NANO100B Timer Trigger PDMA

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	Timer Trigger PDMA
BSP Version	NANO100B BSP CMSIS V3.03.001
Hardware	NANO100B-Tiny

1 Function Description

1.1 Introduction

NuMicro® NANO100B supports TIMER trigger PDMA. In NANO100B, the sample bsp\SampleCode\StdDriver\PDMA_memory shows memory to memory transfer. The sample Timer_Trigger_PDMA modifies BSP sample PDMA_memory to demonstrate TIMER0 to trigger PDMA channel 1 to do memory to memory transfer.

1.2 Principle

This sample enables TIMER0 and configures TIMER0, PDMA0 trigger source is set TIMER0. PDMA will also start PDMA channel 1 to do memory-to-memory transfer when TIMER0 is interrupted.

1.3 Demo Result

```
This sample code demonstrates Timer trigger PDMA channel 1 to memory to memmeoy.
 0 sec
 1 sec  Src[ 0]=2047,Dest[ 0]=2047 PASS
 2 sec  Src[ 1]=2251,Dest[ 1]=2251 PASS
 3 sec  Src[ 2]=2453,Dest[ 2]=2453 PASS
 4 sec  Src[ 3]=2651,Dest[ 3]=2651 PASS
 5 sec  Src[ 4]=2844,Dest[ 4]=2844 PASS
 6 sec  Src[ 5]=3028,Dest[ 5]=3028 PASS
 7 sec  Src[ 6]=3202,Dest[ 6]=3202 PASS
 8 sec  Src[ 7]=3365,Dest[ 7]=3365 PASS
 9 sec  Src[ 8]=3515,Dest[ 8]=3515 PASS
10 sec  Src[ 9]=3650,Dest[ 9]=3650 PASS
11 sec  Src[10]=3769,Dest[10]=3769 PASS
12 sec  Src[11]=3871,Dest[11]=3871 PASS
13 sec  Src[12]=3954,Dest[12]=3954 PASS
14 sec  Src[13]=4019,Dest[13]=4019 PASS
15 sec  Src[14]=4064,Dest[14]=4064 PASS
16 sec  Src[15]=4088,Dest[15]=4088 PASS
17 sec  Src[16]=4095,Dest[16]=4095 PASS
18 sec  Src[17]=4076,Dest[17]=4076 PASS
19 sec  Src[18]=4040,Dest[18]=4040 PASS
20 sec  Src[19]=3984,Dest[19]=3984 PASS
21 sec  Src[20]=3908,Dest[20]=3908 PASS
```

```
    22 sec  Src[21]=3813,Dest[21]=3813 PASS
    23 sec  Src[22]=3701,Dest[22]=3701 PASS
    24 sec  Src[23]=3573,Dest[23]=3573 PASS
    25 sec  Src[24]=3429,Dest[24]=3429 PASS
    26 sec  Src[25]=3272,Dest[25]=3272 PASS
    27 sec  Src[26]=3102,Dest[26]=3102 PASS
    28 sec  Src[27]=2921,Dest[27]=2921 PASS
    29 sec  Src[28]=2732,Dest[28]=2732 PASS
    30 sec  Src[29]=2536,Dest[29]=2536 PASS
    31 sec  Src[30]=2335,Dest[30]=2335 PASS
    32 sec  Src[31]=2132,Dest[31]=2132 PASS
```

2 Code Description

Function for TIMER0 IRQ, declared in `startup_nano100series.s`

```
static uint32_t sec = 0;
void TMR0_IRQHandler(void)
{
    if (sec > 0)
    {
        printf("%2d sec\t", sec);

        if (sec < sizeof(a32Sine) / sizeof(uint32_t))
            printf("Src[%2d]=%d,Dest[%2d]=%d", sec - 1, a32Sine[sec - 1], sec - 1,
a32Dest[sec - 1]);

        if (a32Sine[sec - 1] == a32Dest[sec - 1])
            printf(" PASS\n");
        else
            printf(" FAIL\n");

        sec++;
    }
    else
        printf("%2d sec\n", sec++);

    /* clear timer interrupt flag */
    TIMER_ClearIntFlag(TIMER0);
}
```

