

Câu chuyện bắt đầu từ một cậu bé,  
và một ý tưởng  
có thể  
làm thay đổi thế giới...

PAY IT FORWARD

Đó là khi bạn giúp đỡ 3 người bạn không quen biết,  
dù là bằng thời gian,  
hay công sức,  
hay kinh nghiệm,  
hay kiến thức,  
hay tiền bạc, ...  
của mình.



Mà không chờ đợi một sự báo ân nào.

Chỉ cần mỗi người trong 3 người đó,  
lại đem những gì mình có, mà người khác cần,  
tiếp tục giúp đỡ thêm 3 người nữa.

Chính những người-giúp-đỡ, và người-được-giúp-đỡ,  
sẽ là những người góp phần thay đổi thế giới...

Một thế giới sẽ chia kiến thức - và yêu thương ...

PAY IT FORWARD ...

Chúng tôi không sáng tạo ra câu nói này.

Pay it forward...

Hãy tri ân người giúp mình bằng cách giúp đỡ người khác  
Cho đi không phải để nhận lại.

*CÂU LẠC BỘ NGHIÊN CỨU KHOA HỌC KHOA ĐIỆN-ĐIỆN TỬ  
ĐH BÁCH KHOA TP. HỒ CHÍ MINH*

# **MSP430 COURSE**

## **LESSON 4**

# **CAPTURE – COMPARE – PWM**

**Training document for C8 course**



# CAPTURE

Capture là 1 chức năng của timer được dùng để đo thời gian sự kiện , tính tốc độ và đo thời gian.

Số capture block phụ thuộc số lượng CCR mà nó có. Như ở

MSP430G2553 TIMERA0 có 3 capture CCR0,CCR1,CCR2

Tín hiệu vào của capture được chọn từ external pin hay internal signal bằng software.

