M480 PWM Synchronous

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	Demo the PWM Synchronous function
BSP Version	M480 Series BSP CMSIS V3.04.000
Hardware	NuMaker-FPM-M487 v3.0

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

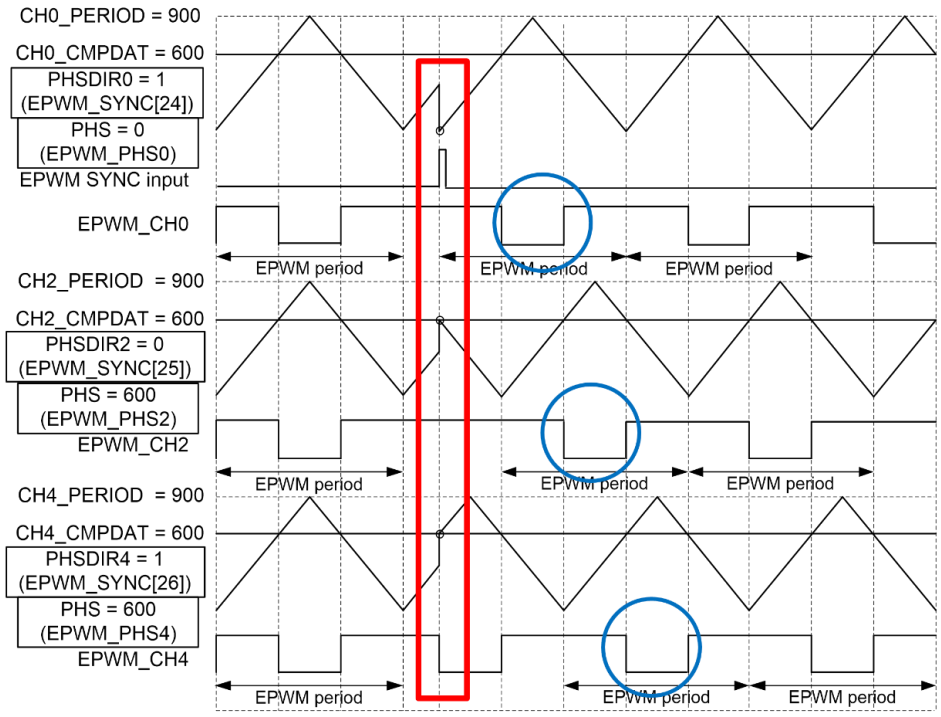
1 Function Description

1.1 Introduction

The PWM Synchronous function is used for making different PWM channel in phase. User can control phase value and direction. Be sure to set PWM in complementary mode when using this function. The SYNC_IN signal for the first EPWM0 pair counter comes from EPWM0_SYNC_IN pin, and the others come from the SYNC_OUT signal of the previous EPWM pair counter.

1.2 Demo Result

After trigger by SYNC_IN or SWSYNC or Counter, the selected channel will start synchronizing.



