M480 Battery Discharge Distribution

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	This sample code uses M480 series microcontroller (MCU) to output 6 PWM waveforms for 6 batteries' discharge distribution.
BSP Version	M480 BSP CMSIS V3.05.001
Hardware	NuMaker-M483KG Ver1.1

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 and microprocessor 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 Overview

Using multiple batteries in parallel will increase the current supply capacity. Due to the difference of battery voltage, the batteries cannot be directly connected in parallel. If the load current is 6A and the power supply time of each battery is 1 ms within 6 ms as Figure 1-1, the average current of one battery is only 1A in a long term.

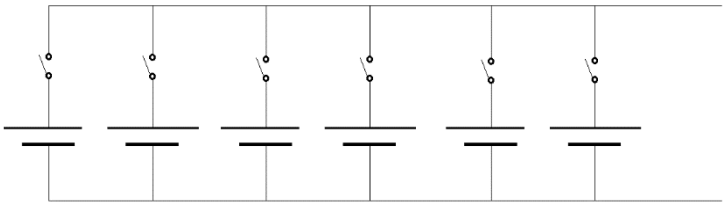

Figure 1-1 6 Batteries Connected in Parallel via Switch

Therefore, it is necessary to generate a set of pulses for switching on the battery distribution.

1.1 Demo Result

If executing this code on M480 series MCU, 6 PWM waveforms will be generated and Figure 1-2 shows 4 waveforms of them.

Figure 1-2 Four of Six PWM Waveforms

2 Code Description

The M480 series MCU has two sets of PWMs, EPWM0 and EPWM1. Both can be configured as 3 pairs of complementary PWMs. Each pair of complementary PWMs can be configured as one pules as Figure 2-1. 6 channels of PWM can be configured to start synchronously. There is a difference in the duty because of the difference in the amount of battery power.

