控制PWM風扇

NuMicro® 32 位系列微控制器範例代碼介紹

文件資訊

代碼簡述	使用 Mini51 的 PWM 通道 0 (P2.2) 控制 4 線式 PWM 風扇的轉速，並且使用 Timer 0 的外部捕獲腳位 (P3.2) 偵測風扇的每分鐘轉速。
BSP 版本	Mini51DE_Series_BSP_CMSIS_v3.02.000
開發平台	NuTiny-SDK-Mini54 V3.1 4 線式 PWM 風扇

1 功能介紹

1.1 簡介

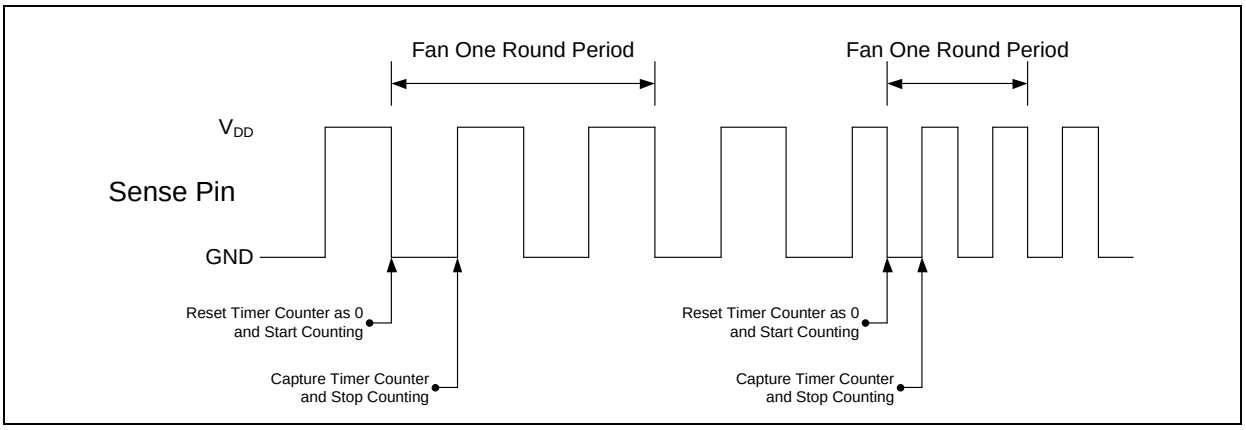
使用Mini51的PWM通道0 (P2.2) 控制4線式PWM風扇的轉速，並且使用Timer 0的外部捕獲腳位 (P3.2) 偵測風扇的每分鐘轉速。輸入PWM波形到PWM風扇的control腳位，可以控制風扇的轉速。藉由改變PWM的佔空比，PWM風扇可以運作在適合的轉速。PWM風扇的sense腳位會輸出轉速器訊號。此訊號在風扇的馬達轉動到下一個磁極時，會變更輸出電平。藉由計算風扇轉一圈需要的時間，使用者可以得到風扇的每分鐘轉速。

在使用這個範例程式前，使用者需要將PWM通道0 (P2.2) 接到PWM風扇的control腳位，並且將Timer 0的外部捕獲腳位 (P3.2) 接到風扇的sense腳位。

1.2 原理

為了在不同的情境下運作在適合的轉速，4線式PWM風扇具有control腳位，可以接收PWM波形，並用來控制MOSFET開關。當增加PWM的佔空比時，MOSFET開關開啟越長，而風扇轉速越快。基於Intel釋出的規格，PWM輸出頻率應為25 kHz。

使用者也可以得到目前的風扇轉速。4磁極馬達的風扇在sense腳位所輸出的轉速器波形，如下所示：



當風扇馬達轉動到下一個磁極時，轉速器波形會變更電平。使用者可以將Timer 0的外部捕獲腳位 (T0EX) 接到風扇的sense腳位，並且使能觸發計數捕獲模式以及電平轉換觸發。當T0EX偵測到由高電平轉換到低電平時，Timer 0會將計數器歸零，並且開始計數。當T0EX偵測到由低電平轉換到高電平時，Timer 0會停止計數，並且將計數器的值捕獲到Timer Capture Data Register (TCAP)。由於PWM風扇的sense腳位是open-drain類型，使用者需要在此腳位使用上

拉電阻。

使用如下所示的計算公式，使用者可以計算出風扇的每分鐘轉速。

$$風扇的每分鐘轉速 = \frac{Timer\ 0\ 時鐘源頻率 * 60}{(馬達磁極數) * (Timer\ 0\ 捕獲數值)}$$

1.3 執行結果

範例程式首先使用100%佔空比的PWM波形，偵測風扇的最快轉速。然後使用0%佔空比的PWM波形，偵測風扇的最慢轉速。每次改變PWM佔空比時，需要等待3秒鐘的時間，確保風扇已經穩定。在完成上述動作後，使用者可以輸入更多的動作，控制PWM佔空比增加10%或1%，或是減少10%或1%。

