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 sẻ 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

payitforward.edu.vn

MSP430 COURSE

LESSON 2
TIMER

Training document for C9 course

TIMER

-Timer là ngoại vi cơ bản của vi xử lý
-Có chức năng định thì, capture, compare
-MSP430G2553 có 2 TimerA 16bit -A0 và A1
-Hỗ trợ đa chức năng:Capture /Compare,ngõ ra PWM và định thì từng -khoảng thời gian.
-Có các chế độ ngắt Timer/Counter và Capture/Compare.
-Có thể lựa chọn để hoạt động với các nguồn xung Clock khác nhau.

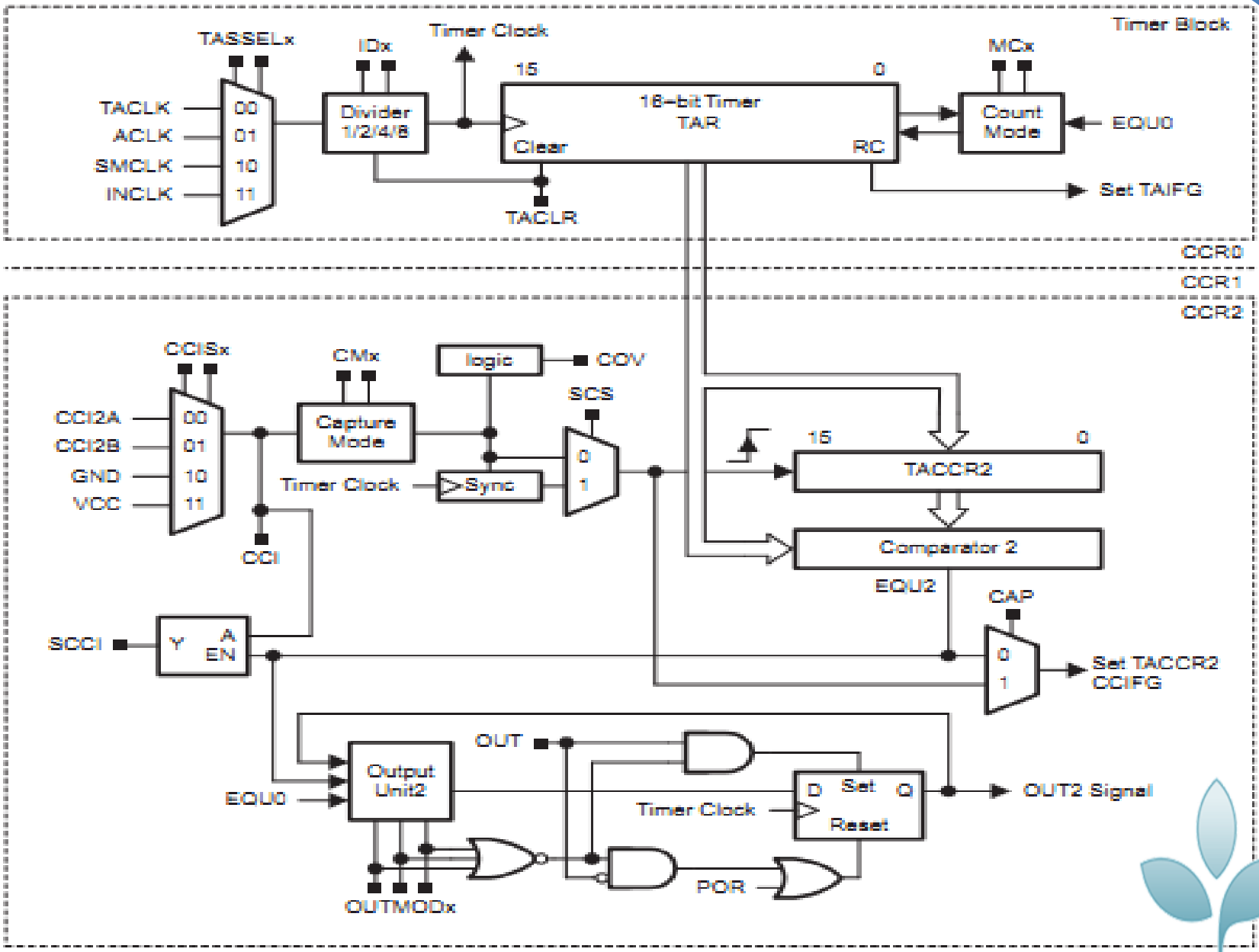


Figure 12-1. Timer_A Block Diagram

TIMER

Clock Source Timer :

• TACLK (Timer A clock): Là một nguồn xung clock ngoại được đưa vào chân P1.0 của MSP430G2231

• ACLK (Auxiliary clock): Là một nguồn clock bổ sung cho Timer A, chỉ được cấp nguồn từ LFXTCL, thường dùng cho hệ thống phụ với clock hoạt động thấp để tiết kiệm năng lượng.

