마이크로프로세서응용
(Interrupt)

2023. 2학기

Kookmin Univ. EMCO Lab.

Contents

1. PIT

2. PWM Interrupt

3. Interrupt Priority

4. 예제

1. PIT

- INTERRUPT = 어떤 조건하에서 조건을 만족하면 Flag를 띄어 사건을 처리 (PIT, PWM INTERRUPT, ADC INTERRUPT 등) ↳ ex) CAN

- PIT = 일정 주기를 가지고 실행되는 Interrupt =) 주변장치 (Module)

  - Periodic Interrupt Timer
  - 설정 주기 마다 발생
  - 최대 4개의 Timer 설정 가능
  - 각각의 Timer 마다 주기 설정 가능

=) 모든 주변장치는 clak을 사용하여 운영

bus로 Register 설정 (clock을 몇까지 세고 설정등)

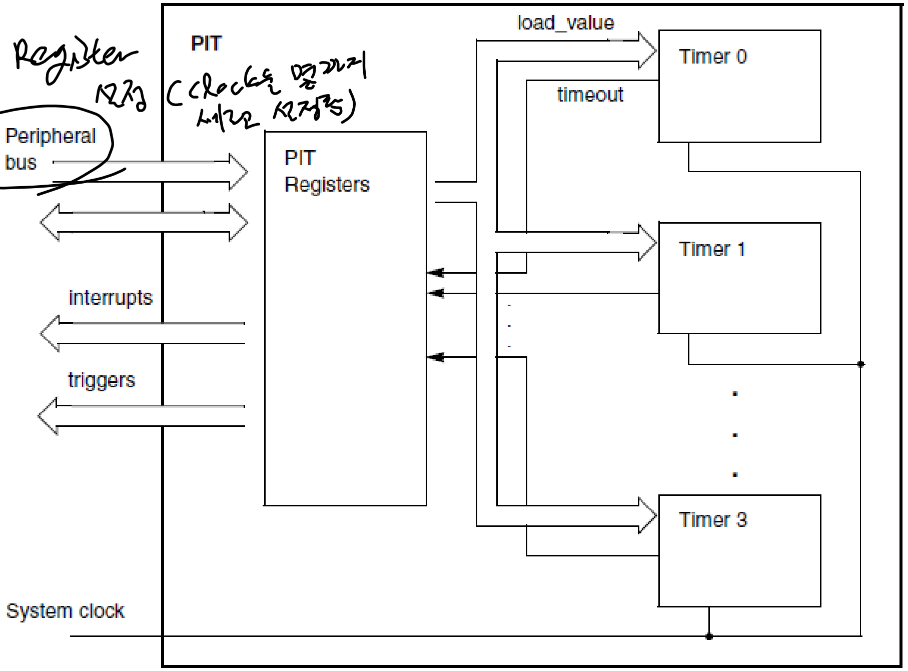


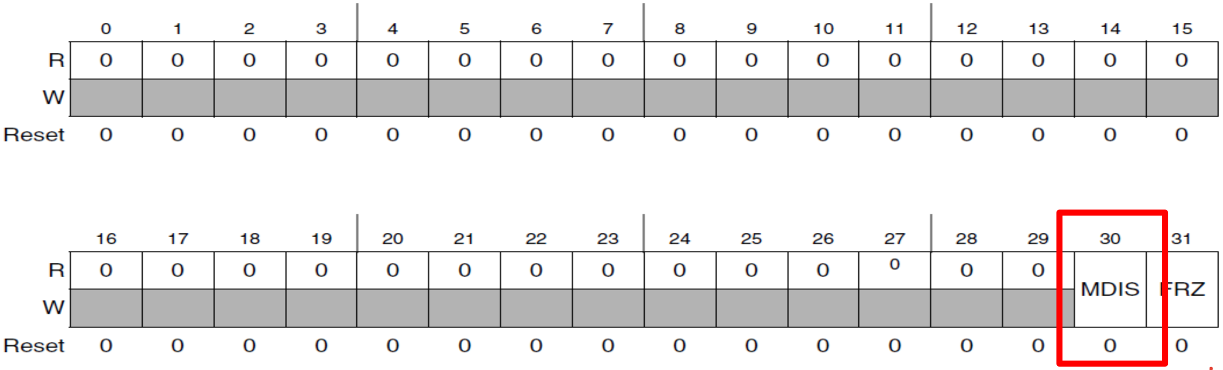
Figure 31-1. PIT block diagram

1. PIT

- PIT Enable Resister Setting

=) 사용하지 않는 것은 disable해서
전력을 줄임

- MDIS Bit 설정
- PIT.PITMCR.B.MDIS = 0

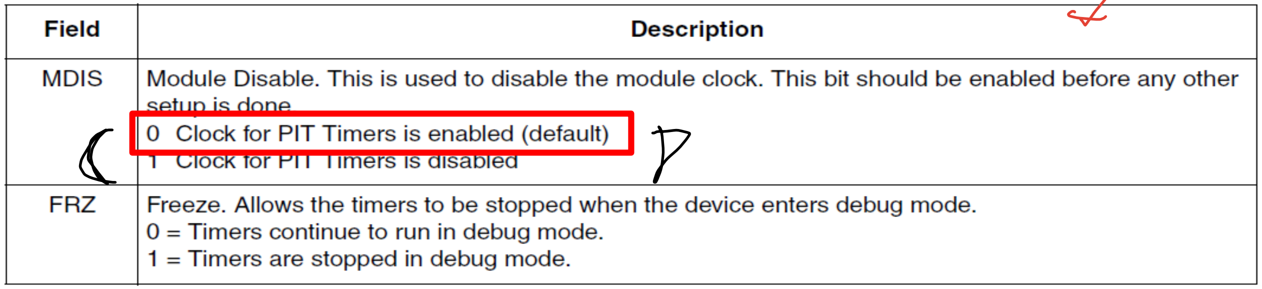


	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	MDIS	FRZ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 36-2. PIT Module Control Registers (PITMCR)