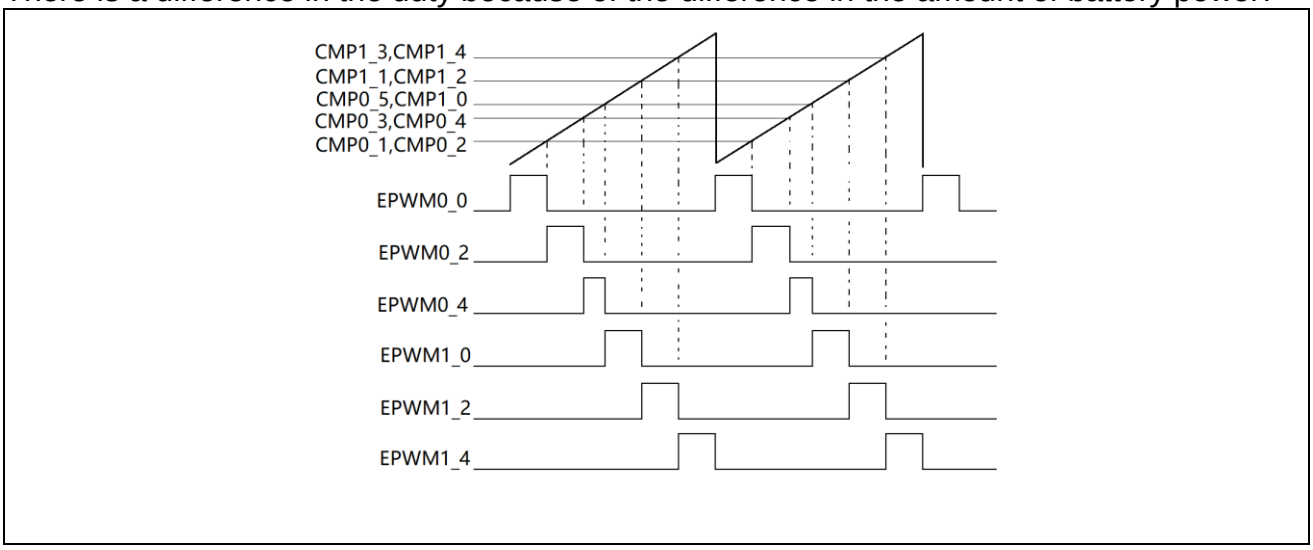

Figure 2-1 6 PWMs Generation

EPWM0_CH024 and EPWM1_CH024 must be configured to start synchronously in complementary mode.

```
    EPWM0->CLKPSC[0] = 159 ;    // prescaler=(159+1),PWM counter frequency =160M/160=1M
    EPWM0->CLKPSC[1] = 159 ;
    EPWM0->CLKPSC[2] = 159 ;
    EPWM1->CLKPSC[0] = 159 ;
    EPWM1->CLKPSC[1] = 159 ;
    EPWM1->CLKPSC[2] = 159 ;

    EPWM0->CTL1   = 0x07000000;    // PWM01,23,45 is complementary mode, Up counter type
    EPWM1->CTL1   = 0x07000000;

    EPWM0->WGCTL0 = 0x00000556 ;      // EPWM0_CH0 Zero point output 1, other CH output 0
    EPWM1->WGCTL0 = 0x00000555 ;

    EPWM0->WGCTL1 = 0x00000004        // Compare point, CMP0_1 output 0
                  | 0x00000060      // CMP0_2 output 1 and CMP0_3 output 0
                  | 0x00000600 ;    // CMP0_4 output 1 and CMP0_5 output 0

    EPWM1->WGCTL1 = 0x00000006        // Compare point, CMP1_0 output 1, CMP1_1 output 0
                  | 0x00000060      // CMP1_2 output 1 , CMP1_3 output 0
                  | 0x00000200 ;    // CMP1_4 output 1

    EPWM0->PERIOD[0] = 100 * PWM_Percent_one - 1 ;        // Set PWM period
    EPWM0->PERIOD[2] = 100 * PWM_Percent_one - 1 ;
    EPWM0->PERIOD[4] = 100 * PWM_Percent_one - 1 ;
    EPWM1->PERIOD[0] = 100 * PWM_Percent_one - 1 ;
    EPWM1->PERIOD[2] = 100 * PWM_Percent_one - 1 ;
    EPWM1->PERIOD[4] = 100 * PWM_Percent_one - 1 ;

    EPWM0->CNTCLR = ~0 ;
    EPWM0->INTSTS0 = ~0 ;
```

```
    EPWM0->INTEN0 = 1 ;             // Enable EPWM0_CH0 Zero point interrupt

    NVIC_SetPriority(EPWM0P0_IRQn, 2) ;
    NVIC_ClearPendingIRQ(EPWM0P0_IRQn);
    NVIC_EnableIRQ(EPWM0P0_IRQn);                 // Enable PWM0-CH0 interrupt

    /* Enable PWM024 starting count synchronously */
    EPWM_ENABLE_TIMER_SYNC(EPWM0, 0x15, EPWM_SSCTL_SSRC_EPWM0);
    EPWM_ENABLE_TIMER_SYNC(EPWM1, 0x15, EPWM_SSCTL_SSRC_EPWM0);

    /* Trigger EPWM counter synchronous start by EPWM0 */
    EPWM_TRIGGER_SYNC_START(EPWM0);

    /* Enable output of PWM0 channel 0,2,4 */
    EPWM_EnableOutput(EPWM0, 0x15);
    EPWM_EnableOutput(EPWM1, 0x15;
```

In order to keep waveform integral, changing duty must be in one period. The duty is changed only when PWM_Distribution.Update = 1, which ensures that the duty of 6 channels is changed in one PWM period.

```
//Updating percent is at the begining of period. So,waveform can be keep integrity.
void EPWM0P0_IRQHandler(void)
{
    int32_t temp32 = PWM_Distribution.Battery1_Percent ;

    if (PWM_Distribution.Update)
    {
        EPWM0->CMPDAT[1] = temp32 * PWM_Percent_one ;               // Battery 1

        EPWM0->CMPDAT[2] = EPWM0->CMPDAT[1] ;                       // Battery 2
        temp32 += PWM_Distribution.Battery2_Percent ;
        EPWM0->CMPDAT[3] = temp32 * PWM_Percent_one ;

        EPWM0->CMPDAT[4] = EPWM0->CMPDAT[3] ;                       // Battery 3
        temp32 += PWM_Distribution.Battery3_Percent ;
        EPWM0->CMPDAT[5] = temp32 * PWM_Percent_one ;

        EPWM1->CMPDAT[0] = EPWM0->CMPDAT[5] ;                       // Battery 4
        temp32 += PWM_Distribution.Battery4_Percent ;
        EPWM1->CMPDAT[1] = temp32 * PWM_Percent_one ;

        EPWM1->CMPDAT[2] = EPWM1->CMPDAT[1] ;                       // Battery 5
        temp32 += PWM_Distribution.Battery5_Percent ;
        EPWM1->CMPDAT[3] = temp32 * PWM_Percent_one ;

        EPWM1->CMPDAT[4] = EPWM1->CMPDAT[3] ;                       // Battery 6

        PWM_Distribution.Update = 0;
    }

    EPWM0->INTSTS0 = ~0 ;                          // Clear Flag of interrupt
}
```