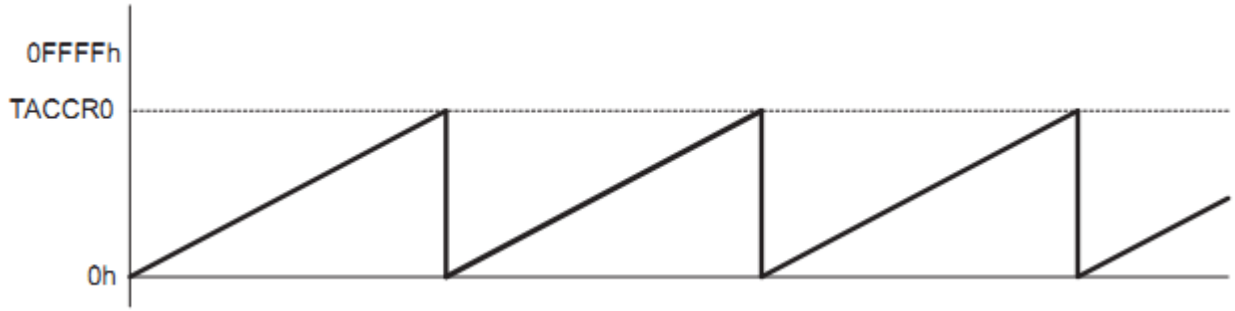
• SMCLK (Submaster clock): thường dùng cho ngoại vi, lấy nguồn từ DCO hoặc XT2.

TIMER

1. Timer Counter :

Các chế độ hoạt động :

• Up : Chế độ đếm lên của Timer A. Giá trị của thanh ghi TAR (Timer A register) sẽ tăng liên tục từ 0 cho đến một giá trị được định sẵng trong thanh ghi TACCR0 rồi reset về giá trị 0,quá trình hoạt động cứ tiếp tục như vậy.



TIMER

1. Timer Counter :

Các chế độ hoạt động :

• Continous: Chế độ đếm lên liên tục của Timer A.Trong một chu kỳ hoạt động giá trị của thanh ghi TARsẽ tăng liên tục từ 0 cho đến 0FFFFh rồi reset về giá trị 0:

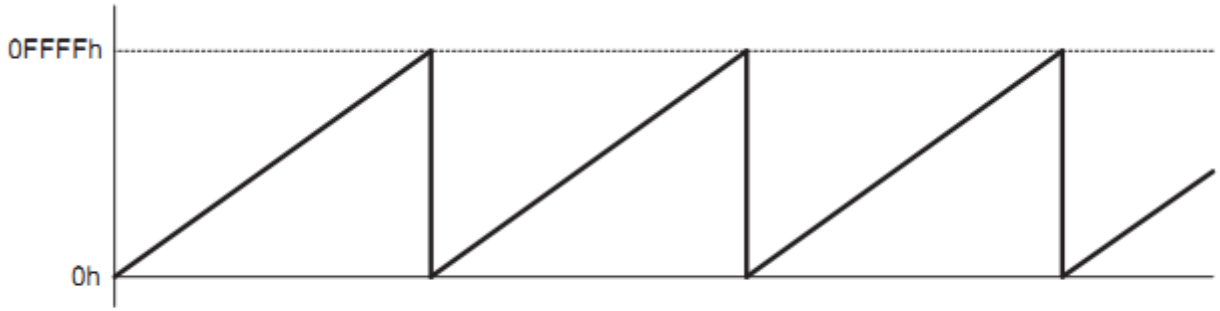


Figure 12-4. Continuous Mode

TIMER

1. Timer Counter :

Các chế độ hoạt động :

• Up/Down: Chế độ đếm lên/xuống của Timer A. Trong một chu kỳ hoạt động giá trị của thanh ghi TAR sẽ tăng liên tục từ 0 cho đến TACCR0 rồi giảm dần về giá trị 0.