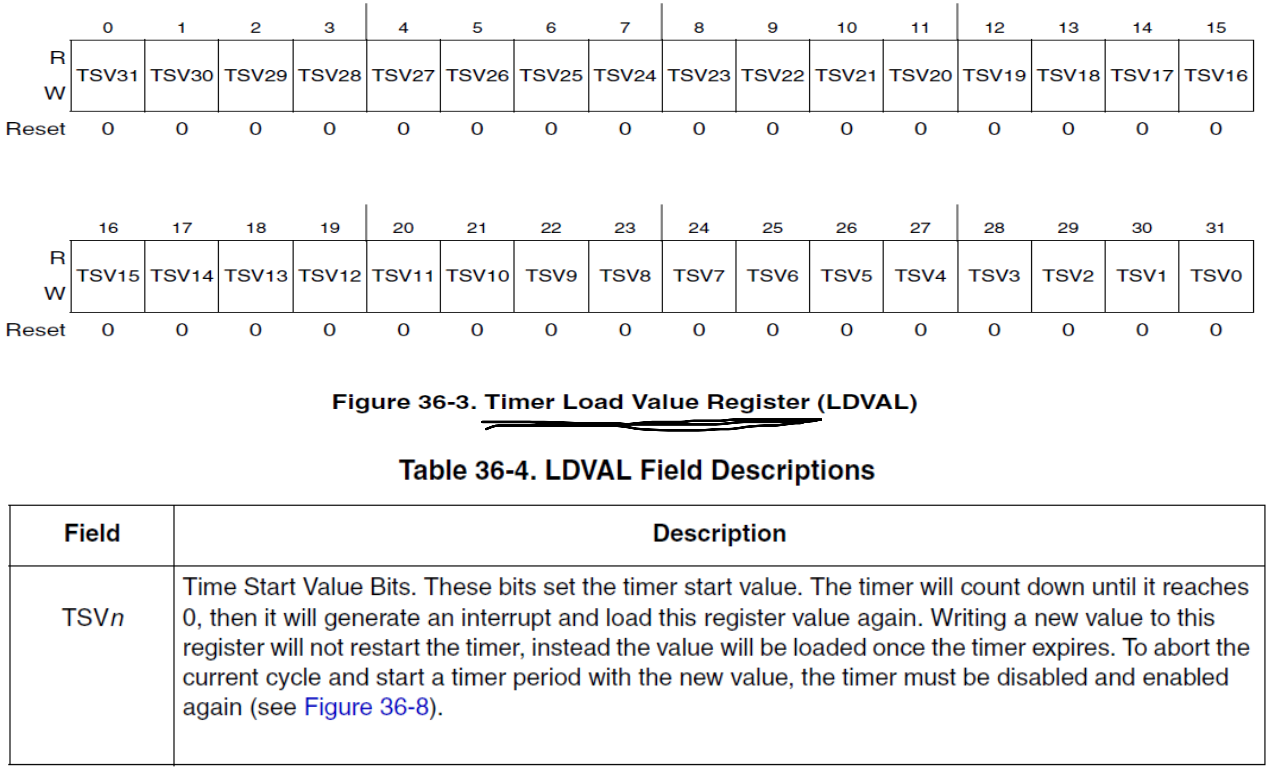
Table 36-3. PITMCR Field Descriptions



Field	Description
MDIS	Module Disable. This is used to disable the module clock. This bit should be enabled before any other setup is done. 0 Clock for PIT Timers is enabled (default) 1 Clock for PIT Timers is disabled
FRZ	Freeze. Allows the timers to be stopped when the device enters debug mode. 0 = Timers continue to run in debug mode. 1 = Timers are stopped in debug mode.

1. PIT

- PIT Period Resister Setting



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	TSV31	TSV30	TSV29	TSV28	TSV27	TSV26	TSV25	TSV24	TSV23	TSV22	TSV21	TSV20	TSV19	TSV18	TSV17	TSV16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	TSV15	TSV14	TSV13	TSV12	TSV11	TSV10	TSV9	TSV8	TSV7	TSV6	TSV5	TSV4	TSV3	TSV2	TSV1	TSV0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 36-3. Timer Load Value Register (LDVAL)

Table 36-4. LDVAL Field Descriptions

Field	Description
