

### 33.1 Introduction

Uses for eCAP include(eCAP 용도는 다음과 같다):

- Speed measurements of rotating machinery (for example, toothed sprockets sensed via Hall sensors) - 회전 기계의 속도 측정 (예 , 홀 센서를 통해 감지 된 톱니 형 스포로킷)
- Elapsed time measurements between position sensor pulses - • 위치 센서 펄스 간의 경과 시간 측정
- Period and duty cycle measurements of pulse train signals-• 펄스 트레인 신호의주기 및 듀티 사이클 측정
- Decoding current or voltage amplitude derived from duty cycle encoded current/voltage sensors
- 듀티 사이클 인코딩(부호화) 된 전류 / 전압 센서에서 파생 된 전류 또는 전압 진폭 디코딩(복호화)

#### 33.1.1 Features

The eCAP module includes the following features(eCAP 모듈에는 다음과 같은 기능이 있습니다.

):

- 4 개의 이벤트 타임 스탬프 레지스터 (4-event time-stamp registers) 각 32 비트 - ( CAP1-CAP4)
- Edge polarity selection for up to four sequenced time-stamp capture events - >4 개의 타임 스탬프 캡처 이벤트 시퀀스에 대한 에지극성 선택.
- Interrupt on either of the four events - • 네 가지 이벤트 중 하나에서 인터럽트
- Single shot capture of up to four event time-stamps - 최대 4 개의 이벤트 타임 스탬프 만큼 단일 샷 캡처 .
- Continuous mode capture of time-stamps in a four-deep circular buffer - 4 깊이 원형 버퍼에서 타임 스탬프의 연속 모드 캡처
- Absolute time-stamp capture - 절대(절대시간) 타임스탬프 캡처?
- Difference (Delta) mode time-stamp capture – Difference (Delta) mode 타임스탬프 캡처
- All above resources dedicated to a single input pin - 단일 입력 핀 전용 리소스.
- When not used in capture mode, the ECAP module can be configured as a single channel PWM output - • 캡처 모드에서 사용하지 않으면 ECAP 모듈을 단일 채널 PWM 출력으로 구성 할 수 있습니다

### 33.1.2 Description (설명)

One eCAP channel has the following independent key resources(하나의 eCAP 채널에는 다음과 같은 독립적 인 주요 리소스가 있습니다.):

- Dedicated input capture pin ( 캡처 전용 입력 핀 존재)
- 32-bit time base (counter) (32 비트 타임베이스 카운터 존재)
- 4 x 32-bit time-stamp capture registers (CAP1-CAP4) 32 비트 타임 스탬프 캡처 레지스터.
- 4-stage sequencer (Modulo 4 counter) that is synchronized to external events, ECAP pin rising/falling edges. - 외부 이벤트들이나 ECAP 핀의 상승/하강 에지에 동기화된 네단계의 시퀀서(Modulo 4 counter)가 존재한다.
- Independent edge polarity (rising/falling edge) selection for all 4 events - • 4 개의 모든 이벤트에 대해 독립적 인 에지 극성 (상승 / 하강 에지) 선택
- Input capture signal prescaling (from 2 to 62) - • 입력 캡처 신호의 프리 스케일링 (2 에서 62 까지)  
→ 입력신호가 너무 빠를경우 분주시켜 느리게 한다.
- One-shot compare register (2 bits) to freeze captures after 1 to 4 time-stamp events - • 1 회 - 4 회 타임 스탬프 이벤트 후 캡처를 고정하는 원샷 비교 레지스터 (2 비트)
- Control for continuous time-stamp captures using a 4-deep circular buffer (CAP1-CAP4) scheme  
- • 4-deep 원형 버퍼 (CAP1-CAP4) 체계를 사용하여 연속 타임 스탬프 캡처를 제어합니다.
- Interrupt capabilities on any of the 4 capture events - • 4 가지 캡처 이벤트 중 하나에서 인터럽트 기능

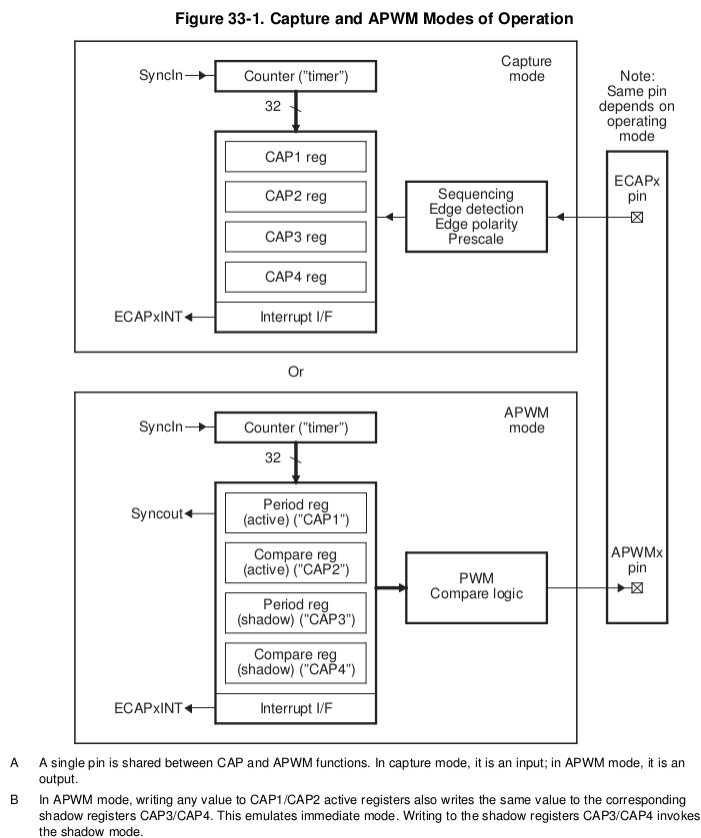
### 33.2 Basic Operation(기본 작동)

#### 33.2.1 Capture and APWM Operating Mode (33.2.1 캡처 및 APWM 작동 모드)

You can use the eCAP module resources to implement a single-channel PWM generator (with 32 bit capabilities) when it is not being used for input captures. The counter operates in count-up mode, providing a time-base for asymmetrical pulse width modulation (PWM) waveforms. The CAP1 and CAP2 registers become the active

period and compare registers, respectively, while CAP3 and CAP4 registers become the period and capture shadow registers, respectively. Figure 33-1 is a high-level view of both the capture and auxiliary pulse-width modulator (APWM) modes of operation.

→입력 캡처에 사용되지 않을 때 eCAP 모듈 리소스를 사용하여 단일 채널 PWM 생성기 (32 비트 기능 포함)를 구현할 수 있습니다. 카운터는 카운트 업 모드로 작동하여 비대칭 펄스 폭 변조 (PWM) 파형을 위한 시간 기준을 제공합니다. CAP1 및 CAP2 레지스터는 활성 기간 및 비교 레지스터가 되고 CAP3 및 CAP4 레지스터는 각각 주기 및 캡처 새도 레지스터가 됩니다. 그림 33-1 은 캡처 및 보조 펄스 폭 변조기 (APWM) 작동 모드의 상위 레벨보기입니다.



A. eCAP 의 단일 핀은 CAP mode 와 APWM 모다 간에 공유 된다. CAP 모드에서 이 핀은 입력이고, PWM 모드에서는 출력이다.

B. In APWM mode, writing any value to CAP1/CAP2 active registers also writes the same value to the corresponding shadow registers CAP3/CAP4. This emulates immediate mode. Writing to the shadow registers CAP3/CAP4 invokes the shadow mode.