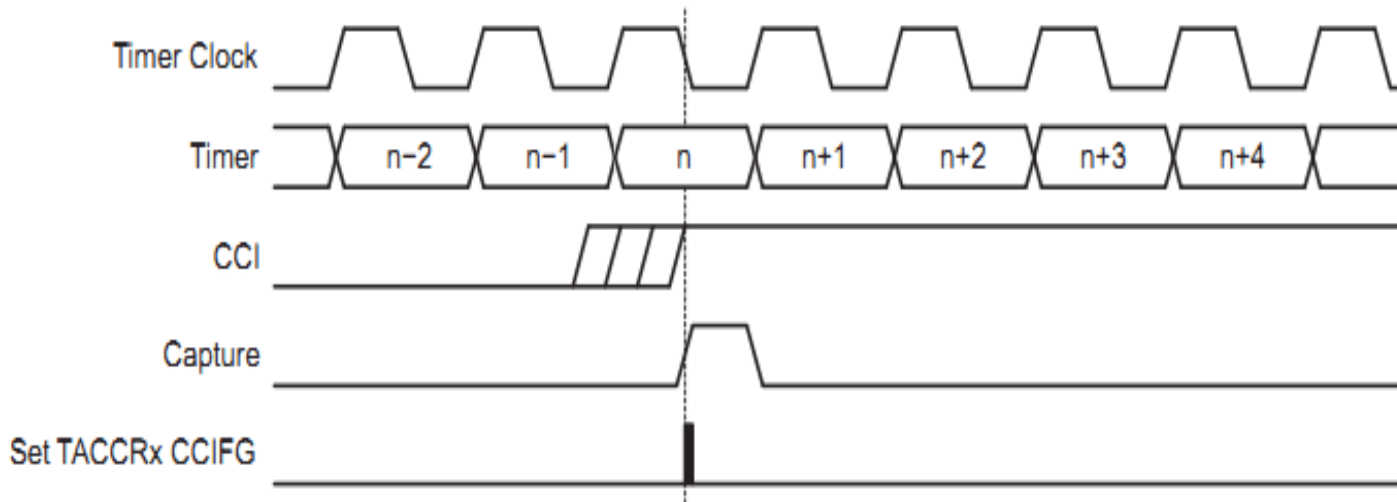


# CAPTURE

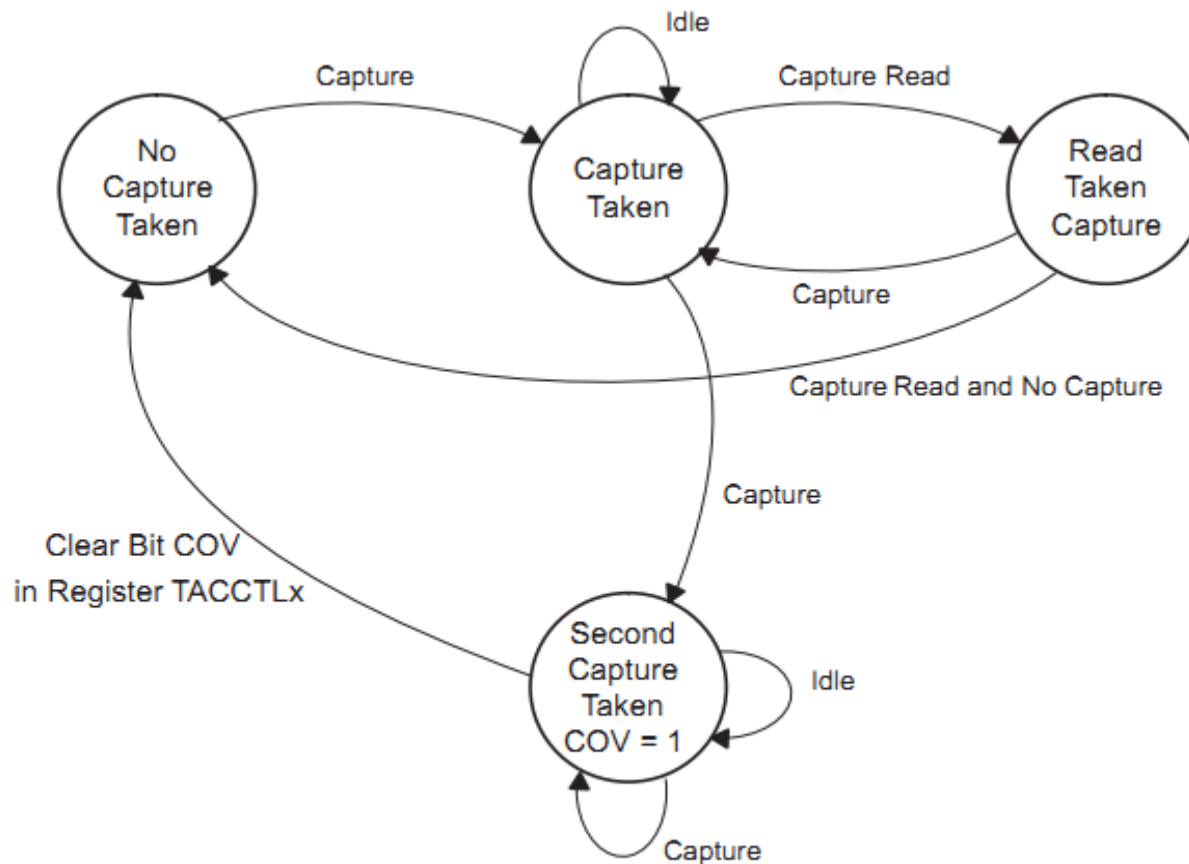
Bit CCI được gán với Input của Capture.

Capture xảy ra khi input của nó có tín hiệu kích cạnh lên hay xuống. Lúc đó :

- Giá trị  $TAxCCRy = TARx$
- CCIFG Interrupt Flag được bật.



# CAPTURE



**Figure 12-11. Capture Cycle**



# CAPTURE

Ban đầu khi chưa có event xảy ra, capture nằm ở trạng thái đầu : No Capture Taken.