Note: ○ SYNC PHS & PHSDIR Load

2 Code Description

These functions are used to initialize and enable EPWM. In the EPWM_ConfigOutputChannel () function, setting the clock source、frequency and duty for the selected channel.

```
    /* Configure EPWM generator and get the nearest frequency in edge aligned auto-reload mode*/
    EPWM_ConfigOutputChannel (EPWM_T *epwm, uint32_t u32ChannelNum, uint32_t u32Frequency, uint32_t u32DutyCycle);
    PWM_EnableOutput (EPWM_T *epwm, uint32_t u32ChannelMask);


    /* Configure synchronization phase of selected channel*/
    EPWM_ConfigSyncPhase (EPWM_T *epwm, uint32_t u32ChannelNum, uint32_t u32SyncSrc, uint32_t u32Direction, uint32_t u32StartPhase);


    /* Enable SYNC phase of selected channel */
    EPWM_EnableSyncPhase (EPWM_T *epwm, uint32_t u32ChannelMask);


    /* set PWM1 module in Complementary MODE */
    EPWM_ENABLE_COMPLEMENTARY_MODE (EPWM_T *pwm);
```

```
/**
 * @brief This function Configure EPWM generator and get the nearest frequency in edge aligned (up counter type) auto-reload mode
 * @param[in] epwm The pointer of the specified EPWM module
 *                - EPWM0: EPWM Group 0
 *                - EPWM1: EPWM Group 1
 * @param[in] u32ChannelNum EPWM channel number. Valid values are between 0~5
 * @param[in] u32Frequency Target generator frequency
 * @param[in] u32DutyCycle Target generator duty cycle percentage. Valid range are between 0 ~ 100. 10 means 10%, 20 means 20%...
 * @return Nearest frequency clock in nano second
 * @note Since every two channels, (0 & 1), (2 & 3), shares a prescaler. Call this API to configure EPWM frequency may affect
 *       existing frequency of other channel.
 * @note This function is used for initial stage.
 *       To change duty cycle later, it should get the configured period value and calculate the new comparator value.
 */
uint32_t EPWM_ConfigOutputChannel (EPWM_T *epwm, uint32_t u32ChannelNum, uint32_t u32Frequency, uint32_t u32DutyCycle)
{
    uint32_t u32Src;
    uint32_t u32EPWMClockSrc;
    uint32_t i;
```

```
    uint32_t u32Prescale = 1U, u32CNR = 0xFFFFU;

    if(epwm == EPWM0)
    {
        u32Src = CLK->CLKSEL2 & CLK_CLKSEL2_EPWM0SEL_Msk;
    }
    else     /* (epwm == EPWM1) */
    {
        u32Src = CLK->CLKSEL2 & CLK_CLKSEL2_EPWM1SEL_Msk;
    }

    if(u32Src == 0U)
    {
        /* clock source is from PLL clock */
        u32EPWMClockSrc = CLK_GetPLLClockFreq();
    }
    else
    {
        /* clock source is from PCLK */
        SystemCoreClockUpdate();
        if(epwm == EPWM0)
        {
            u32EPWMClockSrc = CLK_GetPCLK0Freq();
        }
        else     /* (epwm == EPWM1) */
        {
            u32EPWMClockSrc = CLK_GetPCLK1Freq();
        }
    }

    for(u32Prescale = 1U; u32Prescale < 0xFFFU; u32Prescale++)   /* prescale resolution is
12bits: 0~0xFFF */
    {
        i = (u32EPWMClockSrc / u32Frequency) / u32Prescale;
        /* If target value is larger than CNR, need to use a larger prescaler */
        if(i < (0x10000U))
        {
            u32CNR = i;
            break;
        }
    }
```

```
    /* Store return value here 'cos we're goanna change u16Prescale & u16CNR to the real
value to fill into register */
    i = u32EPWMClockSrc / (u32Prescale * u32CNR);


    /* convert to real register value */
    /* every two channels share a prescaler */
    u32Prescale -= 1U;
    EPWM_SET_PRESCALER(epwm, u32ChannelNum, u32Prescale);
    /* set EPWM to up-down counter type(edge aligned) and auto-reload mode */
    (epwm)->CTL1 = ((epwm)->CTL1 & ~(((1UL << EPWM_CTL1_CNTTYPE0_Pos) << (u32ChannelNum <<
1U))|((1UL << EPWM_CTL1_CNTMODE0_Pos) << u32ChannelNum)));


    u32CNR -= 1U;
    EPWM_SET_CNR(epwm, u32ChannelNum, u32CNR);
    EPWM_SET_CMR(epwm, u32ChannelNum, u32DutyCycle * (u32CNR + 1U) / 100U);




    (epwm)->WGCTL0 = ((epwm)->WGCTL0 & ~(((1UL << EPWM_WGCTL0_PRDPCTL0_Pos) | (1UL <<
EPWM_WGCTL0_ZPCTL0_Pos)) << (u32ChannelNum << 1U))) | \
                     ((uint32_t)EPWM_OUTPUT_HIGH << ((u32ChannelNum << 1U) +
(uint32_t)EPWM_WGCTL0_ZPCTL0_Pos));
    (epwm)->WGCTL1 = ((epwm)->WGCTL1 & ~(((1UL << EPWM_WGCTL1_CMPDCTL0_Pos) | (1UL <<
EPWM_WGCTL1_CMPUCTL0_Pos)) << (u32ChannelNum << 1U))) | \
                     ((uint32_t)EPWM_OUTPUT_LOW << ((u32ChannelNum << 1U) +
(uint32_t)EPWM_WGCTL1_CMPUCTL0_Pos));


    return(i);
}
```

3 Software and Hardware Environment

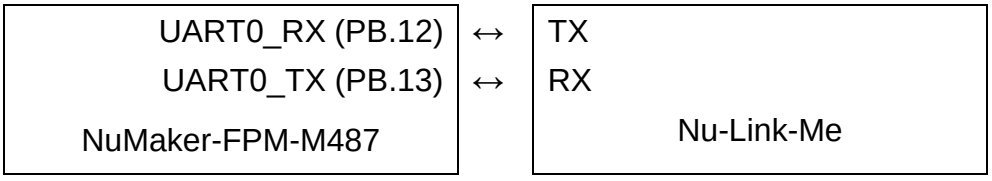
Software Environment

- BSP version
  - M480 Series BSP CMSIS V3.04.000
- IDE version
  - Keil uVersion 5.28

Hardware Environment

- Circuit components
  - NuMaker-FPM-M487
- Diagram

M480's UART0_RX (PB.12) and UART0_TX (PB.13) are connected to Nu-Link Me to print the message. Set COM port and Baud. The number of the COM Port can be found in the device manager "NuBridge Virtual Com Port (COMX)" and Baud is set to 115200.



4 Directory Information

- EC_M480_PWM_Synchronous_V1.00
  - Library — Sample code header and source files
    - CMSIS — Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
    - Device — CMSIS compliant device header file
    - StdDriver — All peripheral driver header and source files
  - SampleCode
    - ExampleCode — Source file of example code

5 How to Execute Example Code

1. Browsing into sample code folder by Directory Information (section 4) and double click M480_PWM_Sync.uvprojx.
2. Enter Keil compile mode
   a. Build
   b. Download
   c. Start/Stop debug session
3. Enter debug mode
   a. Run

6 Revision History

Date	Revision	Description
Dec. 2, 2019	1.00	1. Initially issued.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.