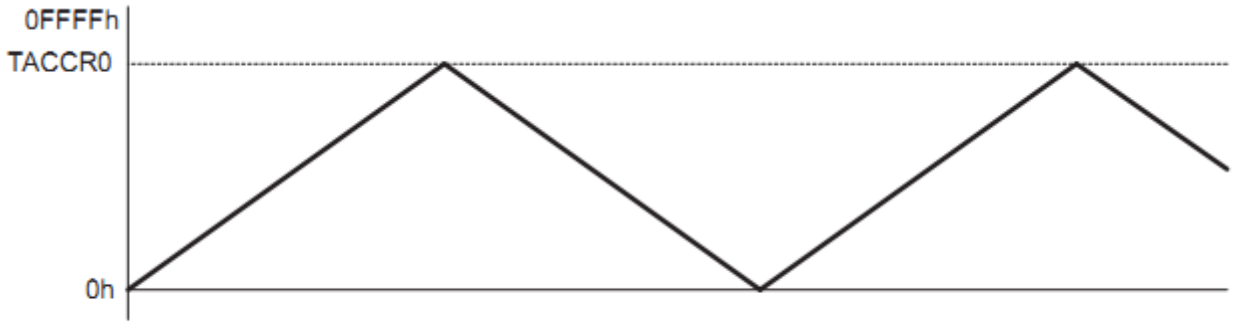


Figure 12-7. Up/Down Mode

TIMER

2. Các thanh ghi config:

-TAxCTL: setup timer

-TARx: thanh ghi giá trị đếm của timer

-TACCRx: thanh ghi giá trị caputre/compare

-TAxCCTL: setup capture/compare mode

-TAIV: thanh ghi ngắt

TAxCTL :

Unused	Bits 15-10	Unused
TASSELx	Bits 9-8	Timer_A clock source select
		00 TACLK
		01 ACLK
		10 SMCLK
		11 INCLK (INCLK is device-specific and is often assigned to the inverted TBCLK) (see the device-specific data sheet)
IDx	Bits 7-6	Input divider. These bits select the divider for the input clock.
		00 /1
		01 /2
		10 /4
		11 /8
MCx	Bits 5-4	Mode control. Setting MCx = 00h when Timer_A is not in use conserves power.
		00 Stop mode: the timer is halted.
		01 Up mode: the timer counts up to TACCR0.
		10 Continuous mode: the timer counts up to 0FFFFh.
		11 Up/down mode: the timer counts up to TACCR0 then down to 0000h.
Unused	Bit 3	Unused
TACLR	Bit 2	Timer_A clear. Setting this bit resets TAR, the clock divider, and the count direction. The TACLR bit is automatically reset and is always read as zero.
TAIE	Bit 1	Timer_A interrupt enable. This bit enables the TAIFG interrupt request.
		0 Interrupt disabled
		1 Interrupt enabled
TAIFG	Bit 0	Timer_A interrupt flag
		0 No interrupt pending
		1 Interrupt pending

TAxCTL :

TASSEL (bit 8 và bit 9 của thanh ghi TAxCTL) : được dùng để chọn source clock cho TIMER : ứng với 3 giá trị clock có thể sử dụng trong MS430G2553 TI đã quy định sẵn các giá trị này bằng tên gọi nhớ :

-TASSEL_1 : chọn clock TACLK(từ chân P1.0)

-TASSEL_2 : chọn clock ACLK(clock dao động thấp của MCU)

-TASSEL_3 : chọn clock SMCK(sub main clock được lấy từ bộ DCO , mặc định là 1Mhz)

Để chọn clock cho TIMER ta có thể config như sau :

TA0CTL = TASSEL_3;(chọn SMCLK)

IDx (bit 7 và bit 6 của thanh ghi TAxCTL) : quy định giá trị chia tần số cho Clock được chọn trước khi vào Timer Clock

Chia tần số qua Idx sẽ làm giảm giá trị tần số đếm của Timer. Ví dụ ta chọn source clock cho TIMER là 1Mhz và ID là 2 thì tần số TimerClock sẽ là 0.5 Mhz. Ti cũng đã quy định tên gọi nhớ cho các ID như sau :

ID_0 : chia tần số cho 1

ID_1: chia tần số cho 2

ID_2 : chia tần số cho 4

ID_3 : chia tần số cho 8

Ta muốn chia tần số source clock cho 2 thì ta ghi như sau :

TA0CTL = TASSEL_3 +ID_1;

TAxCTL :

MC (bit 5 và bit 4 trong thanh ghi TAxCTL) : chọn 1 trong 4 mode cho Timer hoạt động như đã giới thiệu ở trên.