TSV*n*	Time Start Value Bits. These bits set the timer start value. The timer will count down until it reaches 0, then it will generate an interrupt and load this register value again. Writing a new value to this register will not restart the timer, instead the value will be loaded once the timer expires. To abort the current cycle and start a timer period with the new value, the timer must be disabled and enabled again (see Figure 36-8).

Description

- Resister에 설정 된 Value에서 시스템 Clock 의 주기로 0까지 Count down
- 이후 Interrupt 발생 후 다시 원래 Value로 돌아가서 반복.

1. PIT

- PIT Period Resister Setting

  - 수식으로 표현

$$\text{PIT Period} = \frac{1}{\underbrace{SystemClock}_{64MHz}} * \underline{LDVAL\ Resister\ Value}$$ => 실제시간

Example)

System Clock = 64Mhz

PIT.CH[0].LDVAL.R = 6400; // $\frac{1}{SystemClock} * 6400 = \boxed{100us}$

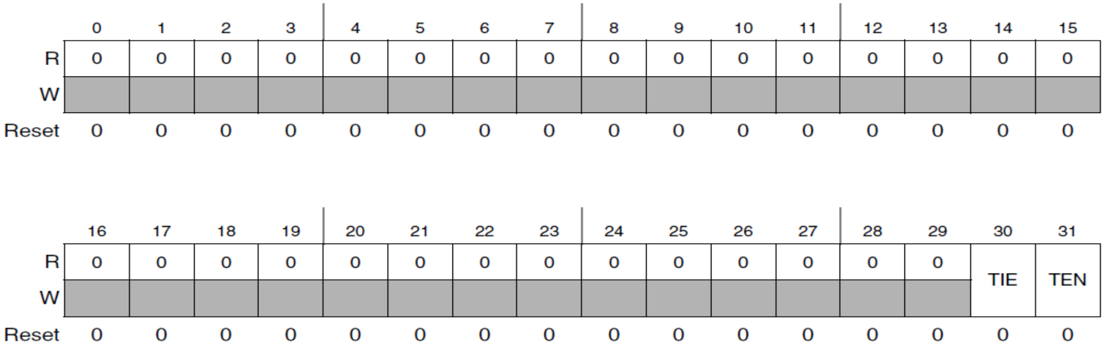
PIT.CH[1].LDVAL.R = 32000; // $\frac{1}{SystemClock} * 32000 = \boxed{500us}$

PIT.CH[2].LDVAL.R = 64000; // $\frac{1}{SystemClock} * 64000 = \boxed{1ms}$

Timer

1. PIT

- PIT 각각의 Timer Enable 및 Start Resister Setting



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TIE	TEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 36-5. Timer Control Register (TCTRL)

