



Preliminary

CANTUS

- TIMER -

32bits EISC Microprocessor *CANTUS*

Ver 1.0
October 8, 2009

Advanced Digital Chips Inc.

History

2009-10-08 Created Preliminary Specification

CANTUS Evaluation Board Application Note : #0005 TIMER

©Advanced Digital Chips Inc.

All right reserved.

No part of this document may be reproduced in any form without written permission from Advanced Digital Chips Inc.

Advanced Digital Chips Inc. reserves the right to change in its products or product specification to improve function or design at any time, without notice.

Office

8th Floor, KookMin 1 Bldg.,
1009-5, Daechi-Dong, Gangnam-Gu, Seoul, 135-280, Korea
Tel : + 82-2-2107-5800
Fax : + 82-2-571-4890
URL : <http://www.adc.co.kr>

– Table of Contents –

1 SUMMARY.....6

2 REGISTER SET7

2.1 REGISTER SET FLOW CHART7

2.2 TIMER INTERRUPT SET8

2.3 TIMER INTERRUPT ENABLE8

2.4 TIMER PRE-SCALE CONTROL REGISTER8

2.5 TIMER CONTROL/PWM PERIOD REGISTER.....9

2.6 TIMER CONTROL REGISTER10

3 FUNCTION SET11

3.1 SETTIMER().....11

3.2 STOPTIMER().....12

3.3 TIMER.C13

4 POINT THIS NOTE.....14

– List of Figures –

그림 2-1 Register Set Flow Chart..... 7

– List of Tables –

<i>Table 2-1</i> Timer Pre-scale Control Register 0 (TP0CTRL).....	8
<i>Table 2-2</i> Timer Counter/PWM Period Register 0 (TM0CNT).....	9
<i>Table 2-3</i> Timer Control Register 0 (TM0CTRL).....	10

1 Summary

이 문서는 CANTUS의 Timer에 대한 Application Note이다.

CANTUS는 Timer/Counter, Capture, PWM 그리고 Output Compare 기능을 가진 32-Bit Timer/Counter를 8채널 내장하고 있다.

2 Register Set

2.1 Register Set Flow Chart

CANTUS의 Timer를 사용하기 위해선 다음과 같은 순서로 Register를 설정한다.

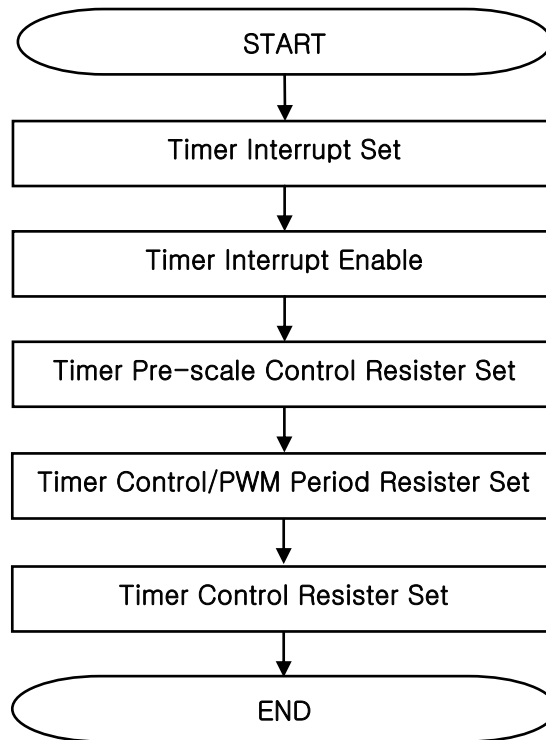


그림 2-1 Register Set Flow Chart

2.2 Timer Interrupt Set

Timer Interrupt를 사용하기 위해 Interrupt.c의

```
BOOL setinterrupt(INTERRUPT_TYPE intnum,void (*fp)());
```

를 호출하여 TIMER Ch0 Interrupt Vector Number 0x21에 ISR로 사용할 함수를 Setting 한다.

2.3 Timer Interrupt Enable

Interrupt Service Routine(ISR)이 Vector Table에 Setting되면 Interrupt를 Enable 한다.

```
void EnableInterrupt(INTERRUPT_TYPE num,BOOL b);
```

2.4 Timer Pre-Scale Control Register

Timer는 Pre-scaler에 의해 분주 된 Clock을 사용하며 Pre-scaler의 입력 Clock을 선택할 수 있다.

