



Preliminary

CANTUS

- SEGMENT -

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 : #0006 SEGMENT

©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 PORT ALTERNATE FUNCTION REGISTER8

2.3 TIMER INTERRUPT SET9

2.4 TIMER INTERRUPT ENABLE9

2.5 SEGMENT DISPLAY9

3 FUNCTION SET10

3.1 FUNCTION SET FLOW CHART.....10

3.1.1 SetSeg().....11

3.1.2 timer0isr();.....11

– List of Figures –

그림 2-1 Register Set Flow Chart..... 7
그림 3-1 Function Set Flow Chart 10

– List of Tables –

Table 2-1 Port 0 Alternate Function 8
Table 2-2 Port Alternate Function 0 Register (PAF0) 8

1 Summary

이 문서는 CANTUS SDK의 Segment에 대한 Application Note이다.

Segment Project는 CANTUS의 Address와 Data Bus를 이용하여 Quad-Digit Segment를 Timer 0번으로 Counting 하는 예제이다.

- TIMER는 AN_0005_TIMER를 참조하라.

2 Register Set

2.1 Register Set Flow Chart

Evaluation Board의 Quad-Digit Segment를 사용하기 위해선 다음과 같은 순서로 Register를 설정한다.

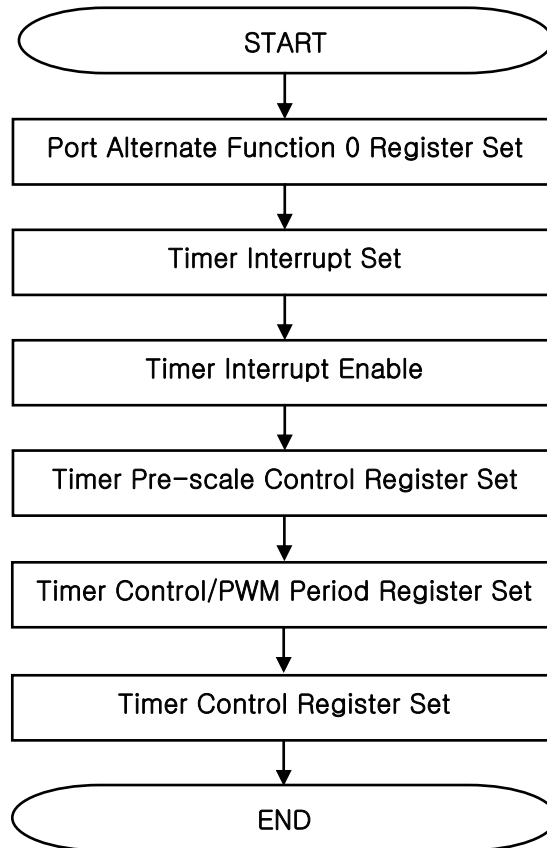


그림 2-1 Register Set Flow Chart

2.2 Port Alternate Function Register

Segment는 PIN 76~79, 82~85번을 Address로 사용한다. 이들 PIN은 아래 표 2-1과 같이 다른 Function을 공유하고 있어 Segment를 사용하기 위해서는 Port Alternate Function 0 Register에서 3rd로 설정하여야 한다.¹

Segment Project는 PIN 76~79, 82~85번을 AD[0]~AD[7]로 설정한다.

표 2-1 Port 0 Alternate Function

Group	Index	Pin	1 st	2 nd	3 rd	4 th (default)
PAF0	0	76			AD[0]	P0.0
	1	77			AD[1]	P0.1
	2	78			AD[2]	P0.2
	3	79			AD[3]	P0.3
	4	82			AD[4]	P0.4
	5	83			AD[5]	P0.5
	6	84			AD[6]	P0.6
	7	85			AD[7]	P0.7

표 2-2 Port Alternate Function 0 Register (PAF0)

Address : 0x8002_0020

Bit	R/W	Description	Default Value
31 : 16	R	Reserved	-
15 : 14	R/W	P0.7 : P0.7 Port Selection bit 00 : Reserved 01 : Reserved 10 : AD[7] 11 : P0.7	11
13 : 12	R/W	P0.6 : P0.6 Port Selection bit 00 : Reserved 01 : Reserved 10 : AD[6] 11 : P0.6	11
11 : 10	R/W	P0.5 : P0.5 Port Selection bit 00 : Reserved 01 : Reserved 10 : AD[5] 11 : P0.5	11
9 : 8	R/W	P0.4 : P0.4 Port Selection bit 00 : Reserved 01 : Reserved 10 : AD[4] 11 : P0.4	11
7 : 6	R/W	P0.3 : P0.3 Port Selection bit 00 : Reserved 01 : Reserved 10 : AD[3] 11 : P0.3	11
5 : 4	R/W	P0.2 : P0.2 Port Selection bit 00 : Reserved 01 : Reserved 10 : AD[2] 11 : P0.2	11
3 : 2	R/W	P0.1 : P0.1 Port Selection bit 00 : Reserved 01 : Reserved 10 : AD[1] 11 : P0.1	11
1 : 0	R/W	P0.0 : P0.0 Port Selection bit 00 : Reserved 01 : Reserved 10 : AD[0] 11 : P0.0	11

¹ CANTUS Datasheet의 8 GPIO 참조.