Function for PDMA IRQ, declared in `startup_nano100series.s`

```
void PDMA_IRQHandler(void)
{
    uint32_t status = PDMA_GET_INT_STATUS();

    if (status & 0x2)   /* CH1 */
    {
        if (PDMA_GET_CH_INT_STS(1) & 0x2)
            g_u32PdmaTDoneInt = 1;

        PDMA_CLR_CH_INT_FLAG(1, PDMA_ISR_TD_IS_Msk);
        printf("DMA1 over interrupt !!\n");
    }
    else if (status & 0x4)     /* CH2 */
    {
        if (PDMA_GET_CH_INT_STS(2) & 0x2)
            g_u32PdmaTDoneInt = 2;

        PDMA_CLR_CH_INT_FLAG(2, PDMA_ISR_TD_IS_Msk);
    }
    else if (status & 0x8)     /* CH3 */
    {
        if (PDMA_GET_CH_INT_STS(3) & 0x2)
            g_u32PdmaTDoneInt = 3;

        PDMA_CLR_CH_INT_FLAG(3, PDMA_ISR_TD_IS_Msk);
    }
    else if (status & 0x10)     /* CH4 */
    {
        if (PDMA_GET_CH_INT_STS(4) & 0x2)
            g_u32PdmaTDoneInt = 4;

        PDMA_CLR_CH_INT_FLAG(4, PDMA_ISR_TD_IS_Msk);
    }
    else if (status & 0x20)     /* CH5 */
    {
        if (PDMA_GET_CH_INT_STS(5) & 0x2)
            g_u32PdmaTDoneInt = 5;

        PDMA_CLR_CH_INT_FLAG(5, PDMA_ISR_TD_IS_Msk);
    }
    else if (status & 0x40)     /* CH6 */
    {
        if (PDMA_GET_CH_INT_STS(6) & 0x2)
            g_u32PdmaTDoneInt = 6;

        PDMA_CLR_CH_INT_FLAG(6, PDMA_ISR_TD_IS_Msk);
    }
    else
        printf("unknown interrupt !!\n");
}
```

MAIN function

```
int32_t main(void)
{

    /* Init System, IP clock and multi-function I/O
       In the end of SYS_Init() will issue SYS_LockReg()
       to lock protected register. If user want to write
       protected register, please issue SYS_UnlockReg()
       to unlock protected register if necessary */
    SYS_Init();

    /* Init UART0 for printf */
    UART_Open(UART0, 115200);

    printf("\nThis sample code demonstrate Timer trigger PDMA channel 1 to memory to 
memmeoy.\n");

    // Set timer frequency to 1Hz
    TIMER_Open(TIMER0, TIMER_PERIODIC_MODE, 1);
    /* Set Timer 0 time out to trigger PDMA */
    TIMER_SetTriggerSource(TIMER0, TIMER_TIMEOUT_TRIGGER);
    TIMER_SetTriggerTarget(TIMER0, TIMER_CTL_PDMA_TEEN_Msk);

    /* Enable TIMER0 interrupt */
    TIMER_EnableInt(TIMER0);
    NVIC_EnableIRQ(TMR0_IRQn);

    TIMER_ClearIntFlag(TIMER0);

    SYS_ResetModule(DMA_RST);
    /* Configure PDMA channel 1 to do memory-to-memory transfer */
    PDMA_Open(1 << 1);
    PDMA_SetTransferCnt(1, PDMA_WIDTH_32, sizeof(a32Sine) / sizeof(uint32_t));
    PDMA_SetTransferAddr(1, (uint32_t)a32Sine, PDMA_SAR_WRA, (uint32_t)a32Dest, 
PDMA_DAR_WRA);

    /* Set TIMER0 to trigger PDMA channel 1 */
    PDMA_SetTransferMode(1, PDMA_TMR0, 0, 0);

    PDMA_EnableInt(1, PDMA_IER_TD_IE_Msk);

    /* Enable PDMA IRQ */
    NVIC_EnableIRQ(PDMA_IRQn);
    PDMA_Trigger(1);
    TIMER_Start(TIMER0);

    while (1);
}
```

3 Software and Hardware Environment

Software Environment

- BSP version
  - NANO100B BSP CMSIS V3.03.001
- IDE version
  - Keil uVersion 5.26

Hardware Environment

- Circuit components
  - NANO100B-Tiny

4 Directory Information

🗁 EC_NANO100B_Timer_TriggerPDMA_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. Browsing into sample code folder by Directory Information (section 4) and double click Timer_Trigger_PDMA.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
June. 17, 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.