Khi có event, thì capture chuyển qua trạng thái Capture Taken, lúc này giá trị thanh ghi TARx được lưu vào thanh ghi TAxCCRy, cờ được bật.

Tại đây sẽ xảy ra 2 trường hợp :

- Thanh ghi TAxCCRy được đọc, hệ thống chuyển qua trạng thái Read Taken Capture. Nếu có capture mới ngay lúc này, thì quay về trạng thái Capture Taken, ngược lại quay về trạng thái No Capture Taken.

- Thanh ghi TAxCCRy chưa được đọc nhưng 1 event mới lại xảy ra, lúc này sẽ chuyển qua trạng thái : Second Capture Taken, giá trị capture cũ bị mất, thay vào giá trị mới của event này, đồng thời bit COV được bật lên báo overflow. Để thoát ra khỏi trạng thái này, ta phải reset bit COV bằng software, khi đó sẽ quay về trạng thái No Capture Taken.



# CAPTURE

<b>CMx</b>	Bit 15-14	Capture mode	
		00	No capture
		01	Capture on rising edge
		10	Capture on falling edge
		11	Capture on both rising and falling edges
<b>CCISx</b>	Bit 13-12	Capture/compare input select. These bits select the TACCRx input signal. See the device-specific data sheet for specific signal connections.	
		00	CClxA
		01	CClxB
		10	GND
		11	V <sub>CC</sub>
<b>SCS</b>	Bit 11	Synchronize capture source. This bit is used to synchronize the capture input signal with the timer clock.	
		0	Asynchronous capture
		1	Synchronous capture
<b>SCCI</b>	Bit 10	Synchronized capture/compare input. The selected CCI input signal is latched with the EQUx signal and can be read via this bit	
<b>Unused</b>	Bit 9	Unused. Read only. Always read as 0.	
<b>CAP</b>	Bit 8	Capture mode	
		0	Compare mode
		1	Capture mode



# CAPTURE

## TAxCTL :

**CM** (bit 15 và bit 14 của thanh ghi TAxCTL) : được dùng để chọn **mode** cho Capture :

- CM\_0 : không capture ( compare mode)
- CM\_1 : capture cạnh lên của input
- CM\_2 : capture cạnh xuống của input
- CM\_3 : capture cạnh lên và xuống của input

**CCISx** : chọn source cho chân capture là loại A hay B. Việc chọn loại A hay B nó sẽ cho ngõ vào tương ứng với pin của MSP được đề cập trên datasheet. Thông thường ta chọn A.

- CCISxA : chọn loại A
- CCISxB : chọn loại B
- Vcc : chọn nguồn nội Vcc
- GND : chọn nguồn nội GND

