控制PWM风扇

NuMicro® 32 位系列微控制器范例代码介绍

文件信息

代码简述	使用Mini51的PWM通道0 (P2.2) 控制4线式PWM风扇的转速，并且使用Timer 0的外部捕获脚位 (P3.2) 侦测风扇的每分钟转速。
BSP 版本	Mini51DE_Series_BSP_CMSIS_v3.02.000
开发平台	NuTiny-SDK-Mini54 V3.1 4线式PWM风扇

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 功能介绍

1.1 简介

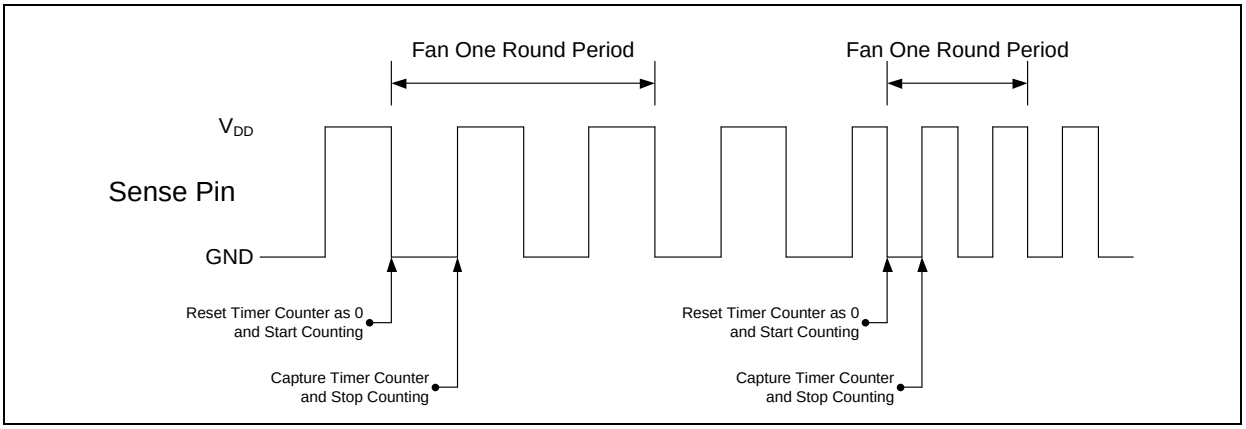
使用Mini51的PWM通道0 (P2.2) 控制4线式PWM风扇的转速，并且使用Timer 0的外部捕获脚位 (P3.2) 侦测风扇的每分钟转速。输入PWM波形到PWM风扇的control脚位，可以控制风扇的转速。藉由改变PWM的占空比，PWM风扇可以运作在适合的转速。PWM风扇的sense脚位会输出转速器讯号。此讯号在风扇的马达转动到下一个磁极时，会变更输出电平。藉由计算风扇转一圈需要的时间，使用者可以得到风扇的每分钟转速。

在使用这个范例程序前，用户需要将PWM通道0 (P2.2) 接到PWM风扇的control脚位，并且将Timer 0的外部捕获脚位 (P3.2) 接到风扇的sense脚位。

1.2 原理

为了在不同的情境下运作在适合的转速，4线式PWM风扇具有control脚位，可以接收PWM波形，并用来控制MOSFET开关。当增加PWM的占空比时，MOSFET开关开启越长，而风扇转速越快。基于Intel释出的规格，PWM输出频率应为25 kHz。

使用者也可以得到目前的风扇转速。4磁极马达的风扇在sense脚位所输出的转速器波形，如下所示：



当风扇马达转动到下一个磁极时，转速器波形会变更电平。使用者可以将Timer 0的外部捕获脚位 (T0EX) 接到风扇的sense脚位，并且使能触发计数捕获模式以及电平转换触发。当T0EX侦测到由高电平转换到低电平时，Timer 0会将计数器归零，并且开始计数。当T0EX侦测到由低电平转换到高电平时，Timer 0会停止计数，并且将计数器的值捕获到Timer Capture Data Register (TCAP)。由于PWM风扇的sense脚位是open-drain类型，使用者需要在此脚位使用上