Table 36-6. TCTRL Field Descriptions

Field	Description
TIE	Timer Interrupt Enable Bit. 0 Interrupt requests from Timer x are disabled 1 Interrupt will be requested whenever TIF is set When an interrupt is pending (TIF set), enabling the interrupt will immediately cause an interrupt event. To avoid this, the associated TIF flag must be cleared first.
TEN	Timer Enable Bit. 0 Timer will be disabled 1 Timer will be active

Description

- TIE BIT 각각의 Timer의 Interrupt enable
- TEN BIT 각각의 Timer의 ~~Interrupt~~ Start

Timer Start

1. PIT

Example)

0,1번 Timer의 Interrupt Enable 및 Start Setting

방법1

Resister Setting 1

```
PIT.CH[0].TCTRL.B.TIE   = 0x1 ;    /* Timer 0 Interrupt : Enabled     */
PIT.CH[1].TCTRL.B.TIE   = 0x1 ;    /* Timer 1 Interrupt : Enabled     */

PIT.CH[0].TCTRL.B.TEN  = 0x1 ;    /*Start Timer 0 is : Enabled    */
PIT.CH[1].TCTRL.B.TEN  = 0x1 ;    /*Start Timer 1 is : Enabled    */
```

Resister의 bit하나 setting 0x0001 or 1

방법2

Resister Setting 2   0x0003 ←

```
PIT.CH[0].TCTRL.R = 0x3 ; /* Timer 0 Interrupt  Enabled & Start    */
PIT.CH[1].TCTRL.R = 0x3 ; /* Timer 01Interrupt  Enabled & Start    */
```

Resister 전체 Setting.

1. PIT

기본 초기화 설정

- PIT Ch0이 100ms마다 한번씩 발생하도록 설정

```
void init_PIT(void)
{
    PIT.PITMCR.R = 0x00000001;        // Enable PIT and Config Stop in debug mode.
    PIT.CH[0].LDVAL.R = 0x0061a800; // 6400000 //Timeout= (6.4M) x 1sec / 64M sysclks = 100 ms
    PIT.CH[0].TCTRL.R = 0x000000003;// Enable PIT0 interrupt & start PIT counting
}
```

Interrupt 함수

```
vuint32_t Pit0cnt=0;
void PIT0ISR(void)
{
    Pit0cnt++;

    PIT.CH[0].TFLG.B.TIF = 1;          // Clear PIT0 flag
}
```

~~Ines~~ Interrupt service routine

Interrupt finish

Interrupt 함수 실행 설정(main 함수 안)

```
init_ADC1();
init_PIT();

INTC_InstallINTCInterruptHandler(PIT0ISR, 59, 6);

SIU.PCR[48].R=0x0100;
```

p13에 설명

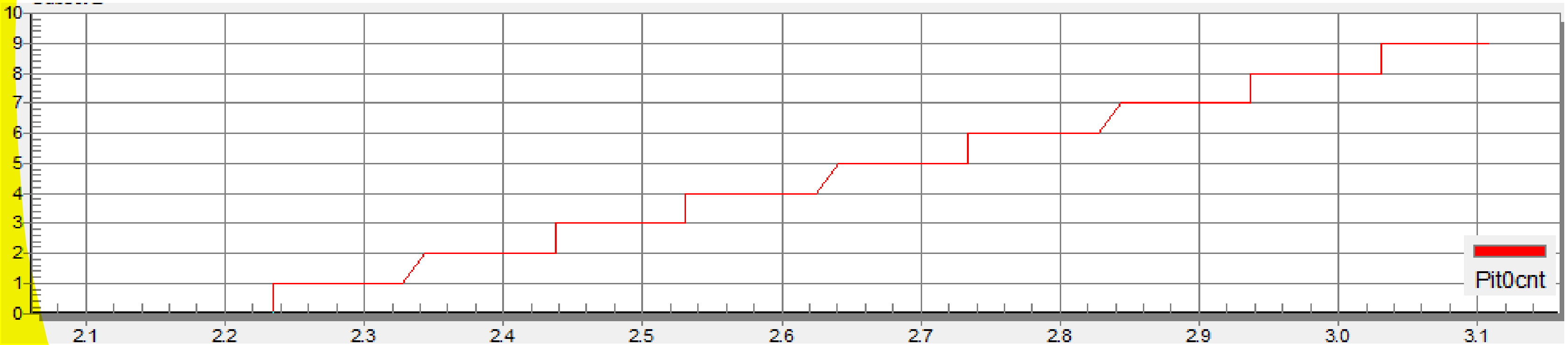
~~Tim~~ Timer 0 interrupt가 발생하면 PIT0ISR로 이동