**SCS** : đồng bộ clock timer với giá trị capture về, TI đề nghị bật bit này khi capture

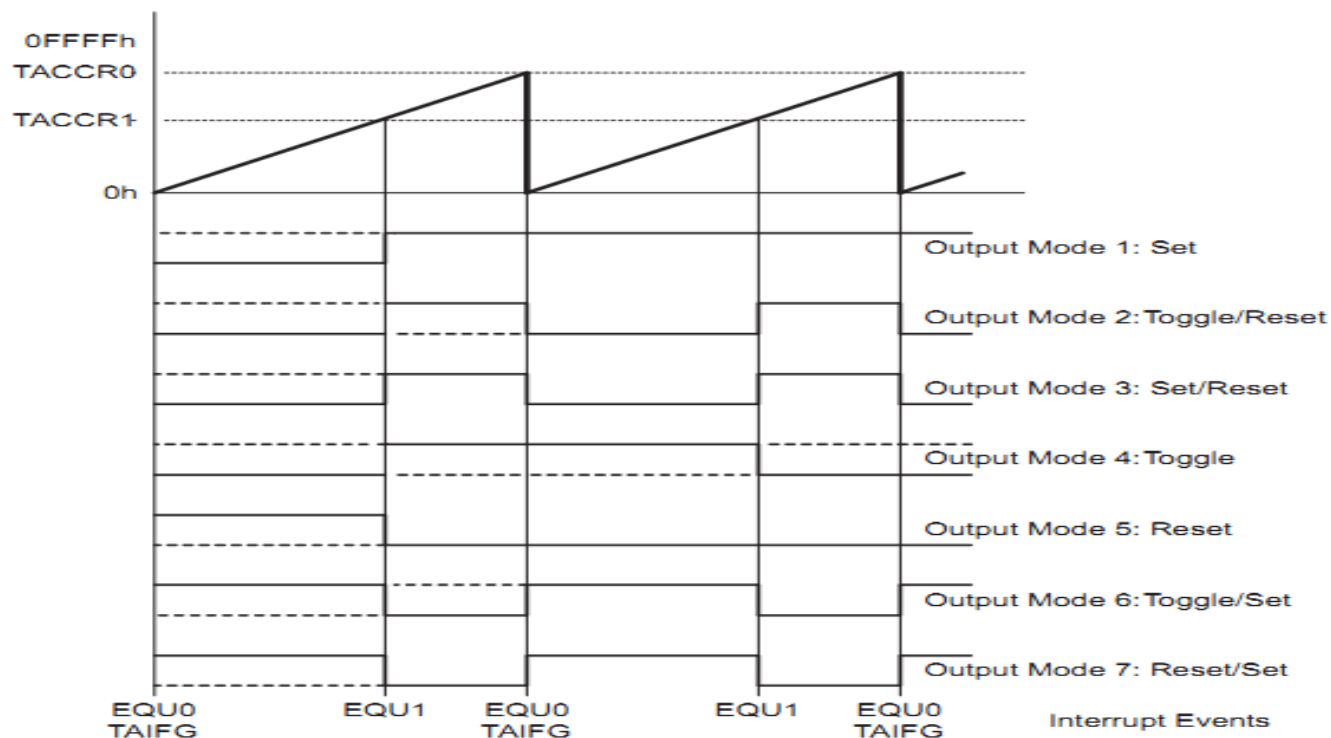
**CAP** : bit quy định bộ CCRx chức năng Capture hay Compare





# COMPARE(tt)

Ngoài chức năng so sánh khi TARx đếm tới giá trị của TACCRx, thì thực thi ngắt, ta còn có thể tạo xung PWM từ những ngõ OUT của CCRx với chu kỳ và duty cycle tùy ý.



# COMPARE

## TAxCTL :

<b>OUTMODx</b>	Bits 7-5	Output mode. Modes 2, 3, 6, and 7 are not useful for TACCR0, because EQUx = EQU0.	
		000	OUT bit value
		001	Set
		010	Toggle/reset
		011	Set/reset
		100	Toggle
		101	Reset
		110	Toggle/set
<b>CCIE</b>	Bit 4	111	Reset/set
		Capture/compare interrupt enable. This bit enables the interrupt request of the corresponding CCIFG flag.	
		0	Interrupt disabled
<b>CCI</b>	Bit 3	1	Interrupt enabled
		Capture/compare input. The selected input signal can be read by this bit.	
<b>OUT</b>	Bit 2	Output. For output mode 0, this bit directly controls the state of the output.	
		0	Output low
		1	Output high
<b>COV</b>	Bit 1	Capture overflow. This bit indicates a capture overflow occurred. COV must be reset with software.	
		0	No capture overflow occurred
		1	Capture overflow occurred