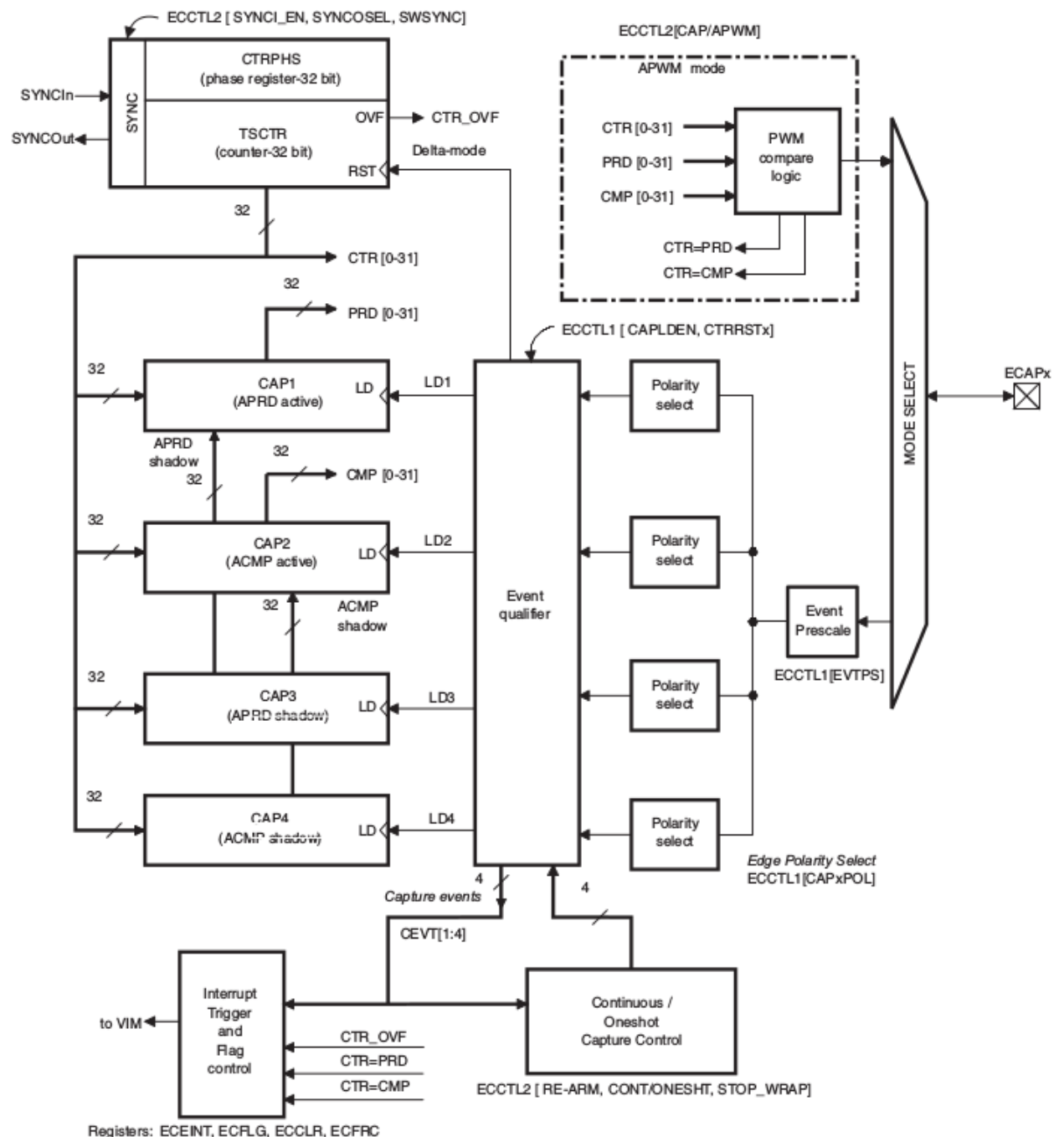
→ APWM 모드에서 CAP1 / CAP2 활성 레지스터에 값을 쓰면 해당 채널 레지스터 CAP3 / CAP4 에도 동일한 값이 기록됩니다. 이것은 즉시 모드를 에뮬레이트(비슷한 기능을 하는 모드가 생성된다)합니다. 채널 레지스터에 쓰기 CAP3 / CAP4 는 새도우 모드를 호출합니다.

### 33.2.2 Capture Mode Description ( 캡처 모드 설명)

Figure 33-2 shows the various components that implement the capture function.

-> 그림 33-2 는 캡처 기능을 구현하는 다양한 구성 요소를 보여줍니다.

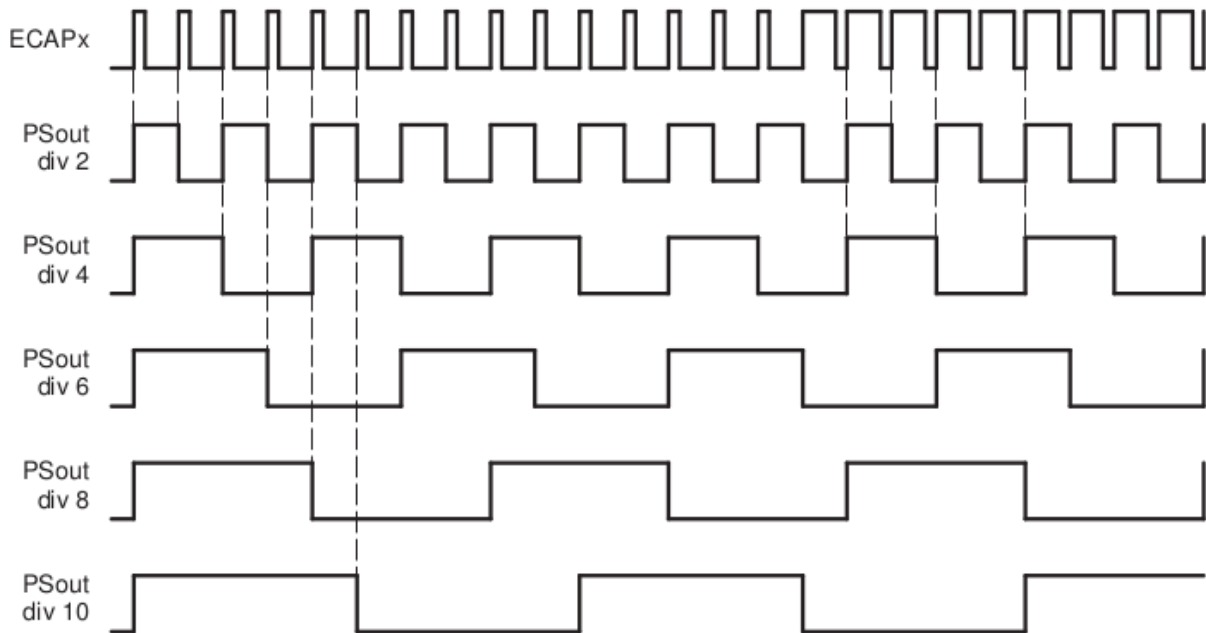
**Figure 33-2. Capture Function Diagram**



### 33.2.2.1 Event Prescaler (이벤트 분주)

ECAPx 는 외부 캡처 신호가 입력으로 들어오는 파형이다. 이 파형을 프리스케일을 통해 N 분주한 파형의 예시들이다.

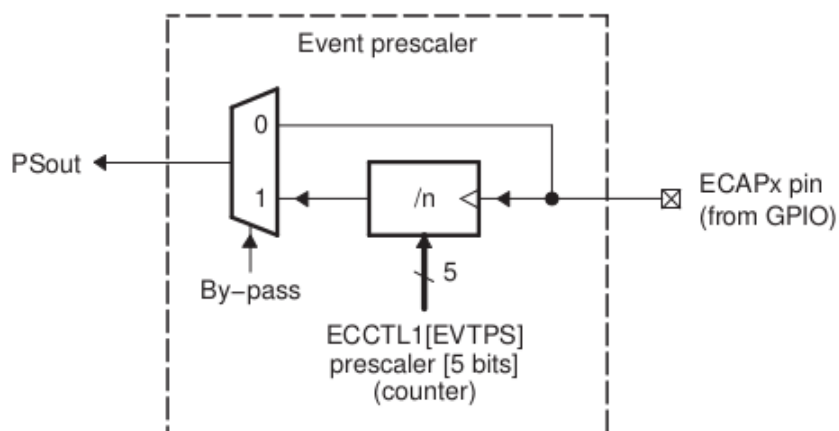
Figure 33-4. Prescale Function Waveforms



- An input capture signal (pulse train) can be prescaled by  $N = 2-62$  (in multiples of 2) or can bypass the prescaler.

→ 입력 캡처 신호는  $N = 2 \sim 62$  (2 의 배수) 까지 N 분주 하거나 바이패스 할 수 있다.

Figure 33-3. Event Prescale Control



- A When a prescale value of 1 is chosen ( $ECCTL1[13:9] = 0,0,0,0,0$ ), the input capture signal by-passes the prescale logic completely.

그림을 보면 ECCTL1 레지스터 값이 0 0 0 0 0 (1 을 선택한 경우)일 경우 입력신호가 바로 MUX 의 0 으로 빠져나가 바이패스 되는걸 알 수 있다.