拉电阻。

使用如下所示的计算公式，使用者可以计算出风扇的每分钟转速。

$$风扇的每分钟转速 = \frac{Timer\ 0\ 时钟源频率 * 60}{(马达磁极数) * (Timer\ 0\ 捕获数值)}$$

1.3 执行结果

范例程序首先使用100%占空比的PWM波形，侦测风扇的最快转速。然后使用0%占空比的PWM波形，侦测风扇的最慢转速。每次改变PWM占空比时，需要等待3秒钟的时间，确保风扇已经稳定。在完成上述动作后，使用者可以输入更多的动作，控制PWM占空比增加10%或1%，或是减少10%或1%。

结果如下图所示：

```
CPU @ 22118400 Hz
+------------------------------------------------------------------+
|    4-wire PWM Fan Control Sample Code                            |
|    Use PWM Channel 0 to control fan speed                        |
|    and use Timer 0 Capture mode to detect fan speed in RPM.      |
|    Please connect PWM Channel 0 (P2.2) to fan control pin        |
|    and connect Timer 0 external capture pin (P3.2) to fan        |
|    sense pin.                                                    |
+------------------------------------------------------------------+
The maximum fan speed is 1440 RPM.
The mimimum fan speed is 660 RPM.


Current PWM duty is 60%, please enter your action.
(1: Increase 10%; 2: Increase 1%; 3: Decrease 10%; 4: Decrease 1%)
1
Changing... Current fan speed is 1140 RPM.


Current PWM duty is 70%, please enter your action.
(1: Increase 10%; 2: Increase 1%; 3: Decrease 10%; 4: Decrease 1%)
_
```

2 代码介绍

初始化Timer 0并且使能触发计数捕获模式信道0捕获功能以及电平转换触发

```
void TMR0_Init(void)
{
    /*---------------------------------------------------------------------------------------------*/
    /* Init Timer 0                                                                                */
    /*---------------------------------------------------------------------------------------------*/
    /* Reset IP */
    SYS_ResetModule(TMR0_RST);

    /* Give a dummy target frequency here. Will over write capture resolution with macro
*/
    TIMER_Open(TIMER0, TIMER_PERIODIC_MODE, 1000000);

    /* Update prescale to set proper resolution */
    TIMER_SET_PRESCALE_VALUE(TIMER0, 0);

    /* Set compare value as large as possible, so don't need to worry about counter
overrun too frequently */
    TIMER_SET_CMP_VALUE(TIMER0, 0xFFFFFF);

    /* Configure Timer 0 trigger counting mode and level change trigger.
       The high to low transition on Timer 0 external capture pin is detected to
       reset TDR as 0 and then starts counting, while low to high transition stops
       counting. */
    TIMER_EnableCapture(TIMER0, TIMER_CAPTURE_TRIGGER_COUNTING_MODE,
TIMER_CAPTURE_FALLING_THEN_RISING_EDGE);

    /* Enable T0EX debounce function */
    TIMER0->TEXCON |= TIMER_TEXCON_TEXDB_Msk;

    /* Enable timer interrupt */
    TIMER_EnableCaptureInt(TIMER0);
    NVIC_EnableIRQ(TMR0_IRQn);
}
```

初始化PWM信道0的频率为25 kHz。

```
void PWM_Init(void)
{
    /*---------------------------------------------------------------------------------------------*/
```

```
    /* Init PWM Channel 0                                                     */
    /*------------------------------------------------------------------------*/
    /* Reset IP */
    SYS_ResetModule(PWM_RST);

    /* Set PWM Channel 0 frequency to 25 kHz and initial duty 100% */
    PWM_ConfigOutputChannel(PWM, 0, 25000, 100);

    /* Enable PWM Channel 0 output */
    PWM_EnableOutput(PWM, BIT0);
}
```