# COMPARE

## TAxCTL :

**CCIE :** cho phép ngắt khi giá trị TIMER đến tới giá trị CCRx tương ứng ở compare mode, hoặc capture được thực thi

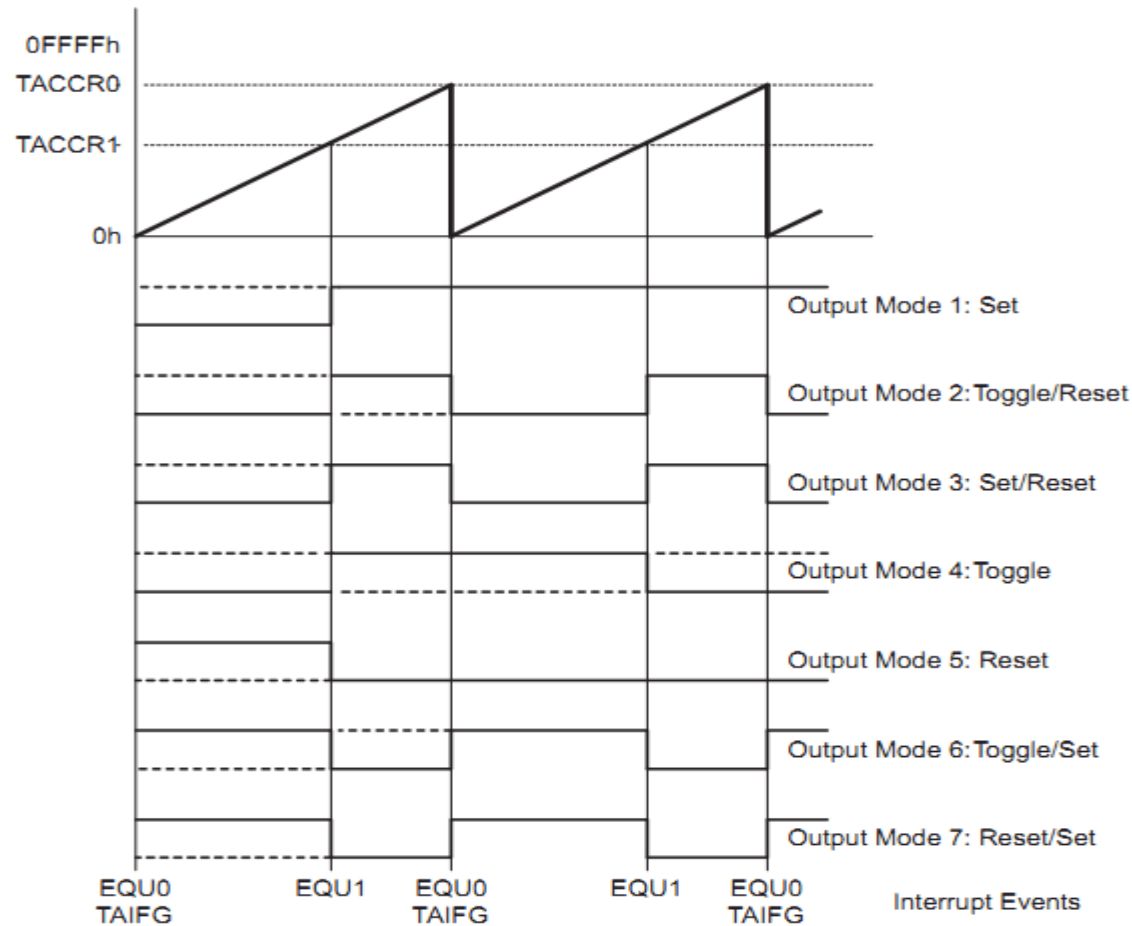
**OUTMODx :** trong chế độ Compare Mode, ta có thể xuất tín hiệu ra theo một quy luật cho trước của MSP, dựa vào giá trị của CCR0,CCR1,CCR2 và Mode của ngõ ra như trong bảng sau :