MC_0 : Timer Stop

MC_1 : Up mode, Timer đếm tới CCR0(cần phải set giá trị CCR0 trước)

MC_2 : Continuous mode, Timer đến tới giá trị tối đa 16 bit : 65536

MC_3 : Up/Down mode, Timer đến tới CCR0 rồi đếm lùi về 0.

Để chọn mode cho timer ta có thể :

TA0CTL = TASSEL_3 + ID_1 + MC_2;

TACLR (bit 2 trong thanh ghi TAxCTL) : Reset giá trị thanh ghi đếm TAxR khi set bit này lên 1 . Bit tự động clear sau khi reset.

TAIE (bit 1 trong thanh ghi TAxCTL): cho phép ngắt tràn TIMER , ngắt tràn TIMER chỉ xảy ra ở 2 mode : Continuous Mode (TAxR từ 655365 về 0), và Up/Down Mode(TAxR từ 1 về 0)

TAIF (bit 0) cờ ngắt, khi cờ bật lên 1 nếu có enabel TAIE , MCU sẽ nhảy vào vector ngắt của tràn TIMER. Việc thực hiện chương trình ngắt sẽ reset cờ.

Khai báo chế độ Continuous Mode có ngắt tràn TIMER :

TA0CTL = TASSEL_2 + ID_2 + MC_2 + TAIE;

** Việc sử dụng các tên gợi nhớ của TI để dễ dàng hơn trong việc code, thực tế các tên đó đã được gắn sẵn những giá trị tương ứng với trên data sheet.

TASSEL_2 = 2*0x100u (tương ứng với 0b 0010 0000 0000 – Bit 9 bằng 1 , bit 8 bằng 0)

ID_2 = 2*0x40u (tương ứng với 0b 0000 1000 0000 – Bit 7 bằng 1 , bit 6 bằng 0)

<Coi thêm trong file MSP430G2553.h để biết thêm >

TAxCCTL :

CCIE	Bit 4	Capture/compare interrupt enable. This bit enables the interrupt request of the corresponding CCIFG fla
		0 Interrupt disabled
		1 Interrupt enabled
CCI	Bit 3	Capture/compare input. The selected input signal can be read by this bit.
OUT	Bit 2	Output. For output mode 0, this bit directly controls the state of the output.
		0 Output low
		1 Output high
COV	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
CCIFG	Bit 0	Capture/compare interrupt flag
		0 No interrupt pending
		1 Interrupt pending

TAxCCTL :

CCIE : cho phéo ngắt khi giá trị TIMER đến tới giá trị CCRx tương ứng.

Nếu ta muống TIMER đếm đến 30000,40000 thì thực hiện ngắt trong chết độ Continuous Mode ta thực hiện lệnh sau :

```
TA0CTL = TASSEL_2 + ID_2 + MC_2 + TAIE;      // Cài đặt Timer
TA0CCR0= 30000;                              // Đặt giá trị mong muốn vào CCR0
TA0CCR1 = 40000;                             // Đặt giá trị mong muốn vào CCR1
TA0CCTL0 = CCIE;                             // Enable ngắt cho CCR0.
TA0CCTL1 = CCIE;                             // Enable ngắt cho CCR1.
```

Ở dòng khai báo trên, ta cho TIMER chạy ở chế độ Continuous Mode (chạy từ 0-65536)

Khi giá trị TIMER đạt 30000 thì sẽ nhảy vào chương trình phục vụ ngắt CCR0,

Trong lúc đó TIMER sẽ đếm tiếp lên 40000 và MCU sẽ nhảy vào chương trình phục vụ ngắt CCR1,

Và khi TIMER chạy đến 65536 và reset về 0 thì MCU thực hiện ngắt tràn TIMER

(CCR2 hoạt động tương tự.)

