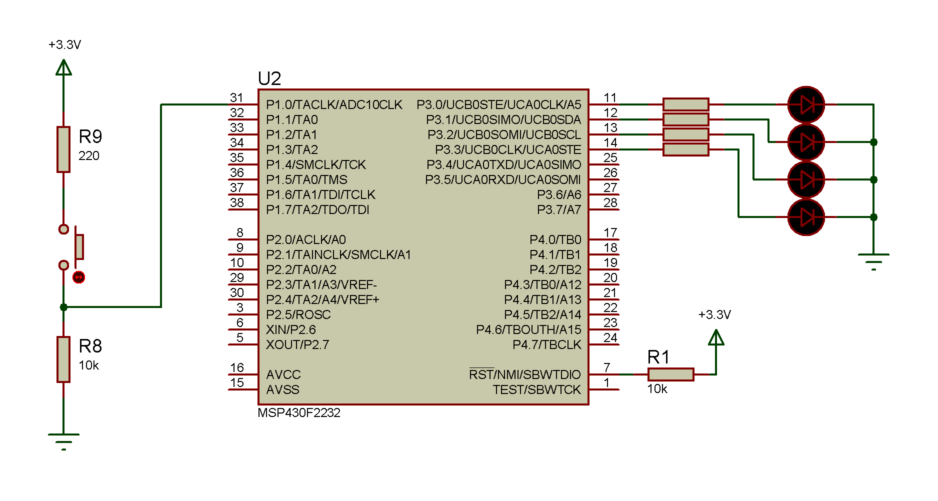
**MSP430 Uyg.16 – TimerA Uygulaması\_2**

Ferudun GÖKCEGÖZ, 06 Ağustos 2011, Cumartesi

Merhaba arkadaşlar. 16. dersimizle yine sizlerle birlikteyiz. Bu dersimizde geçenki derste söylediğimiz gibi TimerA ile ilgili bir uygulama yapacağız. Uygulamamızda MSP430 a bir adet buton bağlı olacak. Timer ın clock kaynağını harici olarak seçeceğiz. Bu buton ile timer modülüne clock sinyallerini sağlayıp, BCD sayıcı uygulaması yapacağız. Aslında TimerA yı counter(sayıcı) modunda kullanmış olacağız. Hadi bakalım başlayalım.

**İşte devre şemamız…**



Yazımızın başında söylediğimiz gibi MSP430 un P1.0 portuna bir adet buton bağlanmış durumda. Ve P3 portunada 4 adet ledimizi BCD sayıcı kullanmak için bağlanmış durumda. İsterseniz yazılım üzerinden devam edelimki, daha anlaşılır olsun.

|  |  |
| --- | --- |
| **[Source code](http://www.mcu-turkey.com/msp430-uyg-16/" \l "codesyntax_1" \o "Click to show/hide code block)** | **[http://www.mcu-turkey.com/wp-content/plugins/wp-synhighlight/themes/default/images/code.png](http://www.mcu-turkey.com/msp430-uyg-16/#codesyntax_1)** **[http://www.mcu-turkey.com/wp-content/plugins/wp-synhighlight/themes/default/images/printer.png](http://www.mcu-turkey.com/msp430-uyg-16/#codesyntax_1)** **[http://www.mcu-turkey.com/wp-content/plugins/wp-synhighlight/themes/default/images/info.gif](http://www.mcu-turkey.com/wp-content/plugins/wp-synhighlight/About.html)** |

*#include "io430.h"*

*#include "in430.h"*

unsigned char i;

void main( void )

{

WDTCTL = WDTPW + WDTHOLD;

DCOCTL=CALDCO\_1MHZ;

BCSCTL1=CALBC1\_1MHZ;

P3OUT = 0x00;

P3DIR = 0x0F;

P1DIR\_bit.P0 = 0;

P1SEL\_bit.P0 = 1;

TACTL=TASSEL\_0 + MC\_1;

TACCR0=1;

TACCTL0=CCIE;

\_BIS\_SR(GIE);

for(;;);

}

*#pragma vector=TIMERA0\_VECTOR*

\_\_interrupt void Timer\_A0 (void)

{

i++;

P3OUT = i;