**Table 12-2. Output Modes**

OUTMODx	Mode	Description
000	Output	The output signal OUTx is defined by the OUTx bit. The OUTx signal updates immediately when OUTx is updated.
001	Set	The output is set when the timer <i>counts</i> to the TACCRx value. It remains set until a reset of the timer, or until another output mode is selected and affects the output.
010	Toggle/Reset	The output is toggled when the timer <i>counts</i> to the TACCRx value. It is reset when the timer <i>counts</i> to the TACCR0 value.
011	Set/Reset	The output is set when the timer <i>counts</i> to the TACCRx value. It is reset when the timer <i>counts</i> to the TACCR0 value.
100	Toggle	The output is toggled when the timer <i>counts</i> to the TACCRx value. The output period is double the timer period.
101	Reset	The output is reset when the timer <i>counts</i> to the TACCRx value. It remains reset until another output mode is selected and affects the output.
110	Toggle/Set	The output is toggled when the timer <i>counts</i> to the TACCRx value. It is set when the timer <i>counts</i> to the TACCR0 value.
111	Reset/Set	The output is reset when the timer <i>counts</i> to the TACCRx value. It is set when the timer <i>counts</i> to the TACCR0 value.

# COMPARE

Ví dụ như trong OUTMOD 7 : khi timer đến tới giá trị TACCR1 thì ngõ ra bị reset. Và khi timer đếm tới TACCR0, ngõ ra được set lên



## TAxCTL :

- \*\* : Các OUTMOD 2,3,6,7 không sử dụng cho OUT0 được.
  - \*\* Với OUTMOD 0, ngõ ra có giá trị phụ thuộc OUT bit trong TAxCCRL
  - \*\* Các OUTx sẽ chịu ảnh hưởng của TACCRx . (Xem thêm example code)
  - \*\* Khi thực hiện OUTMODE, thì phải config I/O pin thành OUT pin (Xem datasheet để biết cụ thể)
  - \*\* Tất cả các mode của Timer đều có thể sử dụng chức năng OUT này, giá trị ngõ ra thay đổi theo OUTMODE khi TARx = TACCRx (TACCRx đặt trước)
- Từ ngõ ra đó, ta có thể tạo được các xung PWM, có dutycycle thay đổi được  
=> điều khiển các thiết bị sử dụng nguồn DC như LED, động cơ,....
- \* Tham khảo thêm tại địa chỉ :  
<http://www.diendanti.com/showthread.php?108-MSP430-TUT-6.-T%E1%BA%A1o-xung-PWM>



## Example

Đây là chương trình minh họa cho việc setup timer cũng như khai báo các ngắt :

```
#include <msp430g2553.h>
```

```
void Timera0_init();
```

```
void Port_init();
```

```
void Timera0_init()
```

```
{
```

```
TAOCTL = TASSEL_2 + MC_1 ;           //Src Clock : SMCLK(1Mhz), UpMode
```

```
           /** TIMER đến lên 1 với mỗi 1us
```

```
TAOCCTL1 = CCIE+ OUTMOD_7; //Enable ngắt CCR1, cho phép OUT1 xuất ra ở MODE7
```

```
TAOCCR0 = 100;           // Với MODE 3, CCR0 sẽ là giá trị chu kỳ xung PWM : 100us ~ 10Khz
```

```
TAOCCR1 = 50;           // CCR1 là thời gian Ton của PWM, trong khoảng thời gian này OUT1=1
```

```
}
```

```
void Port_init()
```

```
{
```

```
P1DIR |= BIT2;           // 1.2 Output
```

```
P1OUT = 0xFF
```

```
P1SEL |= BIT2; // Chuyển chức năng I/O thành ngõ OUT, chân 1.2 là ngõ ra của OUT1 theo như trong
```

```
//datasheet của MSP430
```

```
}
```



## Example

```
void main (void)
```

```
{  
WDTCTL = WDTPW + WDTHOLD;  
Port_init();  
Timera0_init();  
_BIS_SR(LPM0_bits + GIE);  
  
}
```

\*\* Ngõ ra outmode tự động xuất xung nếu như config đúng, ta không cần thực thi ngắt trong ví dụ này. Trong thực tế, do có enable bit CCIE, nên khi TARx đến tới giá trị của TACCR1 ta vẫn có thể thực thi ngắt như bài trước-Ngắt CCR1



PAY IT FORWARD



payitforward.edu.vn