PWM Sine Wave

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	Using PWM0 Channel 0 (PA.5) and RC low-pass filter to generate 60 Hz sine wave for application.
BSP Version	M031_Series_BSP_CMSIS_V3.02.000
Hardware	NuMaker-M032SE V1.3

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

This example code uses M031 PWM0 Channel 0 (PA.5) to generate 60 Hz sine wave. To generate high-resolution sine wave, a single sine wave will have 360 points. Based on sine value calculation and PWM period value, each point has different PWM duty value. Then PWM changes the duty value according to each point value and sine wave frequency. By using external RC filter, PWM output would be smooth sine wave.

1.2 Principle

To get the PWM duty value of 360 points, the example code calculates the sine value with multiplying PWM period value for each point. The result creates an array for lookup table, which is used when PWM changes the duty value. Since the sine value is between -1 and 1, it needs to add offset value 1 to move sine value between 0 and 2. Then divide value with 2 to compress sine value between 0 and 1.

The example code also uses Timer0 and enables Timer0 interrupt to change PWM duty value. The Timer0 uses periodic mode to make Timer0 auto-reload timer value for periodic interrupt. The time-out period formula shows below:

$$\text{Time-out Period} = \frac{1}{\text{TMR0_FREQ}} = \frac{1}{\text{Sine Wave Frequency} * \text{Sine Wave Resolution}}$$

Since the sine wave frequency is 60 Hz and the sine wave resolution is 360, the TMR0_FREQ calculated by the formula is 21600.

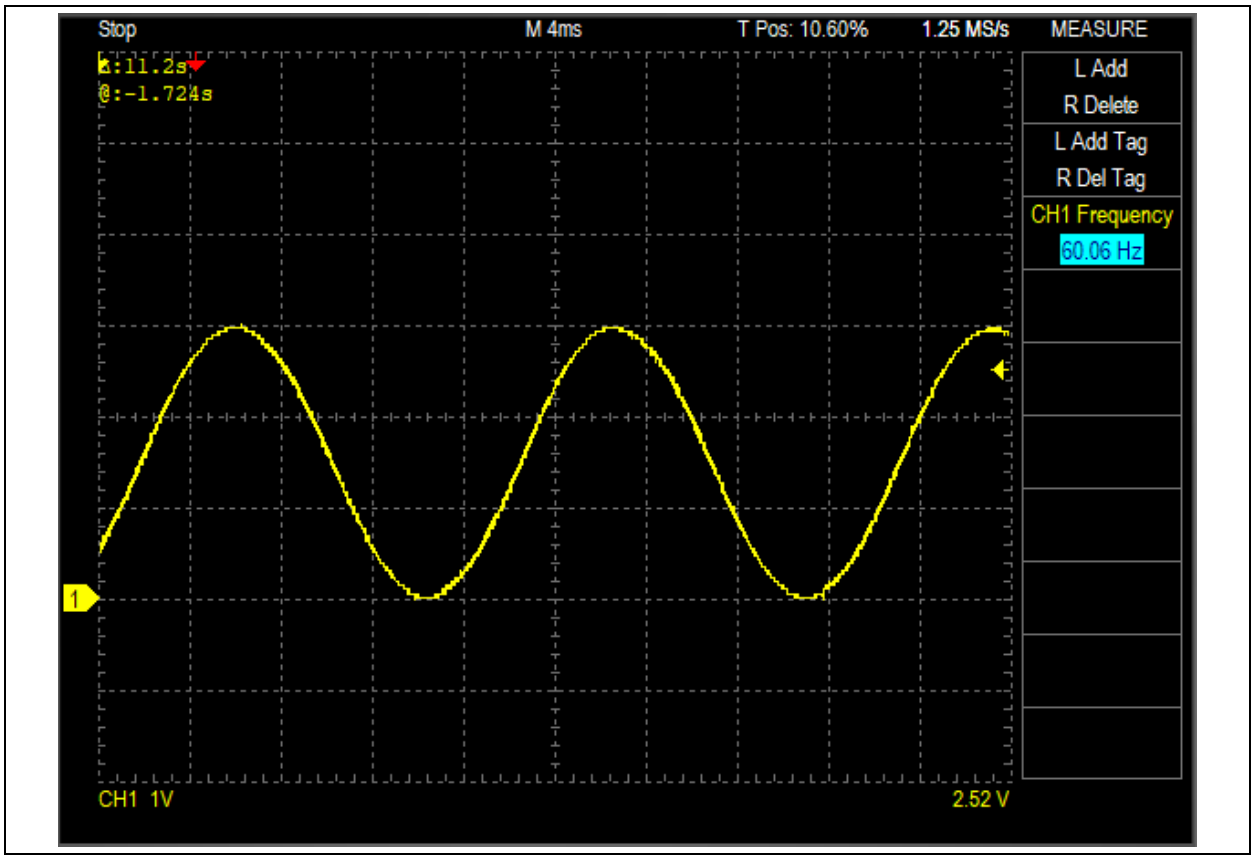
By using high frequency PWM and external RC filter, the output sine wave can be smoother. The PWM frequency formula shows below:

$$\text{PWM Frequency} = \frac{\text{Fsys}}{\text{PWM_PERIOD}}$$

According to formula, PWM_PERIOD will be 2400 when Fsys is 48 MHz and PWM frequency is 20 kHz. Then user can use 100 Ω resistance and 4.7 μF capacitance as external RC filter for the PWM channel output.