** Ở Up Mode : TIMER chỉ đếm tới CCR0, khi đếm tới đó thì ngắt CCR0 được thực thi

Up/Down Mode : TIMER đếm tới CCR0=> ngắt CCR0, và khi TIMER đếm lùi từ 1 về 0 => ngắt tràn TIMER

TAIV :

TAIVx Bits 15-0 Timer_A interrupt vector value

TAIV Contents	Interrupt Source	Interrupt Flag	Interrupt Priority
00h	No interrupt pending	-	
02h	Capture/compare 1	TACCR1 CCIFG	Highest
04h	Capture/compare 2[1]	TACCR2 CCIFG	
06h	Reserved	-	
08h	Reserved	-	
0Ah	Timer overflow	TAIFG	
0Ch	Reserved	-	
0Fh	Reserved	-	Lowest

Thanh ghi ngắt của TIMER, khi cờ ngắt của CCR1 , CCR2 hoặc tràn TIMER (TAIF) bật lên thì thanh ghi này sẽ có 1 giá trị tương ứng như trong bảng.

- 3 ngắt này đều cùng nằm trong 1 VECTOR ngắt TIMER0_A1_VECTOR
- Riêng ngắt CCR0 được nằm trong 1 VECTOR riêng : TIMER0_A0_VECTOR

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()
{
TA0CTL = TASSEL_2 + MC_2 + ID_3+ TAIE ;//Src Clock : SMCLK(1Mhz), Divided by 8, Continuous Mode,
Enable Interrupt overflow     **   TIMER đến lên 1  với mỗi 8us
TA0CCR0 = 31250;      // Ngắt CCR0 khi đếm tới 31250 tương ứng 0.25s so với giá trị đầu 0 của TIMER
TA0CCR1 = 62500;      // Ngắt CCR1 khi đếm tới 62500 tương ứng 0.5s, so với giá trị đầu 0 của TIMER
TA0CCTL0=CCIE;        //Enable ngắt CCR0
TA0CCTL1 = CCIE;      //Enable ngắt CCR1
}
void Port_init()
{
P1DIR |= BIT0 + BIT1+BIT2;                    // 1.0,1.1,1.2 Output
P1OUT = 0xFF;                                 //Tắt hết LED
}
```

Example

Khai báo ngắt :

```
#pragma vector=TIMER0_A1_VECTOR          //Vector ngắt của CCR1,CCR2, TAIF
__interrupt void TAIV_Interrupt (void)   //Tên ct ngắt tự chọn
{
switch(TA0IV)                            //Dùng lệng switch để chọn chương trình ngắt tương ứng với cờ ngắt

{
case 0x02:                               //CCR1 báo ngắt, TA0IV giá trị 0x02
{
P1OUT ^= BIT0;                           //Đảo giá trị port 1.0 khi vào ngắt CCR1
TA0CCR1+=62500;                          //Cộng CCR1 thêm 1 lượng đúng bằng 0.5s để lần ngắt tiếp theo CCR1 sẽ cách lần ngắt này đúng 0.5s<Chỉ áp dụng trong Continuous Mode>


break;
}
case 0x0A:                               //Tràn TIMER báo ngắt, TA0IV giá trị 0x0A
{
P1OUT ^= BIT1;                           //Đảo giá trị port 1.1 khi vào ngắt Tràn TIMER , do giá trị tràn TIMER là 65536 ~ 0.524s nên lúc đầu ta sẽ thấy 1.0 và 1.1 như sáng tắt cùng lúc, nhưng sau 1 lúc các bạn sẽ thấy 2 LED này lệch pha( nếu mắt đáp ứng tần số tốt ^o^)
break;
}
}
}
```

Example

```
#pragma vector=TIMER0_A0_VECTOR          //Vector ngắt của CCR0
__interrupt void CCR0_Interrupt (void)   //Tên ct ngắt tự chọn

{
P1OUT ^= BIT2;                           //Port 1.2 đảo giá trị khi TIMER đếm tới 31250, do TIMER đếm
                                         khá nhanh, nên bạn sẽ thấy P1.2 nhấp nháy tần số gấp đôi P1.1
}
 void main (void)
{
WDTCTL = WDTPW + WDTHOLD;
Port_init();
Timera0_init();
_BIS_SR(LPM0_bits + GIE);                //Nếu không còn gì làm nữa thì cho CPU nó sleep, dừng chương
trình ở đây + Enable ngắt toàn cục, GIE là biến quan trọng để các ngắt được thực thi, các Low-Power-Mode có thể
tham khảo thêm trong datasheet.
}
```

Ta đang sử dụng TIMERA0 của MSP nên tất cả các thanh ghi config đề có prefix TA0xxx, MSP430G2553 còn 1 TIMERA nữa đó là TIMERA1, mọi config đều như TIMERA0 chỉ thay đổi prefix thành TA1xxx

PAY IT FORWARD

payitforward.edu.vn