### 33.2.2.2 Edge Polarity Select and Qualifier (에지 극성 선택과 수식어?)

- Four independent edge polarity (rising edge/falling edge) selection MUXes are used, one for each capture event.

→

- 4 개의 독립적 인 에지 극성 (상승 에지 / 하강 에지) 선택 MUX 가 각 캡처 이벤트에 하나씩 사용됩니다.

- Each edge (up to 4) is event qualified by the Modulo4 sequencer.

→

각각 에지는 Modulo4 시퀀서에 의해 event 로 규정된다.

- The edge event is gated to its respective CAPx register by the Mod4 counter. The CAPx register is loaded on the falling edge.

→

그 엣지 이벤트는 모듈로 4 카운터에 의해 각각의 CAPx 레지스터로 게이트드(통과) 된다.  
그 CAPx 레지스터는 하강 엣지일때 로드 된다.

(메모리에서 값이 로드 된다)

### 33.2.2.3 Continuous/One-Shot Control (연속 / 원샷 제어) – 연속모드, 원샷모드 제어

- The Mod4 (2 bit) counter is incremented via edge qualified events (CEVT1-CEVT4).

→Mod4 (2 비트) 카운터는 에지 규정 이벤트 (CEVT1-CEVT4)를 통해 증가합니다.

\*CEVTx = Capture Event flag 1 일때 CAPx 레지스터가 캡처했다는걸 의미

- The Mod4 counter continues counting (0->1->2->3->0) and wraps around unless stopped.

-> Mod4 카운터는 카운팅을 계속하고 (0-> 1-> 2-> 3-> 0) 멈추지 않으면 wraps around 합니다.

- A 2-bit stop register is used to compare the Mod4 counter output, and when equal stops the Mod4 counter and inhibits further loads of the CAP1-CAP4 registers. This occurs during one-shot operation.

-> 2 비트 스톱 레지스터는 Mod4 카운터 출력과 비교되고 Mod4 카운터 값과 같을때 CAP1-CAP4 레지스터의 로드를 금지시킵니다. 이 동작은 원샷모드일때 발생한다.

The continuous/one-shot block controls the start/stop and reset (zero) functions of the Mod4 counter via a mono-shot type of action that can be triggered by the stop-value comparator and re-armed via software control.

->

연속 / 원샷 블록은 stop-value comparator에 의해 작동되고 소프트웨어 제어를 통해 다시 활성화 될 수 있는 모노 샷(원샷) 타입의 동작을 통해 Mod4 카운터의 시작/정지 및 리셋(0) 기능을 제어한다.

Once armed, the eCAP module waits for 1-4 (defined by stop-value) capture events before freezing both the Mod4 counter and contents of CAP1-4 registers (time-stamps).

->

일단 활성화되면, eCAP 모듈은 Mod4 카운터와 CAP1~4 레지스터(time-stamps)를 고정하기 전에 capture events 1-4 (CEVT1-4, defined by stop-value)를 기다린다. (CEVT 플래그가 정지상태 0 이어야함)

Re-arming prepares the eCAP module for another capture sequence. Also re-arming clears (to zero) the Mod4 counter and permits loading of CAP1-4 registers again, providing the CAPLDEN bit is set.

->

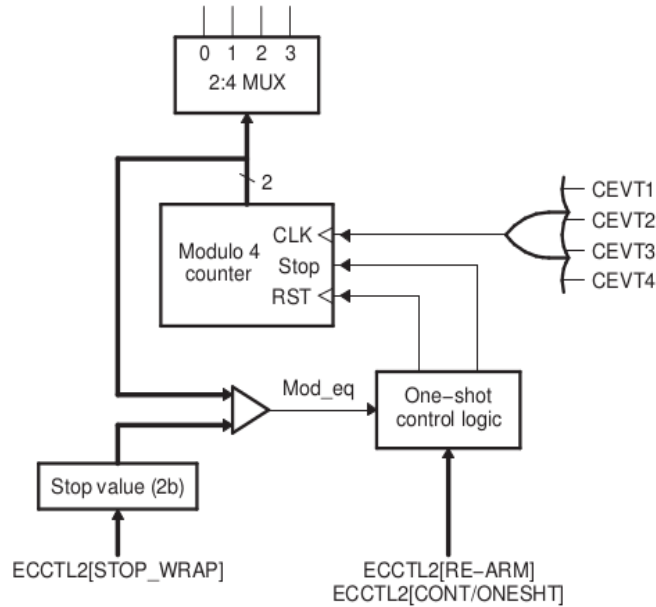
재 활성화는 다른 캡처 시퀀스를 위해 eCAP 모듈을 준비합니다. 또한 다시 활성화하면 Mod4 카운터가 지워지고 CAPLDEN(CAPx를 로드할지 말지 결정하는 레지스터) 비트가 1로 설정되면 CAP1-4 레지스터를 다시 로드 할 수 있습니다.

In continuous mode, the Mod4 counter continues to run (0->1->2->3->0, the one-shot action is ignored, and capture values continue to be written to CAP1-4 in a circular buffer sequence.

->

연속 모드에서 Mod4 카운터는 계속 실행됩니다 (0-> 1-> 2-> 3-> 0, 원샷 동작은 무시되며 캡처 값은 원형 버퍼 시퀀스에서 CAP1-4 에 계속 기록됩니다.

Figure 33-5. Continuous/One-shot Block



#### 33.2.2.4 32-Bit Counter and Phase Control (32 비트 카운터와 위상 제어)

This counter provides the time-base for event captures, and is clocked via the system clock.

->

이 카운터는 이벤트 캡처에 대한 시간 기준을 제공하며 시스템 클럭을 통해 클럭(카운팅)됩니다.

A phase register is provided to achieve synchronization with other counters, via a hardware and software forced sync. This is useful in APWM mode when a phase offset between modules is needed.

->

위상 레지스터는 하드웨어 및 소프트웨어 강제 동기화를 통해 다른 카운터와 동기화를 제공합니다. 이것은 모듈 간의 위상 오프셋이 필요할 때 APWM 모드에서 유용합니다.

(클럭이 들어올 때 마자 카운팅해 어느 시점에 위상을 바꾸는 것을 의미, 하드웨어 소프트웨어의 시간적 타이밍을 맞춰주어야 위상의 변경이 원하는 시점에 된다는걸 설명한거 같음)

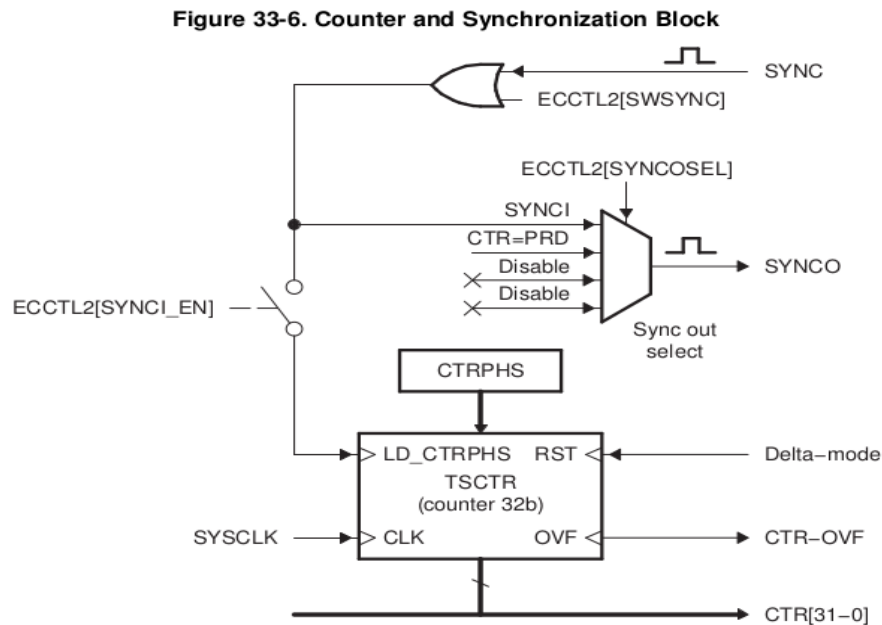
On any of the four event loads, an option to reset the 32-bit counter is given. This is useful for time difference capture. The 32-bit counter value is captured first, then it is reset to 0 by any of the LD1-LD4 signals.