結果如下圖所示：

```
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 代碼介紹

初始化 Timer 0 並且使能觸發計數捕獲模式通道 0 捕獲功能以及電平轉換觸發

```
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 通道 0 的頻率為 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 的中斷處理中，停止 Timer 0，並且計算風扇轉速。

```
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;
}
```

測試風扇的最快和最慢轉速。

```
    /* 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);
```

根據使用者的輸入動作，增加或減少 PWM 的佔空比。

```
    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 軟體與硬體環境

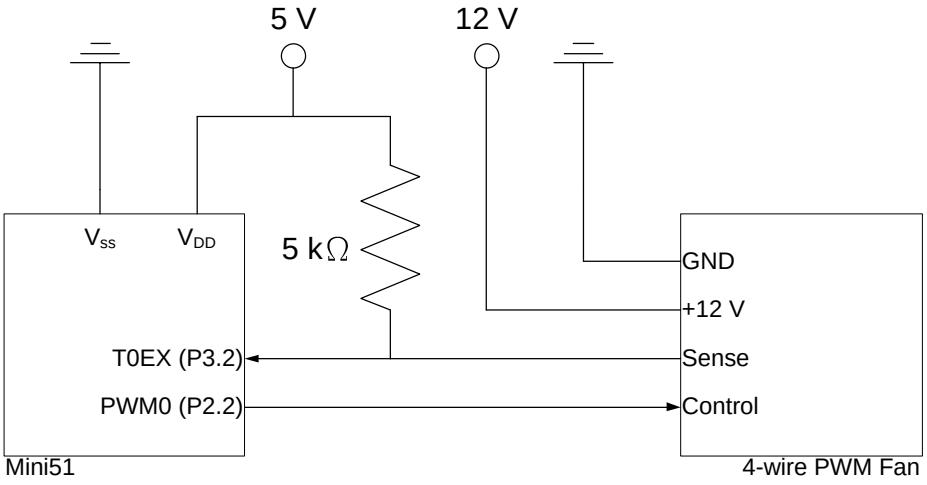
● 軟體環境

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

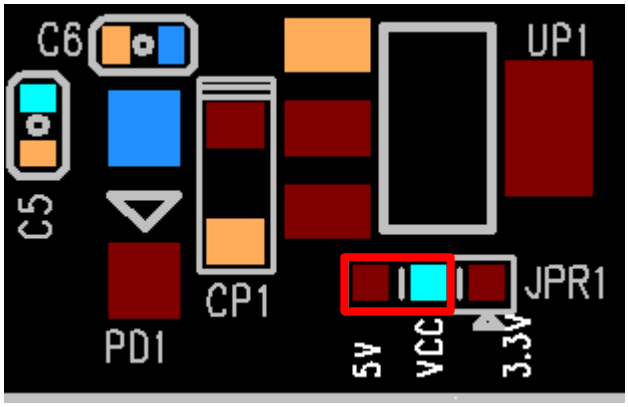
● 硬體環境

- ■ 電路元件
  - ◆ NuTiny-SDK-Mini54 V3.1
  - ◆ 4 線式 PWM 風扇
- ■ 示意圖

將 Mini51 的 PWM 通道 0 (P2.2) 連接到 4 線式 PWM 風扇的 control 腳位，控制風扇轉速。並將 Timer 0 的外部捕獲腳位 (T0EX) (P3.2) 連接到 4 線式 PWM 風扇的 sense 腳位，偵測風扇的每分鐘轉速，並且使用 5 kΩ的上拉電阻。



為了使 NuTiny-SDK-Mini54 運作在 5 V，使用者需要將 Nu-Link-Me 的 JPR1 上的 0 Ω電阻，從 3.3 V 改到 5 V。



4 目錄資訊

- 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 如何執行範例程式

1. 根據目錄資訊章節進入 ExampleCode 路徑中的 KEIL 資料夾，雙擊 Mini51_PWM_Fan_Control.uvproj。
2. 進入編譯模式介面
   a. 編譯
   b. 下載代碼至記憶體
   c. 進入 / 離開除錯模式
3. 將 12 V 電源供應連接到 4 線式 PWM 風扇。
4. 將 4 線式 PWM 風扇的 control 和 sense 腳位連接到 NuTiny-SDK-Mini54。
5. 進入除錯模式介面
   a. 執行代碼
6. 使用終端機軟體，獲得風扇轉速訊息，並且輸入動作來改變目前的 PWM 佔空比。

6 修訂紀錄

Date	Revision	Description
July 25, 2019	1.00	1. 初始發布.

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.