InterruptHandler

priority level

1. PIT

100ms마다 수행 확인



2. PWM Interrupt

PWM Interrupt

- PWM HIGH CENTER

(가변저항으로 파형 확인)

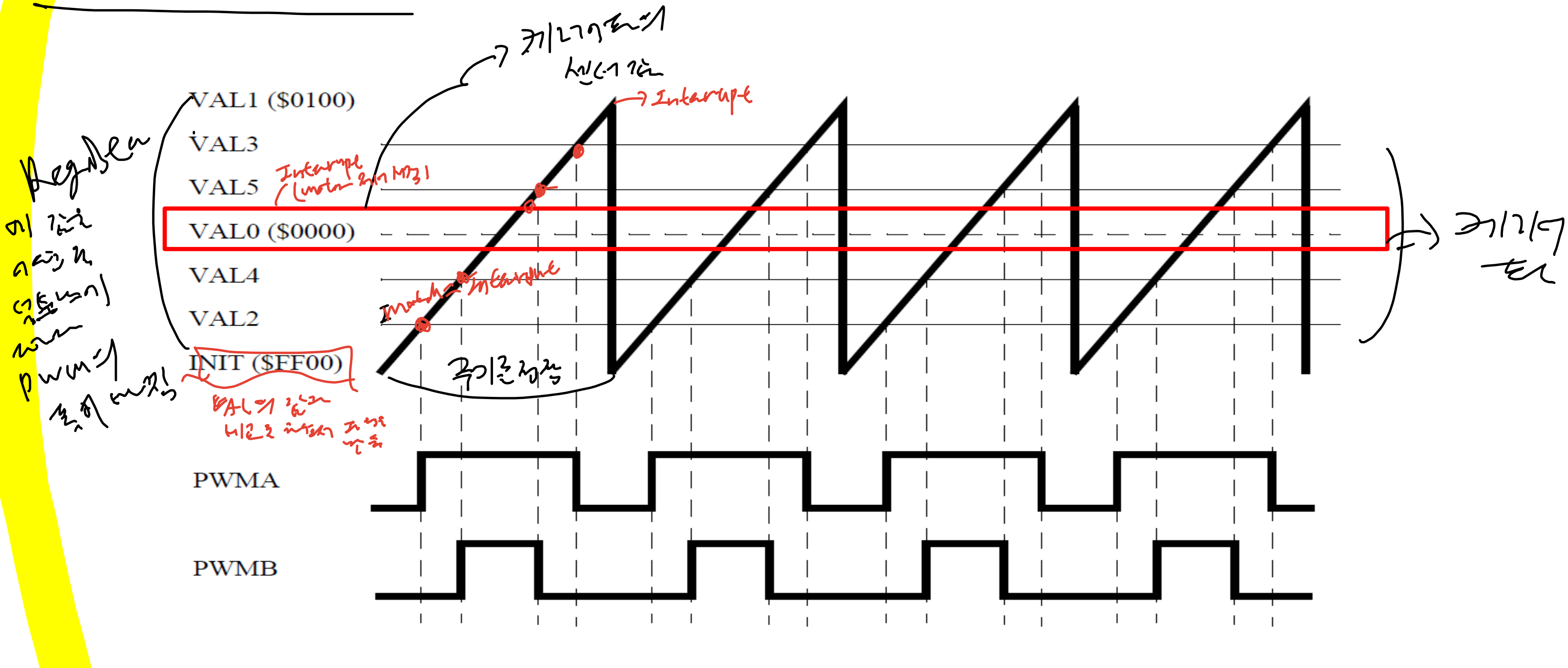


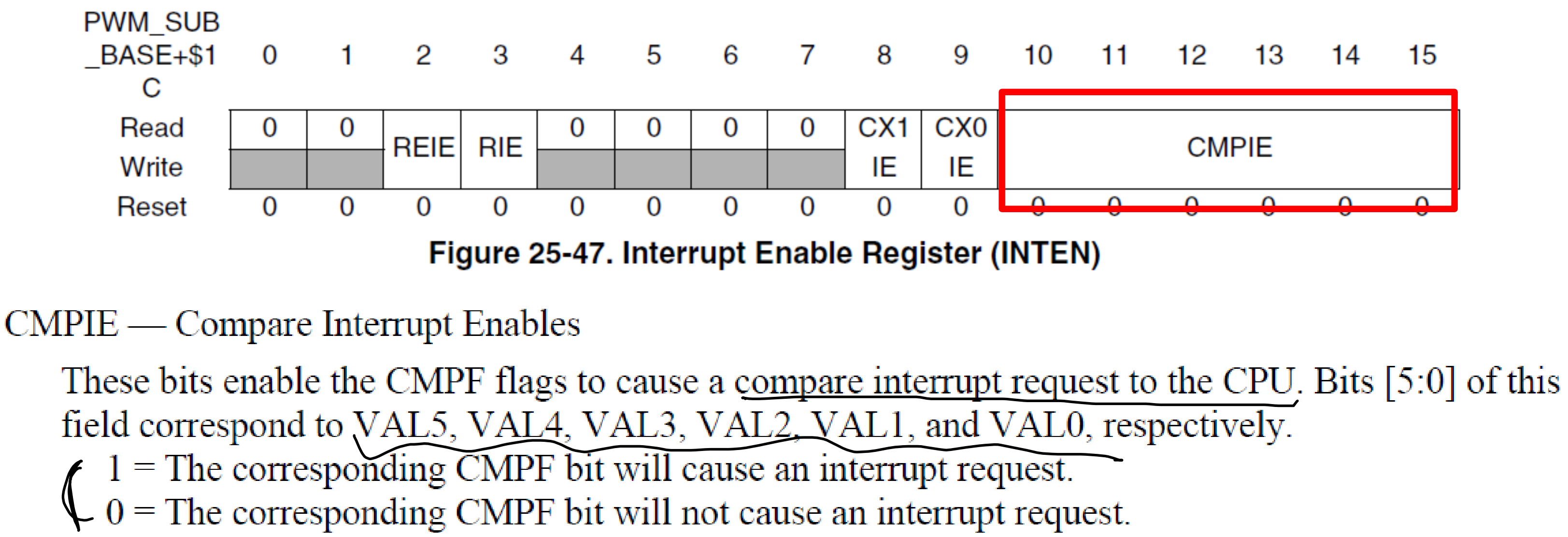
Figure 25-3. Center Aligned Example

- PWM Center Aligned 방식이기 때문에 VAL0 에서 Interrupt를 띄우면 항상 PWM HIGH CENTER 에서 발생

2. PWM Interrupt

PWM Interrupt Register Setting

- PWM HIGH CENTER



PWM_SUB_BASE+$1C	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Read	0	0	REIE	RIE	0	0	0	0	CX1 IE	CX0 IE	CMPIE					
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 25-47. Interrupt Enable Register (INTEN)