→



네 가지 이벤트로드 중 하나에서 32 비트 카운터를 재설정하는 옵션이 제공됩니다. 이것은 시간차 캡처에 유용합니다. 32 비트 카운터 값이 먼저 캡처 된 다음 LD1-LD4 신호 중 하나에 의해 0 으로 재설정됩니다.

(잘 모르겠음.)



### 33.2.2.5 CAP1-CAP4 Registers

These 32-bit registers are fed by the 32-bit counter timer bus, CTR[0-31] and are loaded (that is, capture a time-stamp) when their respective LD inputs are strobed.

→

이 32 비트 레지스터는 32 비트 카운터 타이머 버스 CTR [0-31]에 의해 공급되며 각각의 LD 입력이 스트로브(?) 될 때 로드됩니다 (즉, 타임 스탬프를 포착합니다).

A counter overflow event (FFFFFFFF->00000000) is also provided as an interrupt source (CTROVF).

→카운터 오버플로 이벤트 (FFFFFFFF-> 00000000)도 인터럽트 소스 (CTROVF)로 제공됩니다.

The capture events are edge and sequencer qualified (ordered in time) by the polarity select and Mod4 gating, respectively.

→캡처 이벤트는 극성 선택 및 Mod4 게이팅으로 각각 에지 및 시퀀서 자격이 부여됩니다 (시간 순서대로 정렬 됨).

One of these events can be selected as the interrupt source (from the eCAPx module) going to the PIE.

→

이러한 이벤트 중 하나는 PIE 로가는 인터럽트 소스 (eCAPx 모듈에서)로 선택 될 수 있습니다.

Seven interrupt events (CEVT1, CEVT2, CEVT3, CEVT4, CNTOVF, CTR = PRD, CTR = CMP) can be generated. The interrupt enable register (ECEINT) is used to enable/disable individual interrupt event sources. The interrupt flag register (ECFLG) indicates if any interrupt event has been latched and contains the global interrupt flag bit (INT). An interrupt pulse is generated to the PIE only if any of the interrupt events are enabled, the flag bit is 1, and the INT flag bit is 0. The interrupt service routine must clear the global interrupt flag bit and the serviced event via the interrupt clear register (ECCLR) before any other interrupt pulses are generated. You can force an interrupt event via the interrupt force register (ECFRC). This is useful for test purposes.

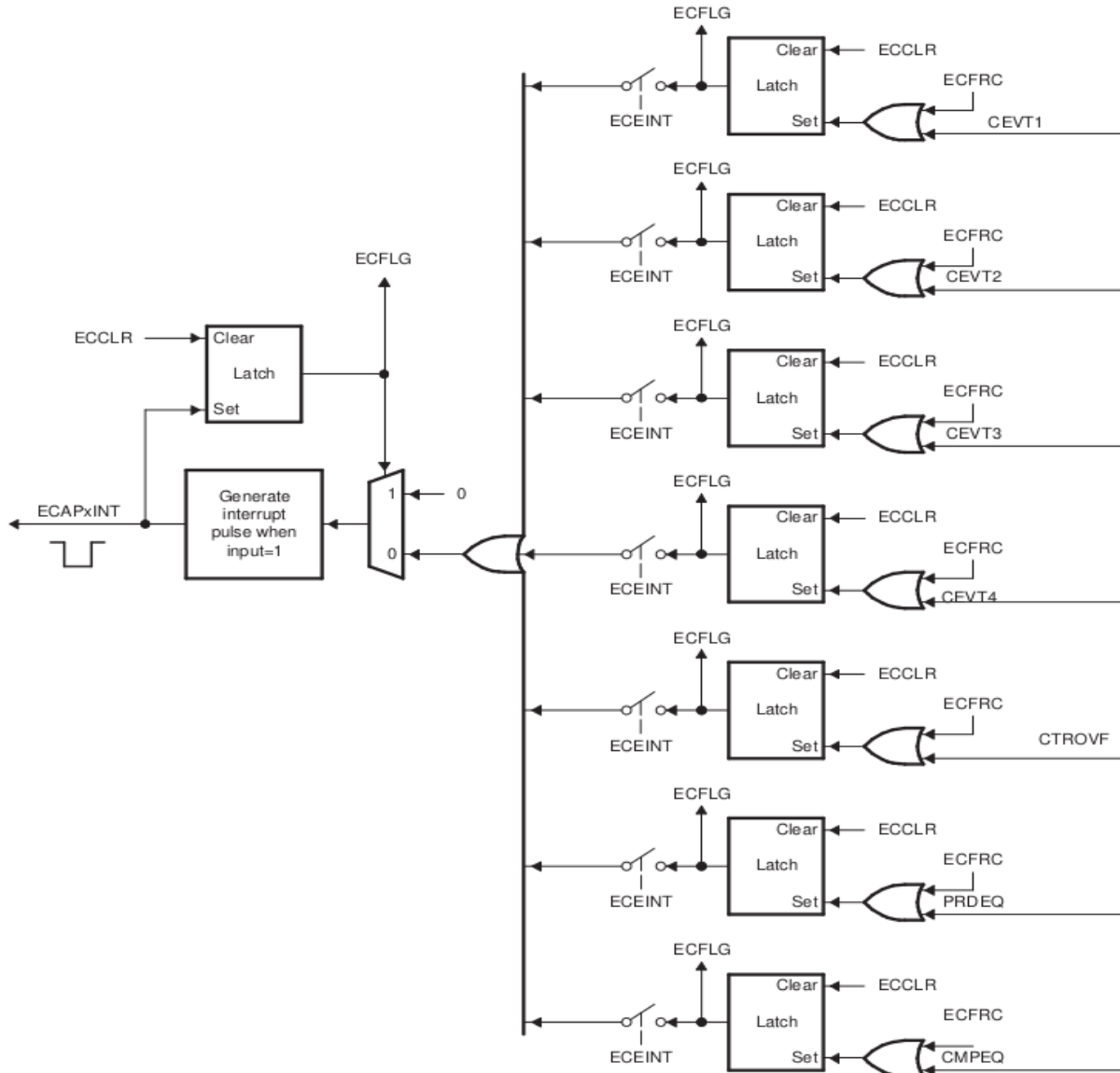
→

7 가지 인터럽트 이벤트 (CEVT1, CEVT2, CEVT3, CEVT4, CNTOVF, CTR = PRD, CTR = CMP)를 생성 할 수 있습니다. 인터럽트 인 에이블 레지스터 (ECEINT)는 개별 인터럽트 이벤트 소스를 활성화 / 비활성화하는 데 사용됩니다. 인터럽트 플래그 레지스터 (ECFLG)는 인터럽트 이벤트가 래치되어 전역 인터럽트 플래그 비트 (INT)를 포함하는지 여부를 나타냅니다. 인터럽트 이벤트 중 하나가 활성화되어 있고 플래그 비트가 1 이고 INT 플래그 비트가 0 인 경우에만 인터럽트 펄스가 PIE 에 생성됩니다. 인터럽트 서비스 루틴은 인터럽트를 통해 전역 인터럽트 플래그 비트와 서비스 이벤트를 지워야합니다 다른 인터럽트 펄스가 생성되기 전에 클리어 레지스터 (ECCLR)에 저장된다. 인터럽트 강제 레지스터 (ECFRC)를 통해 인터럽트 이벤트를 강제 실행할 수 있습니다. 이것은 테스트 목적으로 유용합니다.

**Note:** The CEVT1, CEVT2, CEVT3, CEVT4 flags are only active in capture mode (ECCTL2[CAP\_APWM == 0]). The CTR\_PRD and CTR\_CMP flags are only valid in APWM mode (ECCTL2[CAP\_APWM == 1]). CNTOVF flag is valid in both modes.

**노트 :** CEVT1, CEVT2, CEVT3, CEVT4 플래그는 캡처 모드에서만 활성화됩니다 (ECCTL2 [CAP\_APWM == 0]). CTR\_PRD 및 CTR\_CMP 플래그는 APWM 모드 (ECCTL2 [CAP\_APWM == 1])에서만 유효합니다. CNTOVF 플래그는 두 모드 모두에서 유효합니다.

**Figure 33-7. Interrupts in eCAP Module**



### 33.5.1 Time-Stamp Counter Register (TSCTR)

Bits	Field	Description
31-0	TSCTR	32 bit 카운터 레지스터 로 동작한다. 캡처 타임베이스를 사용된다.