2.3 Timer Interrupt Set

Segment의 Common은 일정 주기로 번갈아 Select 된다. 이를 위해 Timer 0번을 사용하며, 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.

2.4 Timer Interrupt Enable

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

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

Segment에서는 timer.c의

```
BOOL settimer(int nCh,U32 ms);
```

를 호출할 때 nCh에 Timer 번호와 ms에 주기를 전달함으로써 Interrupt Enable 된다.

2.5 Segment Display

Segment의 Display는 Timer 0번의 Interrupt Service Routine에서 각 Digit을 Enable 하고 자리에 맞는 수를 표시 함으로써 이루어 진다.

즉, Timer Interrupt의 주기에 맞춰 Segment가 Display된다. main()함수의 while loop에서는 숫자를 1씩 증가시키고, 각 자리에 들어갈 수를 계산하여 대입하면 ISR에서 이를 읽어 Display 하게 된다.

3 Function Set

3.1 Function Set Flow Chart

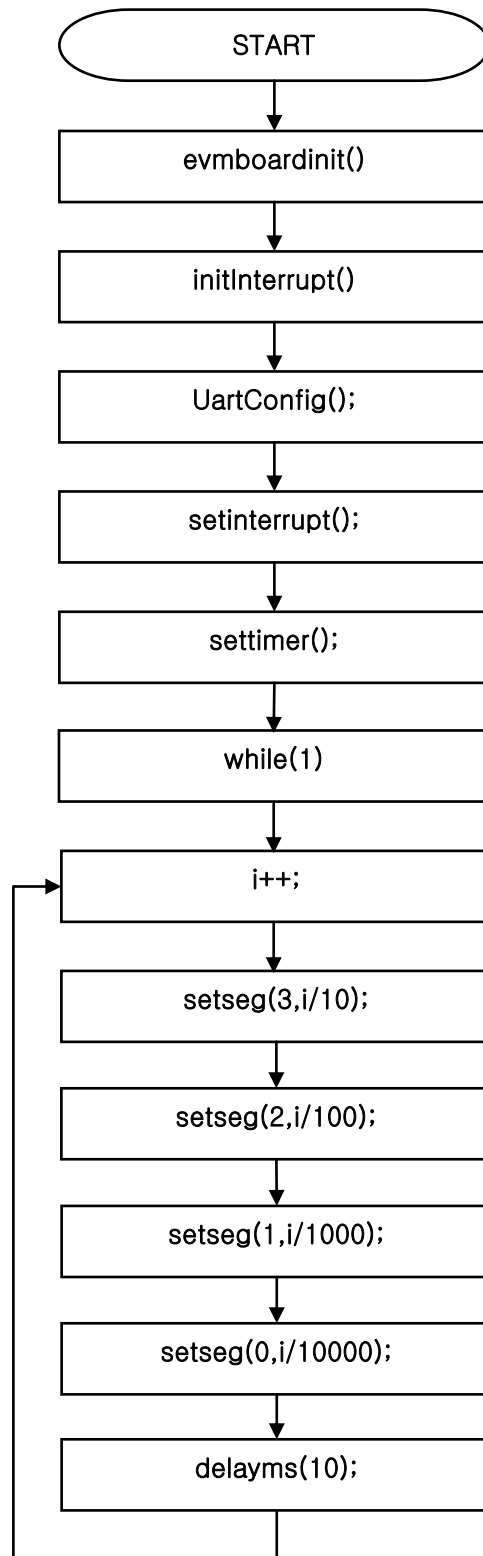


그림 3-1 Function Set Flow Chart

3.1.1 SetSeg()

```
void SetSeg(int n,U32 ch);
```

- int n : Digit Select Number
- U32 ch : 표시될 수(0~9)

```
static U8 segdata[4]={0,};
int tick=0;
static U8 getbitdata[10]={0x3,0x9f,0x25,0xd,0x99,0x49,0x41,0x1b,1,9};

void SetSeg(int n,U32 ch)
{
    if(n>3)
    {
        return;
    }
    ch %=10;
    segdata[n]=getbitdata[ch];
}
```

4개의 Digit에는 각 자리에 맞는 수가 (0~9) 사이의 값을 가져야 한다. 자리에 맞는 수를 표시하기 위한 Digit의 Binary 값을 지정하는 함수이다.

3.1.2 timer0isr();

```
#define FND_BASE_ADDR      (0x60000000 | (1<<17)|(1<<15))
#define FND_BASE_DATA     (0x60000000 | (1<<17)|(1<<14))

#define SEG1      (1<<1)
#define SEG2      (1<<2)
#define SEG3      (1<<3)
#define SEG4      (1)

void timer0isr()
{
    U8 segbit[]={ SEG1, SEG2, SEG3, SEG4 };
    *(volatile U8*)FND_BASE_ADDR = segbit[tick];
    *(volatile U8*)FND_BASE_DATA = segdata[tick];
    tick++;
    tick %=4;
}
```

Timer Interrupt가 발생하면 timer0isr()에서는 Digit를 On 시키고, Data를 Display한다.