3 Software and Hardware Requirements

3.1 Software Requirements

- BSP version
  - M480_BSP_CMSIS_V3.05.001
- IDE version
  - Keil uVersion 5.26

3.2 Hardware Requirements

- Circuit components
  - NuMaker-M483KG V1.1
- Pin Connect

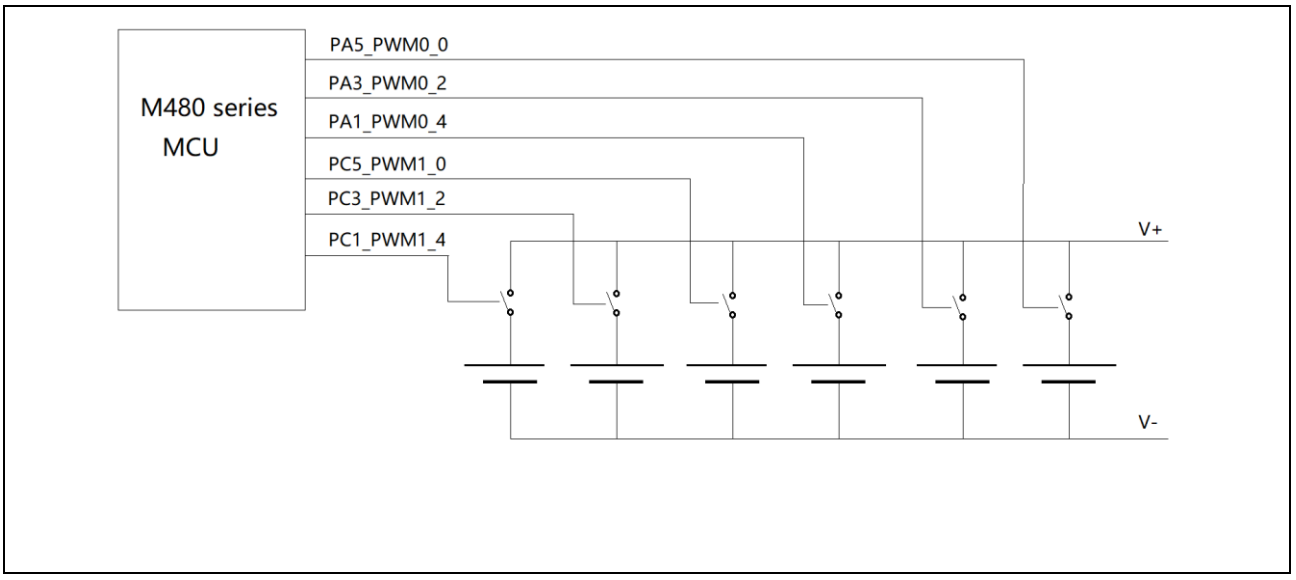

Figure 3- 1 Pin Connect

4 Directory Information

The directory structure is shown below.

EC_ M480_Battery_Discharge_Distribution_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

Figure 4-1 Directory Structure

5 Example Code Execution

1. Browse the sample code folder as described in the Directory Information section and double-click *M480_Battery_Discharge_Distribution.uvproj*.
2. Enter Keil compile mode.
   - Build
   - Download
   - Start/Stop debug session
3. Enter debug mode.
   - Run

6 Revision History

Date	Revision	Description
2021.12.09	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.