### 33.5.2 Counter Phase Control Register (CTRPHS)

Bits	Field	Description
31-0	CTRPHS	Counter phase value register that can be programmed for phase lag/lead. 위상을 늦추거나 빠르게하는 레지스터.

### 33.5.3 Capture-1 Register (CAP1)

NOTE: APWM 모드에서 CAP1 / CAP2 활성 레지스터에 기록하면 해당 값도 해당 새도 레지스터 CAP3 / CAP4 에 기록됩니다. 이것은 즉시 모드를 에뮬레이트합니다. 새도 레지스터에 쓰기 CAP3 / CAP4 는 새도우 모드를 호출합니다.

Bits	Field	Description
31-0	CAP1	이 레지스터는 이런 상황에 로드될 수 있습니다 :  캡처 이벤트가 발생할 때 Time-Stamp (카운터 값). 소프트웨어 - 테스트 목적이거나 초기화에 유용하다. APRD Shadow register (CAP3) APWM 모드일때 /* This register can be loaded (written) by: <ul style="list-style-type: none"> <li>• Time-Stamp (counter value) during a capture event</li> <li>• Software - may be useful for test purposes</li> <li>• APRD shadow register (CAP4) when used in APWM mode</li> </ul> */

### 33.5.4 Capture-2 Register (CAP2)

Bits	Field	Description
31-0	CAP2	This register can be loaded (written) by:  캡처 이벤트가 발생할 때 Time-Stamp (카운터 값). 소프트웨어 - 테스트 목적이거나 초기화에 유용하다. APRD Shadow register (CAP3) APWM 모드일때

		/* • Time-Stamp (counter value) during a capture event • Software - may be useful for test purposes • APRD shadow register (CAP4) when used in APWM mode */
--	--	---

### 33.5.5 Capture-3 Register (CAP3)

Bits	Field	Description
31-0	CAP3	CMP 모드에서, 이 레지스터는 time-stemp capture register 이다. APWM 모드 에서, 이 레지스터는 주기를 도와주는 레지스터다 (APRD). 너희는 이레지스터로 PWM 주기 값을 업데이트 할 수 있다. 이 모드에서, CAP3(APRD) 는 CAP1 을 도와준다.

### 33.5.6 Capture-4 Register (CAP4)

Bits	Field	Description
31-0	CAP4	CMP 모드에서, 이 레지스터는 time-stemp capture 레지스터이다. APWM 모드에서, 이 레지스터는 레지스터 비교하는걸 도와준다.(ACMP) 너희는 PWM compare 값을 이 레지스터를 통해 업데이트 할 수 있다. 이 모드에서 CAP4(ACMP) 는 CAP2 를 도와준다.

eCAP 는 외부 이벤트의(외부 인터럽트??) 정확한 타이밍이 중요한 시스템에서 사용한다. 6 가지 경우를 갖고있다.

대충 이해한거는 외부 신호의 정확한 타이밍을 측정하는 도구?

## ECCTL1 – ECAP controll register

Bit	Field	Value	Description
0	CAP1POL	0 1	Capture Event 1 극성 선택 상승에지에서 Event 1 을 캡처한다. (RE) 하강에지에서 Event 1 을 캡처한다. (FE)
1	CTRRST1	0 1	Capture Event 1 의 Counter 를 초기화. 0 Capture Event 1 의 카운터를 초기화 시키지 않는다.(절대시간 스탬프) 1 Event 1 의 타임스탬프가 캡처 되었을때 초기화 (differece mode operation 에서 쓰임)
2	CAP2POL	0 1	Capture Event 2 극성 선택 상승에지에서 Event 2 를 캡처한다. (RE) 하강에지에서 Event 2 를 캡처한다. (FE)
3	CTRRST2	0 1	Capture Event 2 의 Counter 를 초기화. 0 Capture Event 2 의 카운터를 초기화 시키지 않는다. (절대시간 스탬프). 1 Event 2 의 타임스탬프가 캡처 되었을 때 초기화 (difference mode operation 에서 쓰임)
4	CAP3POL	0 1	Capture Event 3 극성 선택 상승에지에서 Event 3 를 캡처한다. (RE) 하강에지에서 Event 3 를 캡처한다. (FE)
5	CTRRST3	0 1	Capture Event 3 의 Counter 를 초기화. 0 Capture Event 3 의 카운터를 초기화 시키지 않는다. (절대시간 스탬프). 1 Event 3 의 타임스탬프가 캡처 되었을 때 초기화 (difference mode operation 에서 쓰임)
6	CAP4POL	0 1	Capture Event 4 극성 선택 상승에지에서 Event 4 를 캡처한다. (RE) 하강에지에서 Event 4 를 캡처한다. (FE)
7	CTRRST4	0 1	Capture Event 4 의 Counter 를 초기화. 0 Capture Event 4 의 카운터를 초기화 시키지 않는다. (절대시간 스탬프). 1 Event 4 의 타임스탬프가 캡처 되었을 때 초기화 (difference mode operation 에서 쓰임)
8	CAPLDEN	0 1	Capture Event 시에 CAP 1-4 레지스터를 로딩 할지 말지. 0 Capture event time 시 CAP 1-4 레지스터를 로드 하지 않는다. 1 Capture event time 시 CAP 1-4 레지스터를 로드 한다.

9-13	PRESCALE		Event Filter prescale select. (이벤트를 어떤식으로 나눌지)
		0	1 로 나눈다.
		1h	2 로나눈다.
		2h	4 로 나눈다.
		3h	6 으로 나눈다.33.5.1 Time-Stamp Counter Register (TSCTR)
		4h	8 으로 나눈다.
		5h	
		.	
		.	60 으로 나눈다.
		.	62 로 나눈다.
		1Eh	
		1Fh	
14-15	FREE/SOFT		Emulation Control
		0	에뮬레이션이 일시 중지시에 TBCTR 은 즉시 중지한다?
		1h	TBCTR 은 카운터가 0 일때까지 실행한다.
		2h-3h	Emulation 일시 중지 에 영향을 받지 않는다. (Run Free).

### 33.5.7 ECAP Control Register 2 (ECCTL2)

Bit	Field	Value	Description
0	CONT_ONESHT		연속 또는 원샷 모드 제어 (캡처 모드에서만 적용 가능).
		0	Operate in continuous mode.(연속모드로 동작시킨다.)
		1	Operate in one-shot mode. (원샷모드로 동작시킨다.)
1-2	STOP_WRAP		Stop value for one-shot mode. This is the number (between 1-4) of captures allowed to occur before the CAP(1-4) registers are frozen, capture sequence is stopped.
			Wrap value for continuous mode. This is the number (between 1-4) of the capture register in which the circular buffer wraps around and starts again. (모드에 적용되는 레지스터를 선택해주는거 같음)
		0	Stop after Capture Event 1 in one-shot mode. Wrap after Capture Event 1 in continuous mode.
		1h	Stop after Capture Event 2 in one-shot mode. Wrap after Capture Event 2 in continuous mode.
		2h	Stop after Capture Event 3 in one-shot mode.

		3h	<p>Wrap after Capture Event 3 in continuous mode.</p> <p>Stop after Capture Event 4 in one-shot mode.</p> <p>Wrap after Capture Event 4 in continuous mode.</p> <p>Notes: STOP_WRAP is compared to Mod4 counter and, when equal, 2 actions occur: (회로에서의 STOP_Value 레지스터) 카운터값이 같아지면 2 가지 동작을 한다.</p> <ul style="list-style-type: none"> <li>• Mod4 counter is stopped (frozen) 카운터 를 멈추고</li> <li>• Capture register loads are inhibited 캡처레지스터 로드를 금지하고, In one-shot mode, further interrupt events are blocked until re-armed. 원샷모드일 경우 재동작 할때까지 인터럽트를 블록하고 기다린다.</li> </ul>
3	REARM	0  1	<p>원샷 재동작 제어, 정지 트리거를 기다린다. 노트: 재동작 함수는 원샷모드랑 컨티너스 모드에서 쓸수있다.</p> <p>Has no effect (reading always returns a 0).</p> <p>Arms the one-shot sequence as follows: //원샷모드에서 밑에순서대로 한다.</p> <ol style="list-style-type: none"> <li>1) Resets the Mod4 counter to 0 //카운터를 0 으로 초기화</li> <li>2) Unfreezes the Mod4 counter // 카운터를 다시 동작시키고</li> <li>3) Enables capture register loads //캡처 레지스터 로드를 허용한다.</li> </ol>
4	TSCTRSTOP	0 1	<p>Time Stamp (TSCTR) Counter Stop (freeze) Control.</p> <p>0 TSCTR 을 멈춘다.</p> <p>1 TSCTR 을 계속 러닝시킨다.</p>
5	SYNCl_EN	0 1	<p>Counter (TSCTR) Sync-In select mode.</p> <p>0 Disable sync-in option.</p> <p>1 Enable counter (TSCTR) to be loaded from CTRPHS register upon either a SYNCl signal or a S/W force event.</p> <p>싱크신호나 소프트웨어에서 강제로 이벤트를 주는 경우에 CTRPHS 레지스터 값을 카운터로 로드될수 있게 한다.</p>
6-7	SYNCO_SEL	0  1h	<p>Sync-Out select.</p> <p>0 선택한 Synq-in 이벤트를 synq-out 신호로 통과시킨다.(synq-in = SYNC + ECCTL2[SWSYNC])</p> <p>1h CTR(Counter) = PRD(Period)인 event 를 sync-out 신호로 선택한다.</p>