CMPIE — Compare Interrupt Enables

These bits enable the CMPF flags to cause a compare interrupt request to the CPU. Bits [5:0] of this field correspond to VAL5, VAL4, VAL3, VAL2, VAL1, and VAL0, respectively.

1 = The corresponding CMPF bit will cause an interrupt request.
0 = The corresponding CMPF bit will not cause an interrupt request.

Example)

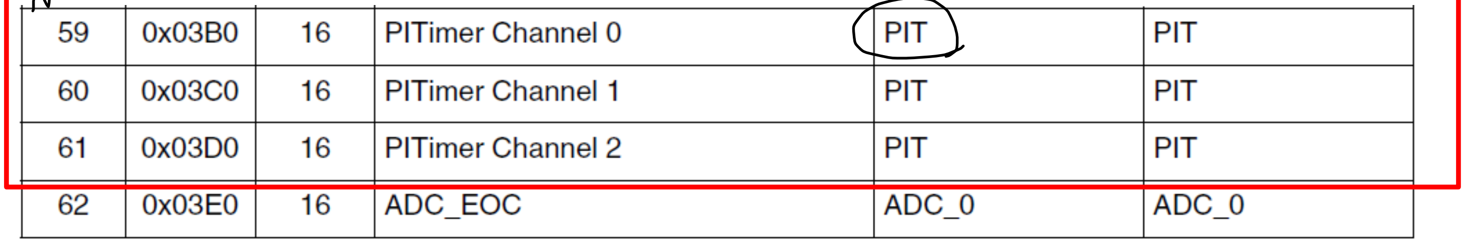
```
FLEXPWM_0.SUB[0].INTEN.R = 0x0000; // Value 0에서 Interrupt 발생 HIGH CENTER
FLEXPWM_0.SUB[0].INTEN.R = 0x0001; // Value 1에서 Interrupt 발생 LOW CENTER
```