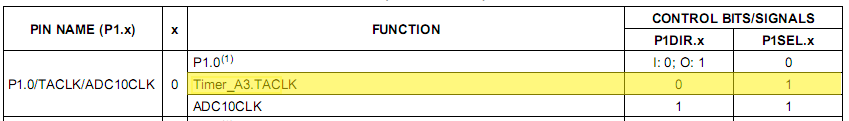
TAR = 0x0000;

if(i==15)i=0;

}

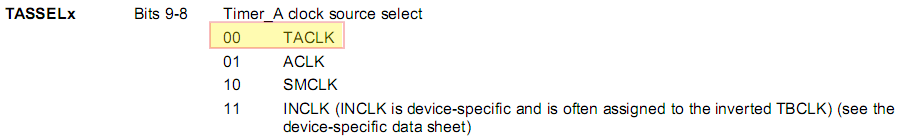
Yazılımı baştan itibaren inceleyecek olursak, ilk olarak unsigned char türünden “i” isimli bir değişken tanımlanmış. Bu değişkenimizi Timer kesme vektöründe ledlerimize göndereceğimiz sayıyı tutması için kullanacağız. Devam edecek olursak, main fonksiyonu içerisinde ise ilk olarak WDT ve Clock ayarları yapılmış. Daha sonra ise port ayarlarına geçilmiş.

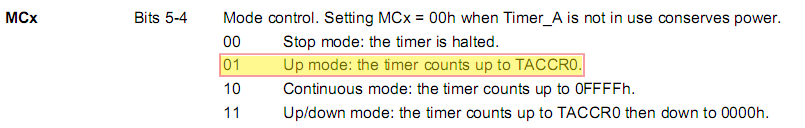
Port ayarlarında ise ilk olarak ledlerin bağlı olduğu P3 portu temizlenmiş ve ilk 4 biti çıkış olarak şartlanmıştır. Daha sonra ise butonumuzun bağlı olduğu P1.0 pini giriş olarak tanımlanmıştır. Yalnız bu port için ekstra bir ayar yapılmıştır. Devre şemasında tekrardan bakılırsa, P1.0 pini alternatif giriş olarak, TimerA nın harici clock girişidir. Bu nedenle bu pin TimerA nın clock girişi olarak kullanılacaktır. Bu yüzdendende bu pinin PxSEL kaydedicisindeki değeri Lojik 1 yapılmıştır.



Yukarıdaki tablodanda görüldüğü gibi P1.0 pinini TimerA için harici clock girişi olarak kullanacaksak, ilgili P1DIR.x kaydedicisi Lojik0, P1SEL.x kaydedicisi Lojik 1 olmalıdır.

Yazılıma devam edecek olursa sonraki satırlarda timer ayarları yapılmıştır. TimerA Control Register ı için, TASSEL\_0 ve MC\_1 seçilmiştir. Bu bitler içinde durumları aşağıdaki tablodan görebiliriz.





Daha sonra ise TimerA CCR değerine “1″ yazılmıştır. Bunun anlamı, butona her basılışta timer ın saydığı değer 0 dan 1 e gidecek ve kesme üretilecektir. Bu değeri “5″ yapsaydık timer her 5 basışta kesme üretecektir manasına gelmektedir. Daha sonra ise timer için kesme aktif duruma getirilmiş ardındanda genel kesmelere izin verilmiş ve sonsuz döngüye girilmiştir.

Bundan sonra timer kesme alt programını inceleyecek olursak, ilk olarak global olarak tanımladığımız i değişkeni bir artırılmış.

Hatırlarsınız ki, global olarak tanımlanan değişkenlerin ilk değerleri default olarak sıfırdır.

i değeri bir artırıldıktan sonra P3 portuna gönderilmiştir. Ardındanda TimerA ın sayma register ı (TAR : TimerA Register) temizlenmiştir. Son olarak ta i değerinin 15 e ulaşıp ulaşmadığı kontrol edilmiş, eğer ulaşmışsa sıfırlanması sağlanmıştır. Sizde IAR üzerinde kodları derleyip, proteus üzerinde simulasyonunu yaptığınızda göreceksinizki, butona her basıldığında, P3 portuna bağlı ledler binary olarak artacaktır. 15 ulaştığında da sıfırlanacaktır.

Böylelikle bir dersimizin daha sonuna gelmiş olduk. Gelecek dersimizde, TimerB ile ilgili örnek yapacağız. Bir dahaki dersimizde görüşmek üzere. Şimdilik Hoşçakalın..

***Ferudun GÖKCEGÖZ***

***fgokcegoz@yahoo.com***