		2h	Synq out signal 을 disable
		3h	synq out signal 을 disable
8	SWSYNC		소프트웨어 강제 카운터 (TSCTR) 동기화. 이것은 일부 또는 모든 ECAP 시간 기반을 동기화하기위한 편리한 소프트웨어 방법을 제공합니다. APWM 모드에서 CTR = PRD 이벤트를 통해 동기화를 수행 할 수도 있습니다. TSCTR 의 카운터를 맞춰준다는거 같음.
		0	Writing a 0 has no effect. Reading always returns a 0.(항상 무슨짓을하든 0 이다)
		1	1 을 쓰면 SYNCO_SEL 값이 TSCTR 한테 넘어가는거 같음. 그리고 사시 1 쓰면 0 을 반환하는거 같은데 잘 모르겠음..
9	CAP_APWM		CAP/APWM 동작모드 선택
		0	캡처모드다. 아래사항을 반드시 설정한다? 1. CTR = PRD 일때 TSCTR 을 리셋하지 않는다. 2. 새도우 레지스터가 CAP1 과 2 레지스터를 로드할 수 없다. 3. CA 1-4 레지스터를 로드가 가능하게 설정한다. 4. CAPx/APWMx 핀을 캡처 입력으로 동작한다.
		1	ECAP 가 APWM 모드로 동작한다. 1. TSCTR 은 CTR = PRD 일때 리셋한다. 2. 새도우 레지스터가 CAP1 과 2 레지스터를 로드할 수 있다. 3.타임 스탬프 값( 타임캡처값) CAP1-4 레지스터의 로딩을 디스에이블시킨다. 4. CAPx/APWM 핀을 APWM 출력으로 사용한다.
10	APWMPOL		APWM output polarity select. This is applicable only in APWM operating mode.
		0	Output is active-high (Compare value defines high time).
		1	Output is active-low (Compare value defines low time).
11-15	Reserved	0	Reserved

원샷 – Stop value 레지스터와 2 비트 카운터가 숫자가 같아지면 카운터가 멈추고 CAPx 레지스터를 로드하지 않는다.

컨티너스/원샷 이 카운터의 시작/정지 및 재설정 기능을 제어한다.

ECAP 가 동작하면 일단 캡처이벤트가 일어났을때 정지동작이 정의된 CAP1-4 를 기다린다.

CAPLDEN 가 셋 되었다는 조건하에 CAP 1-4 레지스터가 다시 로드가 가능하고, 2 비트 카운터를 초기화시키고 재동작시킨다. 그리고 eCAP 모듈은 다른 캡처동작을 준비한다.

In continuous mode, the Mod4 counter continues to run (0->1->2->3->0, the one-shot action is ignored, and capture values continue to be written to CAP1-4 in a circular buffer sequence.

다시 시작하는거 같음.

### 33.5.9 ECAP Interrupt Flag Register (ECFLG)

Bit	Field	Value	Description
0	INT	0 1	인터럽트 상태 플래그. 인터럽트가 생성되지 않았다. 인터럽트가 생성되었음을 나타냅니다.
1	CEVT1	0 1	캡처 이벤트 1 상태 플래그. 이 플래그는 CAP 모드에서만 동작한다. 이벤트가 일어나지 않았다. ECAPx 핀에서 첫번째 이벤트가 발생했음을 나타냅니다.
2	CEVT2	0 1	캡처 이벤트 2 상태 플래그. 이 플래그는 CAP 모드에서만 동작한다. 이벤트가 일어나지 않았다. ECAPx 핀에서 첫번째 이벤트가 발생했음을 나타냅니다.
3	CEVT3	0 1	캡처 이벤트 3 상태 플래그. 이 플래그는 CAP 모드에서만 동작한다. 이벤트가 일어나지 않았다. ECAPx 핀에서 첫번째 이벤트가 발생했음을 나타냅니다.
4	CEVT4	0 1	캡처 이벤트 4 상태 플래그. 이 플래그는 CAP 모드에서만 동작한다. 이벤트가 일어나지 않았다.

			ECAPx 핀에서 첫번째 이벤트가 발생했음을 나타냅니다.
5	CTROVF	0	카운터 오버플로우 상태 플래그. 이 플래그는 CAP 과 APWM 모드에서 동작한다.
		1	이벤트가 생성되지 않았다.
		1	카운터(TSCTR)가 FFFF FFFF 에서 0000 0000 으로 바뀌었다.
6	CTR_PRD	0	이플래그는 APWM 모드에서만 동작한다.
		1	아무 일도 일어나지 않았다.
		1	카운터(TSCTR)과 주기 레지스터(APRD)가 같아져서 리셋되었다는 플래그.
7	CTR_CMP	0	비교 상태와 같은지 비교하는 플래그. 이플래그는 APWM 모드에서만 동작한다.
		1	아무 사건도 일어나지 않았다.
		1	카운터(TSCTR)가 비교레지스터(ACMP)와 같아졌음을 나타냄.
8-15	Reserved	0	Reserved

### 33.5.10 ECAP Interrupt Enable Register (ECEINT)

Bit	Field	Value	Description
0	Reserved	0	Reserved
1	CEVT1	0	Capture Event 1 Interrupt Enable.
		1	Disable Capture Event 1 as an Interrupt source.
		1	Enable Capture Event 1 as an Interrupt source.
2	CEVT2	0	Capture Event 1 Interrupt Enable.
		1	Disable Capture Event 2 as an Interrupt source.
		1	Enable Capture Event 2 as an Interrupt source.

3	CEVT3	0 1	Capture Event 1 Interrupt Enable. Disable Capture Event 3 as an Interrupt source. Enable Capture Event 3 as an Interrupt source.
4	CEVT4	0 1	Capture Event 1 Interrupt Enable. Disable Capture Event 4 as an Interrupt source. Capture Event 4 Interrupt Enable.
5	CTROVF	0 1	Counter Overflow Interrupt Enable. Disabled counter Overflow as an Interrupt source. Enable counter Overflow as an Interrupt source.
6	CTR_PRD	0 1	Counter Equal Period Interrupt Enable. Disable Period Equal as an Interrupt source. Enable Period Equal as an Interrupt source.
7	CTR_CMP	0 1	Counter Equal Compare Interrupt Enable. Disable Compare Equal as an Interrupt source. Enable Compare Equal as an Interrupt source.
8-15	Reserved	0	Reserved

### 33.5.11 ECAP Interrupt Forcing Register (ECFRC)

Bit	Field	Value	Description
0	Reserved	0	Any writes to these bit(s) must always have a value of 0.
1	CEVT1	0 1	강제 캡처 이벤트 1. CEVT1 플래그 비트를 셋한다. 아무 영향없다.0 으로 읽음.
2	CEVT2	0 1	강제 캡처 이벤트 2 아무 영향 없다 항상 0 으로 읽힘 CEVT2 플래그에 1 을 셋팅한다.
3	CEVT3	0 1	강제 캡처이벤트 3 아무 영향 없다 항상 0 을 반환 CEVT3 플래그 비트에 1 을 셋팅한다.

