Three Phase Sine Wave

Example Code Introduction for 8-bit 8051 NuMicro® Family

Information

Application	This code uses MS51 PWM Channel 0 (P1.2), Channel 1 (P1.1) and Channel 2 (P1.0) to generate 60 Hz three phase sine wave for application.
BSP Version	MS51_Series_BSP_Keil_V1.00.002
Hardware	NuTiny-MS51FB V1.1

1 Function Description

1.1 Introduction

This example code uses MS51 PWM Channel 0 (P1.2), Channel 1 (P1.1) and Channel 2 (P1.0) to generate 60 Hz 3 phase sine wave for application, such as motor control. 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. To make three PWM channels have 120 degree offset, PWM Channel 0 begins from 1st point, PWM Channel 1 begins from 121th point and PWM Channel 1 begins from 241th point.

The example code also uses Timer0 and enables Timer0 interrupt to change PWM duty value. The Timer0 uses mode 2 to make Timer 0 auto-reload timer value for periodic interrupt. The time-out period formula shows below:

$$\text{Time-out Period} = \frac{\text{0xFF - TIMER0\_Value}}{\text{Clock Frequency}} = \frac{1}{\text{Sine Wave Frequency} * \text{Sine Wave Resolution}}$$

According to formula, TIMER0_Value will be 0xC2 when clock frequency is 16 MHz with divider 12, sine wave frequency is 60 Hz and sine wave resolution is 360.

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

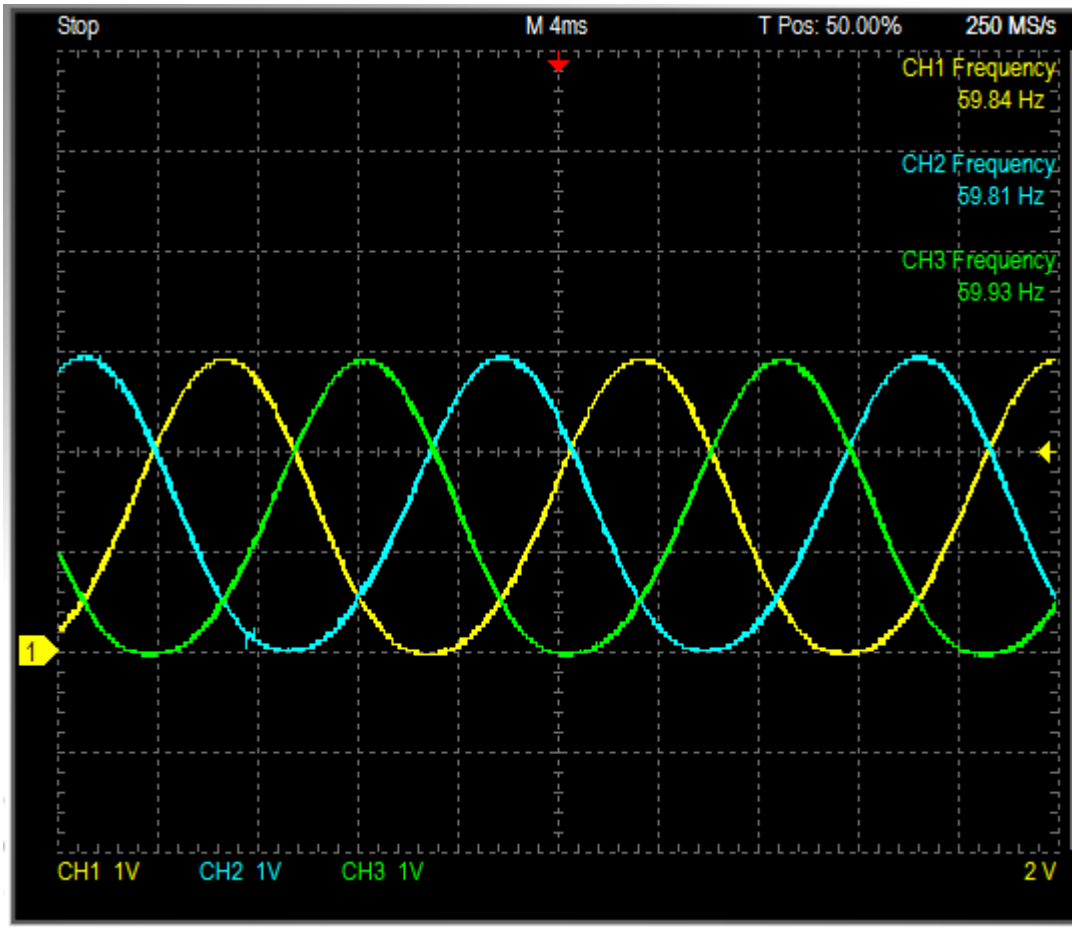
$$\text{PWM Frequency} = \frac{\text{Fsys}}{\text{PWM\_Period}}$$

According to formula, PWM_Period will be 800 when Fsys is16 MHz and PWM frequency is 20 kHz. Then user can use 100 Ω resistance and 4.7 μF capacitance as external RC filter for each PWM channel output.

1.3 Demo Result

This example code uses MS51 PWM Channel 0 (P1.2), Channel 1 (P1.1) and Channel 2 (P1.0) to generate 60 Hz 3 phase sine wave.