3. Interrupt Priority

Interrupt Priority(우선순위)

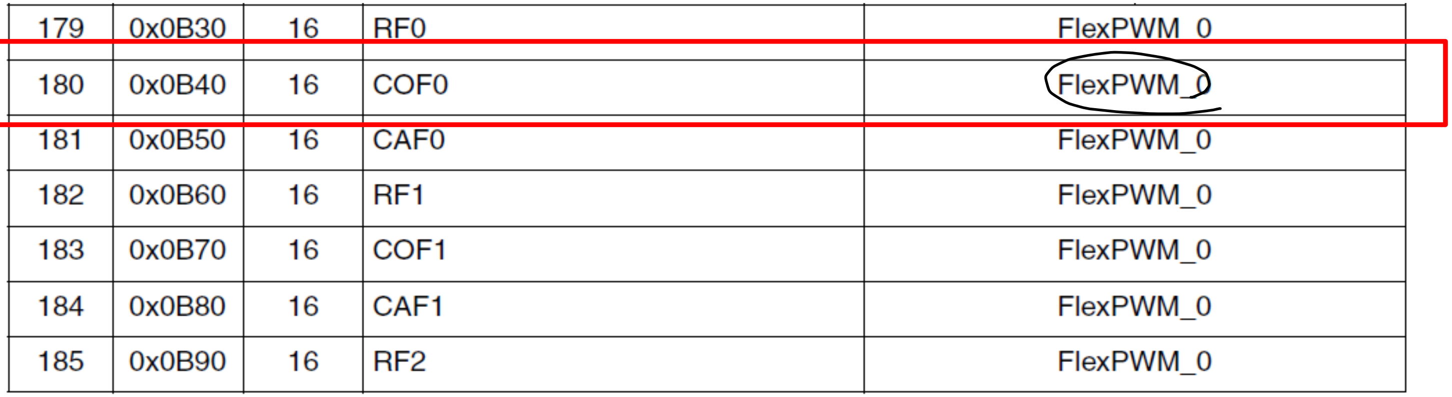
- INTC_InstallINTCInterruptHandler 사용 하여 우선순위 설정
- INTERRUPT VECTOR TABLE 사용

Hardware Number



59	0x03B0	16	PITimer Channel 0	PIT	PIT
60	0x03C0	16	PITimer Channel 1	PIT	PIT
61	0x03D0	16	PITimer Channel 2	PIT	PIT
62	0x03E0	16	ADC_EOC	ADC_0	ADC_0

Hard ware Number



179	0x0B30	16	RF0	FlexPWM_0
180	0x0B40	16	COF0	FlexPWM_0
181	0x0B50	16	CAF0	FlexPWM_0
182	0x0B60	16	RF1	FlexPWM_0
183	0x0B70	16	COF1	FlexPWM_0
184	0x0B80	16	CAF1	FlexPWM_0
185	0x0B90	16	RF2	FlexPWM_0

3. Interrupt Priority

INTC_InstallINTCInterruptHandler Setting

PIT3 = 127

- INTC_InstallINTCInterruptHandler(함수이름, 벡터넘버, 우선순위)

  Example)

  Vector Number

  ```
  INTC_InstallINTCInterruptHandler(INT_PIT_1ms,61,6);     ③
  INTC_InstallINTCInterruptHandler(INT_PIT_5ms,127,6);    ②
  INTC_InstallINTCInterruptHandler(INT_U_CENTER,180,10);  ①
  ```

  => 수가 되면 Vector Number가 더 큰 것이 우선 순위

  우선순위 숫자가 더 큰 것이 우선 순위

- 인터럽트 대기와 실행

  동시에 여러 인터럽트가 요청되었을 때는, 큰 숫자의 벡터 값의 인터럽트가 높은 우선순위를 가짐

  높은 우선순위의 인터럽트가 발생하면, 현재 인터럽트 서비스 루틴을 중단하고, 새로운 인터럽트 서비스 루틴을 실행

- 위 Handler에서는 INT_U_CENTER 우선순위를 가짐

4. 예제

예제1) 1초 마다 LED를 깜박이시오.

예제2) 2초마다 LED를 오른쪽으로 SHIFT 시키시오.