표 2-1 Timer Pre-scale Control Register 0 (TP0CTRL)

Address : 0x8002_1000

Bit	R/W	Description	Default Value
31 : 2	R	Reserved	-
1	R/W	CNTCLR : Pre-scale Counter and Timer Counter Reset When this bit is "1", the Timer Pre-scale and Counter will be reset.	0
0	R/W	CLKSEL : Pre-scale Clock Selection 0 : System clock 1 : CAPx	0

*** CAPx는 Timer 채널 별로 할당되어 있다.

2.5 Timer Control/PWM Period Register

Pre-scaler에 의해 분주 된 Clock을 사용하여 0x0 의 초기값에서 매 클럭마다 카운터 값을 “1” 씩 증가하여 사용자가 설정한 TM#CNT 레지스터 값에 도달하면 다시 0x0 이 되면서 인터럽트를 발생한다.

표 2-2 Timer Counter/PWM Period Register 0 (TM0CNT)

Address : 0x8002_1008

Bit	R/W	Description	Default Value
31 : 0	R/W	(Timer mode) - Write : Timer Counter Value - Read : Current Up-counter Value (PWM mode) - Read/Write : PWM Period Value	0xFFFF_FFFF

Timer 주기는 선택된 클럭, Pre-scaler 그리고 Timer Counter 에 의해 결정된다.

$$Timer\ Period = \frac{1}{Clock\ Source\ Frequency} \times \frac{1}{Pre - scaler\ Factor} \times (TMCNT) [sec] \quad \{TMCNT \geq 3\}$$

$$Timer\ Period = \frac{1}{Clock\ Source\ Frequency} \times \frac{1}{Pre - scaler\ Factor} \times (TMCNT + 1) [sec] \quad \{TMCNT < 3\}$$

Timer Period Example :

- Clock Source Frequency : 12MHz System Clock
- Pre-scaler Factor : 1 / 1024
- Timer Counter Value (TMCNT) : 1000

$$\Rightarrow 1/12MHz \times 1024 \times 1000 = 85.333msec = 11.718Hz$$

3 Function Set

3.1 settimer()

BOOL settimer(int nCh,U32 ms);

- Return : 올바른 Timer set일 때 True
- Int nCh : Timer Channel Number
- U32 ms : Timer Period

```

BOOL settimer(int nCh,U32 ms)
{
    if(nCh > 7)
        return FALSE;
    U32 r_tpcon = 0x80021000 + 0x20*nCh;
    U32 r_tmcon = 0x80021004 + 0x20*nCh;
    U32 r_tmcnt = 0x80021008 + 0x20*nCh;

    int intnum = nCh*4+1;
    int conval;

    long long int nanosecpertick = 1000000000/(GetAPBclock()/PRESCALE);
    long long int usec = (long long int)ms*1000;
    long long int cnt = ((long long int)usec*1000) / nanosecpertick;

    if(cnt > 0xffffffff) // ms is too big to set, adjust PRESACLE.
        return FALSE;
    //reset
    *(volatile U32*)r_tpcon = 1<<1;
    *(volatile U32*)r_tmcon = 0;

#ifdef PRESCALE==2
    conval = 0;
#elif PRESCALE==8
    conval = 1<<1;
#elif PRESCALE==32
    conval = 2<<1;
#elif PRESCALE==128
    conval = 3<<1;
#elif PRESCALE==512
    conval = 4<<1;
#elif PRESCALE==2048
    conval = 5<<1;
#elif PRESCALE==8192
    conval = 6<<1;
#elif PRESCALE==32768
    conval = 7<<1;
#else
#error "Invalid PRESCALE value"
#endif

    *(volatile U32*)r_tmcnt = cnt;
    *(volatile U32*)r_tmcon = conval|1;
    EnableInterrupt(intnum,TRUE);
    return TRUE;
}

```

settimer() 함수를 호출 함으로써 Timer 관련 Register를 설정하고 Interrupt를 Enable한다.

3.2 stoptimer()

BOOL stoptimer (int nCh);

- int n : 유효한 Channel TRUE/ 유효하지 않는 Channel FALSE
- U32 nCh : Timer Channel Number

```
BOOL stoptimer(int nCh)
{
    if(nCh>7)
        return FALSE;
    U32 r_tmcon = 0x80021004 + 0x20*nCh;
    *(volatile U32*)r_tmcon = 0;//stop
    return TRUE;
}
```

3.3 timer.c

/Cantuslib/timer.c

timer.c는 Timer를 사용하기 위한 정의 및 함수로 구성되어 있다. 사용자는 Timer의 Register들을 직접 사용하지 않아도 timer.c에서 정의된 함수를 사용하여 Timer를 설정하여 사용할 수 있다.

4 Point This Note

- CANTUS의 Timer는 15bit Pre-scaler를 내장한 32 Bit Timer이다.
- CANTUS의 Timer/Counter Clock은 system clock과 external clock 중 하나를 선택하고 15 bit Pre-scaler를 통해 구해진다.