在Timer 0的中断处理中，停止Timer 0，并且计算风扇转速。

```
void TMR0_IRQHandler(void)
{
    uint32_t u32_FanRPM;

    /* Stop Timer 0 */
    TIMER_Stop(TIMER0);

    /* Calculate RPM = (Timer 0 Clock Source Frequency / (Fan_Pole * Timer 0 Capture
Value)) * 60 */
    u32_FanRPM = (22118400 / (Fan_Pole * TIMER_GetCaptureData(TIMER0))) * 60;
    printf("%d RPM.\n", u32_FanRPM);

    /* Clear Timer 0 Capture interrupt flag */
    TIMER_ClearCaptureIntFlag(TIMER0);

    /* Set Detect done flag*/
    g_u8DetectFlag = 1;
}
```

测试风扇的最快和最慢转速。

```
    /* Test maximum fan speed */
    /* Start PWM Channel 0 */
    PWM_Start(PWM, BIT0);
    /* Wait 3 seconds */
    Delay_mS(3000);
    /* Start Timer 0 to detect the time which fan takes when rotating one round */
    g_u8DetectFlag = 0;
    printf("The maximum fan speed is ");
    TIMER_Start(TIMER0);
```

```
    while (!g_u8DetectFlag);

    /* Test maximum fan speed */
    /* Force PWM Channel 0 output low */
    PWM->PHCHG = PWM->PHCHGNXT & ~(PWM_PHCHGNXT_PWM0_Msk | PWM_PHCHGNXT_D0_Msk);
    /* Wait 3 seconds */
    Delay_mS(3000);
    /* Start Timer 0 to detect the time which fan takes when rotating one round */
    g_u8DetectFlag = 0;
    printf("The mimimum fan speed is ");
    TIMER_Start(TIMER0);

    while (!g_u8DetectFlag);
```

根据使用者的输入动作，增加或减少PWM的占空比。

```
    while (1)
    {
        printf("\n\nCurrent PWM duty is %d%%, please enter your action.\n", g_u8PWMDuty);
        printf("(1: Increase 10%%; 2: Increase 1%%; 3: Decrease 10%%; 4: Decrease
1%%)\n");
        u8Action = getchar();
        printf("%c\n", u8Action);

        switch (u8Action)
        {
            case '1':
            {
                /* Increase PWM duty 10% */
                (g_u8PWMDuty > 90) ? (g_u8PWMDuty = 100) : (g_u8PWMDuty += 10);

                break;
            }

            case '2':
            {
                /* Increase PWM duty 1% */
                (g_u8PWMDuty > 99) ? (g_u8PWMDuty = 100) : (g_u8PWMDuty++);

                break;
            }
```

```
            case '3':
            {
                /* Decrease PWM duty 10% */
                (g_u8PWMDuty < 10) ? (g_u8PWMDuty = 0) : (g_u8PWMDuty -= 10);

                break;
            }

            case '4':
            {
                /* Decrease PWM duty 1% */
                (g_u8PWMDuty < 1) ? (g_u8PWMDuty = 0) : (g_u8PWMDuty--);

                break;
            }

            default:
            {
                continue;
            }
        }

        /* Set PWM Channel 0 frequency and duty */
        if (g_u8PWMDuty > 0)
        {
            PWM->PHCHG |= PWM_PHCHGNXT_PWM0_Msk;
            PWM_ConfigOutputChannel(PWM, 0, 25000, g_u8PWMDuty);
        }
        else
            PWM->PHCHG = PWM->PHCHGNXT & ~(PWM_PHCHGNXT_PWM0_Msk | PWM_PHCHGNXT_D0_Msk);

        /* Wait 3 seconds */
        printf("Changing... ");
        Delay_mS(3000);
        /* Start Timer 0 to detect the time which fan takes when rotating one round */
        g_u8DetectFlag = 0;
        printf("Current fan speed is ");
        TIMER_Start(TIMER0);

        while (!g_u8DetectFlag);
    }
```