4	CEVT4	0 1	강제 캡처 이벤트 4. 아무 영향 없다 항상 0 을 반환 플래그에 1 을 셋팅한다.
5	CTROVF	0 1	강제 카운터 오버플로우. 아무 영향 없다 0 반환 CTROVF 플래그에 1 을 셋팅한다.
6	CTR_PRD	0 1	강제 카운터 = 주기 인터럽트. 아무 영향 없다. 항상 0 을 반환한다. CTR_PRD 플래그 비트에 1 을 셋팅한다.
7	CTR_CMP	0 1	강제 카운터 = 비교레지스터값 인터럽트. 아무 영향 없다.항상 0 을 반환한다. 1 을 CTR_CMP 플래그 비트에 셋한다.
8-15	Reserved	0	Any writes to these bit(s) must always have a value of 0.

### 33.5.12 ECAP Interrupt Clear Register (ECCLR)

Bit	Field	Value	Description
0	INT	0 1	Global Interrupt Clear Flag. 0 을 쓰면 아무 영향 없다. 항상 0 을 반환한다. 1 을 쓰면 INT 플래그가 지워지고 이벤트 플래그 중 하나가 1 로 설정된 경우 추가 인터럽트가 생성됩니다.
1	CEVT1	0 1	Capture Event 1 Status Flag. Writing a 0 has no effect. Always reads back a 0. Writing a 1 clears the CEVT1 flag condition. 플래그를 초기화.
2	CEVT2	0 1	Capture Event 2 Status Flag. Writing a 0 has no effect. Always reads back a 0. Writing a 1 clears the CEVT2 flag condition. 플래그를 초기화.
3	CEVT3	0	Capture Event 3 Status Flag. Writing a 0 has no effect. Always reads back a 0.

		1	Writing a 1 clears the CEVT3 flag condition. 플래그를 초기화.
4	CEVT4	0 1	Capture Event 4 Status Flag. Writing a 0 has no effect. Always reads back a 0. Writing a 1 clears the CEVT4 flag condition. 플래그를 초기화.
5	CTROVF	0 1	카운터 오버플로우 상태 플래그 0 은 아무영향 없으며 항상 0 을 반환 1 을 쓰면 CTROVF 플래그를 초기화
6	CTR_PRD	0 1	카운터와 주지가 같은지 상태 플래그 0 을 쓰면 아무 영향 없고 항상 0 을 반환 1 을 쓰면 CTR_PRD 플래그 상태가 초기화 됨
7	CTR_CMP	0 1	카운터와 컴패어가 같은지 상태 플래그 0 을 쓰면 아무 영향 없으며 항상 0 을 반환한다. 1 을 쓰면 CTR_CMP 플래그 상태가 초기화 된다
8-15	Reserved	0	Any writes to these bit(s) must always have a value of 0.

## eCAP example Halcogen & CCS

- Duty, period

### 1. Halcogen setting

☐ Enable EQEP driver

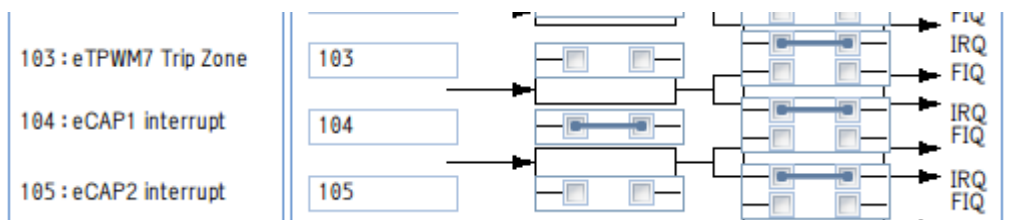
☐ Enable EQEP1 driver \*\*

☐ Enable EQEP2 driver \*\*

☒ Enable ETPWM driver

☒ Enable ECAP driver

Enable driver ETPWM , ECAP 다른건 Disable.



VM Channel 104 eCAP1 인터럽트 활성화.

Enable / Disable Peripherals

<input type="checkbox"/> HET1	<input type="checkbox"/> GIOA	<input type="checkbox"/> MIBSPI2
<input type="checkbox"/> HET2	<input type="checkbox"/> GIOB	<input type="checkbox"/> MIBSPI4
<input type="checkbox"/> EMIF	<input type="checkbox"/> EQEP	<input type="checkbox"/> AD1EVT
<input checked="" type="checkbox"/> ETPWM	<input checked="" type="checkbox"/> ECAP	<input type="checkbox"/> AD2EVT

PinMux 에서 ETPWM , ECAP 체크 ( 핀 충돌 일어나는데 상관없음. 충돌이 보기 싫다면 ECAP1 과 ETPWMA1 을 따로 찾아서 활성화 시키면 된다.)