The waveform is shown below picture, which CH1 is PWM Channel 0, CH2 is PWM Channel 1, CH3 is PWM Channel 2:



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.

```
    /* Enable Timer0 Mode 2 (8-bit Timer Auto-reload) */
    TIMER0_MODE2_ENABLE;
    TIMER0_FSYS_DIV12;
    /* Set Timer0 counter value */
    TH0 = TL0 = TIMER0_VALUE;
    /* Enable Timer0 interrupt */
    ENABLE_TIMER0_INTERRUPT;
    /* Enable Global Interrupt */
    ENABLE_GLOBAL_INTERRUPT;
```

Configure PWM Channel 0, Channel 1 and Channel 2 output and set PWM frequency in 20 kHz. In addition, set PWM Channel 0 begins from 1st point, PWM Channel 1 begins from 121th point and PWM Channel 1 begins from 241th point.

```
UINT16 data g_u16Sine_Count_u = 0;
UINT16 data g_u16Sine_Count_v = 120;
UINT16 data g_u16Sine_Count_w = 240;

    /* Set GPIO P1.2, P1.1 and P1.0 as Push-pull mode */
    P12_PUSHPULL_MODE;
    P11_PUSHPULL_MODE;
    P10_PUSHPULL_MODE;
    /* Enable PWM Channel 0, Channel 1 and Channel 2 output */
    PWM0_P12_OUTPUT_ENABLE;
    PWM1_P11_OUTPUT_ENABLE;
    PWM2_P10_OUTPUT_ENABLE;
    /* Set PWM as independent mode */
    PWM_IMDEPENDENT_MODE;
    /* Set PWM Period */
    PWMPH = HIBYTE(PWM_Period);
    PWMPL = LOBYTE(PWM_Period);
    /* Set PWM Channel 0 Duty */
```

```
    PWM0H = HIBYTE(g_au16Sine[g_u16Sine_Count_u]);
    PWM0L = LOBYTE(g_au16Sine[g_u16Sine_Count_u]);
    /* Set PWM Channel 1 Duty */
    PWM1H = HIBYTE(g_au16Sine[g_u16Sine_Count_v]);
    PWM1L = LOBYTE(g_au16Sine[g_u16Sine_Count_v]);
    /* Set PWM Channel 2 Duty */
    PWM2H = HIBYTE(g_au16Sine[g_u16Sine_Count_w]);
    PWM2L = LOBYTE(g_au16Sine[g_u16Sine_Count_w]);
    /* Load PWM setting */
    set_PWMCON0_LOAD;
```

Start Timer0 and PWM.

```
    /* Start Timer0 */
    set_TCON_TR0;
    /* Start PWM */
    set_PWMCON0_PWMRUN;
```

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 Timer0_ISR(void) interrupt 1
{
    /* Set PWM new setting according to sine phase from lookup table */
    PWM0H = HIBYTE(g_au16Sine[g_u16Sine_Count_u]);
    PWM0L = LOBYTE(g_au16Sine[g_u16Sine_Count_u]);
    PWM1H = HIBYTE(g_au16Sine[g_u16Sine_Count_v]);
    PWM1L = LOBYTE(g_au16Sine[g_u16Sine_Count_v]);
    PWM2H = HIBYTE(g_au16Sine[g_u16Sine_Count_w]);
    PWM2L = LOBYTE(g_au16Sine[g_u16Sine_Count_w]);

    /* Add sine array pointer */
    g_u16Sine_Count_u++;
    g_u16Sine_Count_v++;
    g_u16Sine_Count_w++;

    /* Reset sine array pointer */
    if (g_u16Sine_Count_u == 360)
        g_u16Sine_Count_u = 0;

    if (g_u16Sine_Count_v == 360)
        g_u16Sine_Count_v = 0;

    if (g_u16Sine_Count_w == 360)
```

```
        g_u16Sine_Count_w = 0;

    /* Load PWM new setting and start PWM */
    set_PWMCON0_LOAD;
    set_PWMCON0_PWMRUN;
}
```

3 Software and Hardware Environment

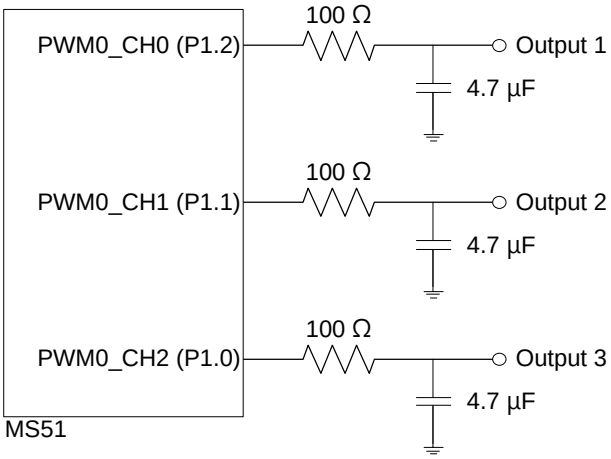
Software Environment

- BSP version
  - MS51 Series BSP Keil V1.00.002
- IDE version
  - Keil C51 V9.55

Hardware Environment

- Circuit components
  - NuTiny-MS51FB V1.1
- Diagram

MS51 uses PWM Channel 0 (P1.2), Channel 1 (P1.1) and Channel 2 (P1.0) to generate 60 Hz 3 phase 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_MS51_Three_Phase_Sine_Wave_V1.00
  - Library — Sample code header and source files
    - Device — Device associated header file
    - Startup — Startup code for classic 8051 devices
    - 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 MS51_Three_Phase_Sin_Wave.uvproj.
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
Oct 24, 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.