3 软件与硬件环境

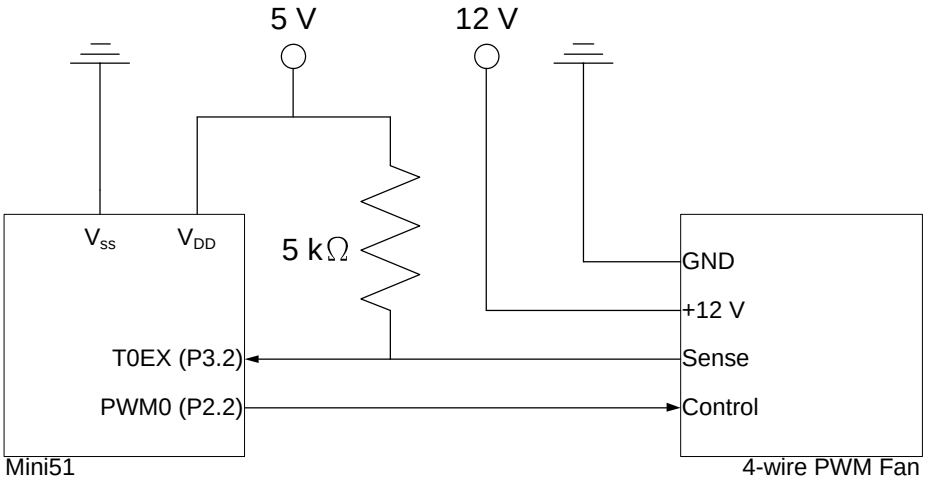
● 软件环境

- ■ BSP 版本
  - ◆ Mini51DE Series BSP CMSIS v3.02.000
- ■ IDE 版本
  - ◆ Keil uVersion 5.26

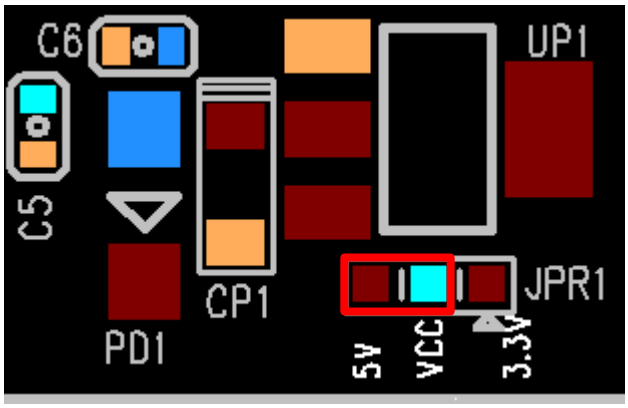
● 硬件环境

- ■ 电路组件
  - ◆ NuTiny-SDK-Mini54 V3.1
  - ◆ 4 线式 PWM 风扇
- ■ 示意图

将Mini51的PWM通道0 (P2.2) 连接到4线式PWM风扇的control脚位，控制风扇转速。并将Timer 0的外部捕获脚位 (T0EX) (P3.2) 连接到4线式PWM风扇的sense脚位，侦测风扇的每分钟转速，并且使用5 kΩ的上拉电阻。



为了使NuTiny-SDK-Mini54运作在5 V，使用者需要将Nu-Link-Me的JPR1上的0 Ω电阻，从3.3 V改到5 V。


C6
C5
CP1
PD1
UP1
JPR1
5V
VCC
3.3V

4 目录信息

- EC_Mini51_PWM_Fan_Control_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 如何执行范例程序

1. 根据目录信息章节进入 ExampleCode 路径中的 KEIL 文件夹，双击 Mini51_PWM_Fan_Control.uvproj。
2. 进入编译模式接口
   a. 编译
   b. 下载代码至内存
   c. 进入 / 离开除错模式
3. 将 12 V 电源供应连接到 4 线式 PWM 风扇。
4. 将 4 线式 PWM 风扇的 control 和 sense 脚位连接到 NuTiny-SDK-Mini54。
5. 进入除错模式接口
   a. 执行代码
6. 使用终端机软件，获得风扇转速讯息，并且输入动作来改变目前的 PWM 占空比。

6 修订纪录

Date	Revision	Description
July 25, 2019	1.00	1. 初始发布.

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.