而从 LA 抓到的波形来看确实如此，符合预期。

第二段波形（P23）为 PWM1 Channel 5，其预分频因子 Prescaler（CLKPSC）与 PWM1 Channel 6 共用（即为 1），分频因子 Divider 被配置为 PWM_CLK_APB，表示当前的 PWM 通道内部时钟直接使用 APB 时钟（即忽略预分频因子和分频因子）：

$$\text{CntFreq} = APBCLK = 64\text{MHz}$$

即内部计数一个 count 的时间为 $1/64 = 0.015625$ us。

于是根据公式有：

输出周期:

$$t_{OutputPeriod} = t_{percnt} \times (PERIOD + 1) = 0.015625 \times (65535 + 1) = 1024 \text{ (us)}$$

输出高电平脉宽:

$$t_{OutputPulse} = t_{percnt} \times (CMPDAT + 1) = 0.015625 \times (16383 + 1) = 256 \text{ (us)}$$

即预期输出波形的周期为 1024us，其中高电平 256us，又由于代码中开启了 Output Inverter，即输出波形反相功能，因此最终输出波形应为周期 1024us，其中低电平为 256us，而从 LA 抓到的波形来看确实如此，符合预期。

2.4.6.2 演示 Demo 2

测试目的:

使用简单 API 接口，演示输出可变占空比的 PWM 波形。

测试预期:

Demo 例程行为正常。

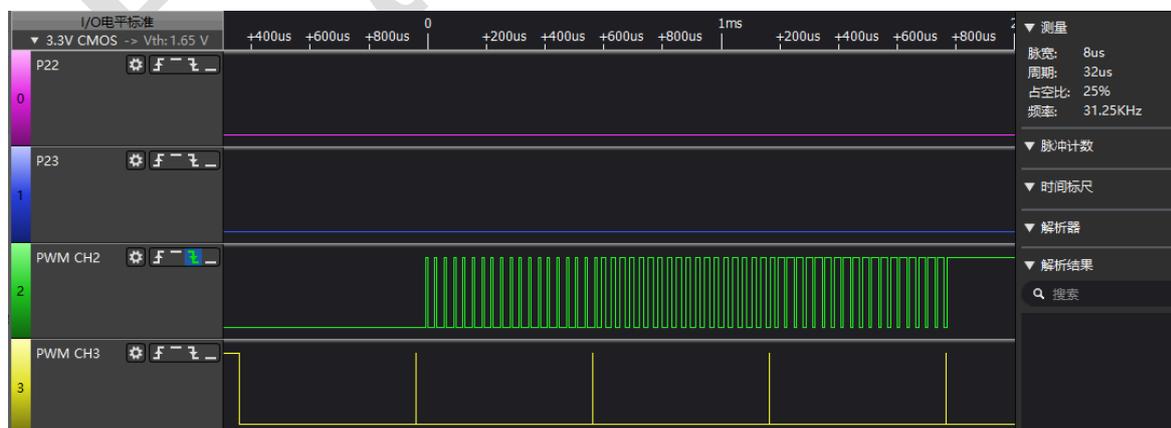
测试现象:

先正确连接 EVB 与逻辑分析仪，然后输入 ‘C’ 命令，可以看到如下的 Log 打印，表示成功输出 PWM 波形:

```
C
Demonstrate PWM changing duty cycle.
PWM Stopped.
```

```
+-----+
| Press key to test specific function: |
|                                         |
| Input 'A'   PWM Proc 1 Init.         |
| Input 'B'   PWM Proc 1 Start/Stop.   |
| Input 'C'   PWM Proc 2 Demo.         |
| Press ESC key to back to the top level case list. |
+-----+
```

观察 LA 波形，发现 P30 引脚有 PWM 波形输出，P02 引脚有 GPIO 电平翻转，持续一段时间后停止:



在 GPIO P02 低电平的 5 个窗口内（每个窗口持续约 0.6ms），分别观察到 PWM 输出波形：

窗口 1：PWM 输出长低电平；

窗口 2：PWM 输出周期 32us，其中高电平脉宽持续 8.03us；

窗口 3：PWM 输出周期 64us，其中高电平脉宽持续 16.03us；

窗口 4：PWM 输出周期 64us，其中高电平脉宽持续 24.03us；

窗口 5：PWM 输出长高电平；

测试分析：

从测试代码可知，系统时钟为 64MHz，AHB 与 APB 均不分频，而 PWM 模块挂在 APB1 上，因此 PWM 模块的输入时钟与 APB1 相同，为 64MHz。

此段波形（P30）使用 PWM1 Channel 6，其预分频因子 Prescaler（CLKPSC）被配置为 1，分频因子 Divider 被配置为 1，由计数频率公式：

$$\text{CntFreq} = \frac{APBCLK}{(\text{CLKPSC} + 1) * (\text{Divider})} = \frac{64\text{MHz}}{(1 + 1) * 1} = 32\text{MHz}$$

可知此 PWM 通道的内部计数频率为 32MHz，即内部计数一个 count 的时间为 $1/32 = 0.03125$ us。

又由测试代码，PWM 输出周期 PERIOD 被配置为 1023，输出高电平脉宽在窗口 1 内被配置为 0，在窗口 2 内被配置为 256，在窗口 3 内被配置为 512，在窗口 4 内被配置为 768，在窗口 5 内被配置为 1024；