1.3 Demo Result

This example code uses M031 PWM0 Channel 0 (PA.5) to generate 60 Hz sine wave. The waveform is shown below picture, which CH1 is PWM0 Channel 0:



2 Code Description

Calculate the lookup table for PWM duty value based on sine value and PWM period value.

```
    /* Create sine table for lookup */
    for (i = 0; i < 360; i++)
        /* Add 1.0 to offset sine result from [-1, 1] to [0, 2],
        and divide with 2.0 to compress to [0, 1] */
        g_au16Sine[i] = ((sin((i * Pi_Value) / 180.0) + 1.0) / 2.0) * PWM_PERIOD;
```

Configure Timer0 and enable interrupt to change PWM duty value.

```
    /* Set timer frequency to 21600Hz */
    TIMER_Open(TIMER0, TIMER_PERIODIC_MODE, TMR0_FREQ);
    /* Enable timer interrupt */
    TIMER_EnableInt(TIMER0);
    NVIC_EnableIRQ(TMR0_IRQn);
```

Configure PA5 as PWM0 Channel 0 output and set PWM frequency in 20 kHz.

```
    /* Set PA5 multi-function pin for PWM0 Channel 5 */
    SYS->GPA_MFPL = (SYS->GPA_MFPL & ~(SYS_GPA_MFPL_PA5MFP_Msk)) |
                    (SYS_GPA_MFPL_PA5MFP_PWM0_CH0);
    …
    …
    …
    /* PWM0 channel 0 frequency and duty configuration are as follows */
       PWM_ConfigOutputChannel(PWM0, 0, PWM_FREQ, 50);
    /* Enable output of PWM0 channel 0 */
       PWM_EnableOutput(PWM0, 0x01);
```

Start Timer0 and PWM.

```
    /* Start Timer0 counter */
    TIMER_Start(TIMER0);
    /* Start PWM counter */
    PWM_Start(PWM0, 0x01);
```

In Timer0 interrupt handler, change PWM duty and move pointer to the next value. If pointer is equal to 360, reset it to 0. Then start PWM again.

```
void TMR0_IRQHandler(void)
{
    if(TIMER_GetIntFlag(TIMER0) == 1)
    {
        /* Clear Timer0 time-out interrupt flag */
        TIMER_ClearIntFlag(TIMER0);
```

```
        /* Set PWM new setting according to sine phase from lookup table */
        PWM_SET_CMR(PWM0, 0, g_au16Sine[g_u16Sine_Count]);

        /* Add sine array pointer */
        g_u16Sine_Count++;

        /* Reset sine array pointer */
        if (g_u16Sine_Count == 360)
        {
            g_u16Sine_Count = 0;
        }
    }
}
```

3 Software and Hardware Environment

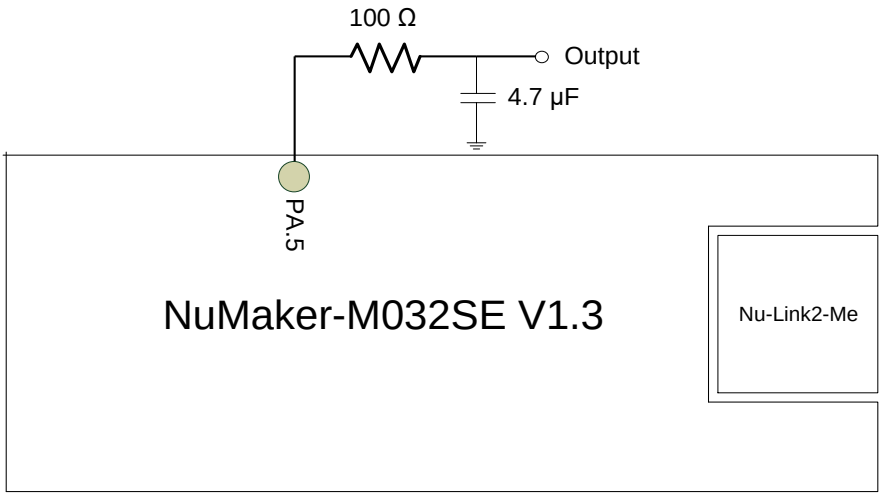
Software Environment

- BSP version
  - M031_Series_BSP_CMSIS_V3.02.000
- IDE version
  - Keil uVersion 5.28

Hardware Environment

- Circuit components
  - NuMaker-M032SE V1.3
- Diagram

M031 uses PWM0 Channel 0 (PA.5) to generate 60 Hz sine wave. There should also be a 100 Ω resistance and 4.7 μF capacitance as external RC filter for each PWM channel output.



4 Directory Information

- EC_M031_PWM_Sine_Wave_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. This project supports Keil uVersion 5.28 and higher version
2. Browsing into sample code folder by Directory Information (section 4) and double click EC_M031_PWM_Sine_Wave.uvprojx
3. Enter the Keil compile mode window
    a. Build the project
    b. Download
    c. Start / Stop the debug mode
4. Enter debug mode
    a. Run

6 Revision History

Date	Revision	Description
Nov. 22, 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.