☐ Use HET1\_LOOP\_SYNC for time-base sync

☒ Enable TBCLK sync\*\*

nTZ1 ASYNC

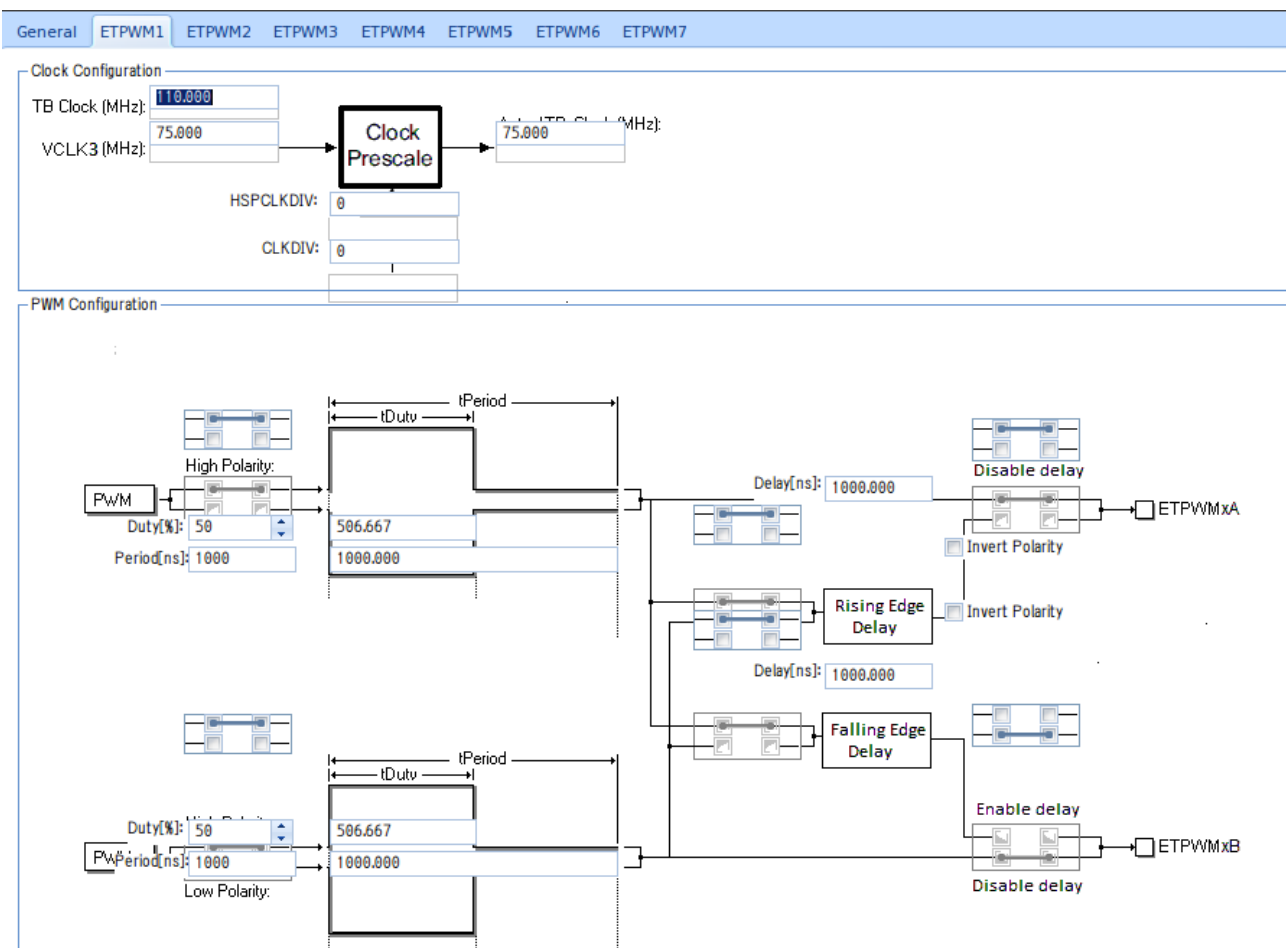
nTZ2 ASYNC

nTZ3 ASYNC

EPWM1 SYNC1 ASYNC

\*\*Done in etpwmInit

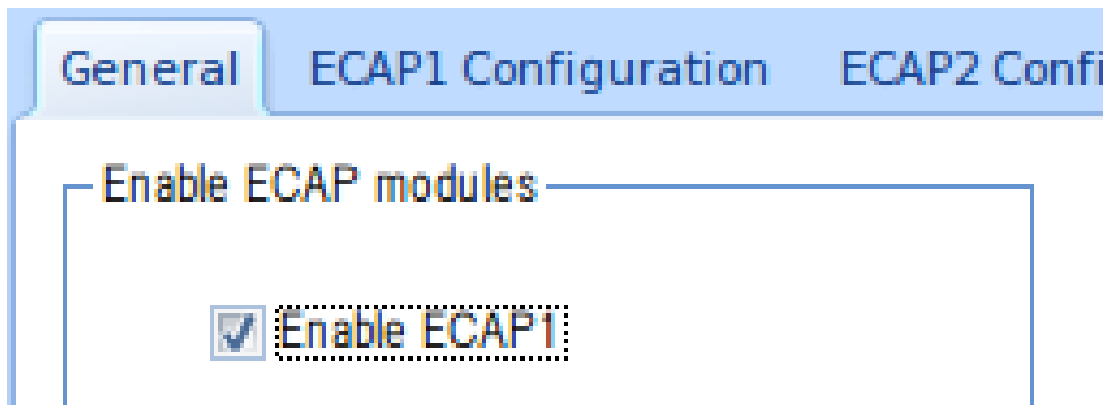
Special Pinmux 가서 Enable TBCLK sync 체크. (etpwm 타임 베이스 클럭이랑 동기화 시키는 것)



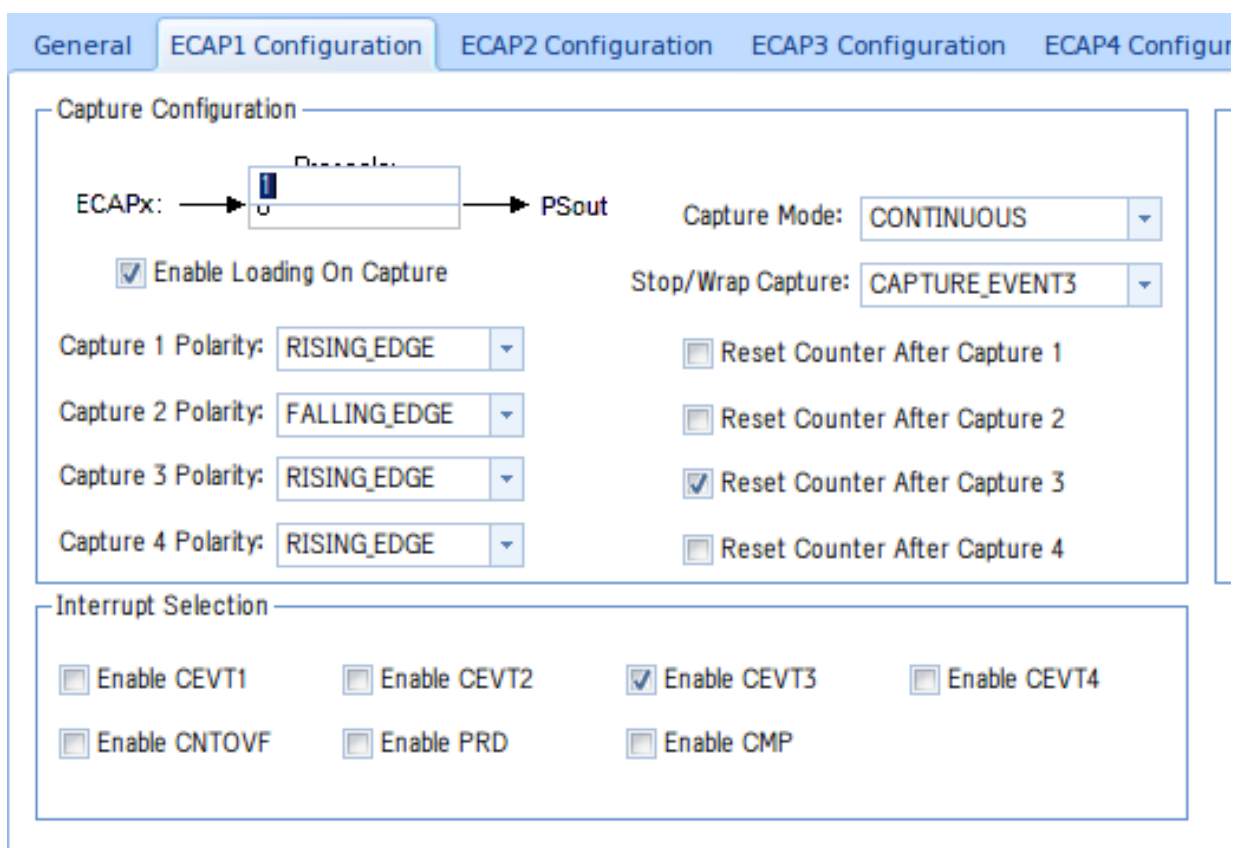
ETPWM1 설정.

( 주기와 듀티비는 자기 마음대로 바꿔도 상관없음. 외부신호만 만들어주면 됨)





ECAP1 만 사용하므로 ECAP1 활성화



Capture Polarity 는 캡처 동작을 실시할 엷지를 선택해 주는 것.

우리는 파형의 시작 듀티비 끝 파형의 끝을 알아야한다.

처음 파형의 rising edge 를 감지할 Capture event 1, 듀티의 끝의 falling edge 를 감지할 Capture event2, 파형의 끝 다시 시작되는 부분의 rising edge 를 감지할 Capture event3, 그리고 3 이 감지되면 초기화시키고 다시 파형을 감지하는 동작을 반복한다.

CONTINUOUS mode 한 이유는 절대시간이 아닌 상대시간을 알기 위해 즉 델타값을 알기 위해 설정.

Capture 3 이벤트가 감지되면 인터럽트 발생시켜 3 듀티 주기를 계산한다.

#### 소스코드

```
#include "HL_sys_common.h"
#include "stdio.h"
#include "HL_system.h"
#include "HL_etpwm.h"
#include "HL_ecap.h"

void main(void)
{
    _enable_interrupt();

    etpwmInit();
    ecapInit();
    etpwmStartTBCLK();
    ecapStartCounter(ecapREG1);
    ecapEnableCapture(ecapREG1);

    while(1);
}

void ecapNotification(ecapBASE_t *ecap, uint16 flags)
{
    uint32 cap1, cap2, cap3;
    float64 duty, period;

    cap1 = ecapGetCAP1(ecapREG1);
    cap2 = ecapGetCAP2(ecapREG1);
    cap3 = ecapGetCAP3(ecapREG1);

    duty = (cap2 - cap1)*1000/VCLK3_FREQ;
    period = (cap3 - cap1)*1000/VCLK3_FREQ;
```

```

printf("cap1 = %d\\n",cap1);
printf("cap2 = %d\\n",cap2);
printf("cap3 = %d\\n",cap3);
printf("Duty = %fns\\n", duty);
printf("Period = %fns\\n\\n", period);
}

```

printf 를 쓰게되면 느리다.

그래서 sci 로 값을 받아오게 코드를 다시 짜보도록 한다.

Halcogen 에서 sci1 driver 를 enable 하고 코드 제너레이션을 한다.

밑에 코드를 작성하고 업로드해 푸티를 실행시킨다.

```

#include "HL_sys_common.h"
#include "stdio.h"
#include "string.h"
#include "HL_sci.h"
#include "HL_system.h"
#include "HL_etpwm.h"
#include "HL_ecap.h"

void main(void)
{
    _enable_interrupt();

    etpwmInit();
    ecapInit();
    sciInit();
    etpwmStartTBCLK();
    ecapStartCounter(ecapREG1);
    ecapEnableCapture(ecapREG1);

    while(1);
}

```