于是根据公式分别有：

输出周期：

$$t_{\text{OutputPeriod}} = t_{\text{percnt}} \times (\text{PERIOD} + 1) = 0.03125 \times (1023 + 1) = 32 \text{ (us)}$$

窗口 1 的输出高电平脉宽：

由于 CMPDAT 被配置为 0 时，硬件会固定输出长低电平，因此有：

$$t_{\text{OutputPulse1}} = 0 \text{ (us)}$$

窗口 2 的输出高电平脉宽：

$$t_{\text{OutputPulse2}} = t_{\text{percnt}} \times (\text{CMPDAT} + 1) = 0.03125 \times (256 + 1) = 8.03125 \text{ (us)}$$

窗口 3 的输出高电平脉宽：

$$t_{\text{OutputPulse3}} = t_{\text{percnt}} \times (\text{CMPDAT} + 1) = 0.03125 \times (512 + 1) = 16.03125 \text{ (us)}$$

窗口 4 的输出高电平脉宽：

$$t_{\text{OutputPulse4}} = t_{\text{percnt}} \times (\text{CMPDAT} + 1) = 0.03125 \times (768 + 1) = 24.03125 \text{ (us)}$$

窗口 5 的输出高电平脉宽：

$$t_{OutputPulse5} = t_{percent} \times (CMPDAT + 1) = 0.03125 \times (1024 + 1) = 32.03125 \text{ (us)}$$

注意此处计算得出的高电平脉宽 32.03125us，大于周期 32us，因此窗口 5 将会输出长高电平。

从 LA 抓到的波形来看，窗口 1~窗口 5 输出的 PWM 波形均与计算一一对应，符合预期。

PANCHIP

第3章 使用注意事项

- 1、注意每个通道对 (Channel Pair) 共用一个 CLKPSC (Prescaler)，但 CLKDIV (Divider)、CMPDAT、PERIOD 等寄存器是每个通道独立的，因此当希望同一个 channel pair 在 Independent Mode 下输出不同的波形时，PWM_ConfigOutputChannel() API 会保证 CLKPSC 值设定后不会再被修改 (做法是只有发现 CLKPSC 为 0 时，才会重新计算 Prescaler，否则将会保留 CLKPSC 中的值，并基于此值来计算 Divider 等其他参数)。然而，这样也会带来一些问题，比如：
 - a) Channel Pair 的两个通道，周期/占空比设定值相差过大，则可能导致正确计算出前一个通道的参数后，CLKPSC 被固定在一个值，而这个值无法满足后一个通道的参数要求，从而导致后一个通道无法正确输出期望的波形。解决办法一是尽量不要用一个 Channel Pair 的两个通道输出差别过大的波形，二是使用其他独立的通道来输出需要的波形。
 - b) 同一个通道，先配置输出某个波形，停止输出后，再配置输出另一个波形。这种情况下，第二次配置不会改变第一次计算出的 CLKPSC 值，于是也可能会出现 CLKPSC 的值无法满足当前参数要求的情形。解决办法是在调用 PWM_ConfigOutputChannel() 之前，先使用 PWM_ResetPrescaler() API 将 CLKPSC 清掉，这样就可以重新计算合适的 CLKPSC 了。
- 2、只有 Center-Aligned Type 下才可开启 Asymmetric Mode，在 Precise Center-Aligned Type 不可开启，否则将会无波形输出
- 3、因为 Precise Center-Aligned Type、Center-Aligned Type 和 Edge-Aligned Type 设置均是对所有 channel 均有效，因此当需要多通道同时输出波形的时候，使用 PWM_ConfigOutputChannel() API 来配置的时候，Operate Type 应当保持一致，否则会得不到预期波形
- 4、注意两个 API: PWM_Stop() 与 PWM_ForceStop() 的区别：前者仅仅是将 PERIOD 设置为 0，CNTEN 未清掉，因此波形在调用 PWM_Stop() 后不会立刻停止，而是会把当前周期输出完成，才会将输出电平拉低；后者是直接清 CNTEN，所以波形会立刻停止。另外需注意，如果使用 PWM_Stop()，后面又配置了 PERIOD 为非零值 (如调用了 PWM_ConfigOutputChannel() 函数)，则会立刻有输出，而不会再等到 PWM_Start()
- 5、互补输出模式下，Enable Dead-time insertion 后，若执行了 stop()，则第二个波形仍然会有 dead-time 的输出，需要软件 DisableDeadZone() 才可以：
 - a) Enable DeadZone 硬件实际上是在已有波形的基础上叠加一个相位有偏移的波形，因此 EnableDeadZone() 与 Start() 调用位置越接近越好，同理 DisableDeadZone() 与 Stop() 也是如此。
 - b) EnableOutput() 最好与 Start() 调用位置越接近越好，否则可能在 Start 之前会有较长时间的初始无效电平
- 6、注意不同 counting type 下，PERIOD 和 CMPDAT 的关系，不合理的相对值将可能会导致输出全为高电平或低电平
- 7、PWM Module 下 ADC 相关寄存器是无效的，若要使用 ADC Trigger，应去 ADC Module 下

配置相关寄存器。

- 8、CLKPSC (Prescaler) 不可以为 0，否则将会无波形输出。
- 9、PERIOD 或者 CMPDAT 为 0 时，固定输出低电平

PANCHIP