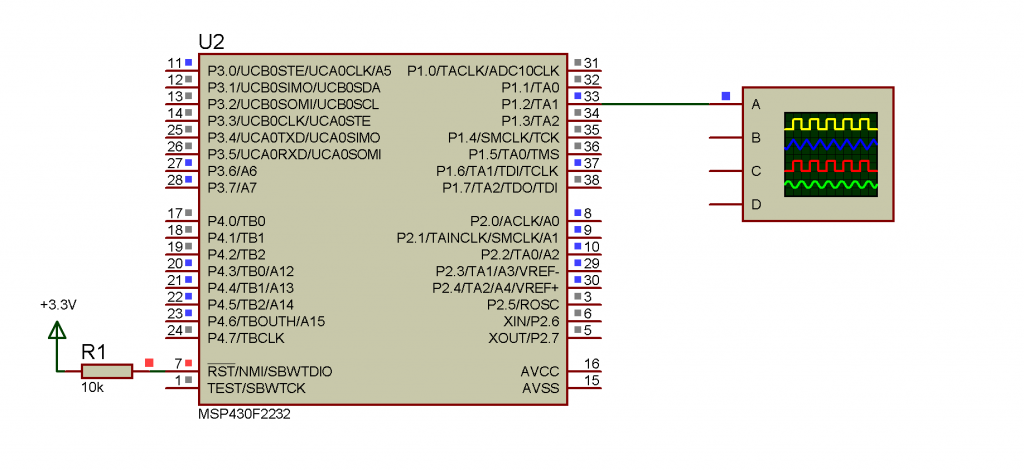
**MSP430 Uyg.21 – PWM Uygulaması\_1**

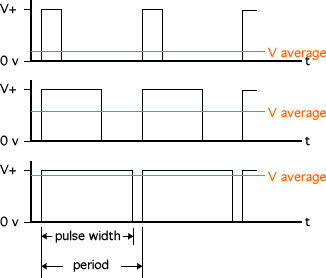
Ferudun GÖKCEGÖZ, 08 Ağustos 2011, Pazartesi

Merhaba arkadaşlar. Yeni bir dersimizle yine sizlerle birlikteyiz. Daha önce söylediğimiz gibi bu dersimizde PWM(pulse width modulation) uygulaması yapacağız. Bildiğimiz gibi en basit ifadeyle,  PWM dalgası bir adet kare dalga demektir. Kare dalganın 0-Π arasında Lojik1, Π-2Π arasında ise Lojik0 durumdadır. Yani doluluk oranı(duty cycle) %50 dir. PWM dediğimiz şey ise az önce bahsettiğimiz duty cycle değerinin değiştirilmesiyle oluşan dalgalara denir. Yalnız burada dikkat etmemiz gerekirki, duty cycle değeri değişse bile frekans sabittir. Bu uygulamamızda ise MSP430 un timer birimini kullanarak PWM sinyalleri üreteceğiz.

**İlk olarak devre şemasını verelim.**

****

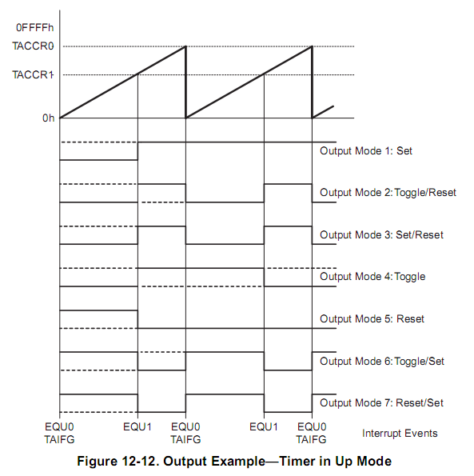
Bu uygulamamızda devre şemamızın pek bir önemi yok. Sadece TimerA birimi sayesinden üreteceğimiz PWM sinyallerini görmek için P1.2 pinine osiloskop konulmuştur.

****

Yukarıdaki grafikte 3 adet PWM sinyali görülmektedir. Yazının başında bahsettiğimiz gibi PWM sinyalleri; frekansı sabit tutup, darbe genişliğini değiştirilerek elde edilirler. Sonuç olarak elde edilen sinyalin “Ortalama Gerilimi” (Mean Value – V\_average) değişir. Bu şekilde bir sinyal bir dc motor sürücüsüne uygulandığında, çok açık bir şekilde görülecektirki, uygulanan gerilimin ortalama değeri değiştiğinden, dc motorunda endüvi gerilimi azalacak, dolayısıylada hızı azalacaktır.

Bir başka örnek olarak, yine bu şekilde bir sinyal ile Led sürer isek, darbenin genişliği arttıkça parlaklık artacak, azaldığında ise parlaklık azalacaktır. Yine bu tür sinyallerin kullanıldığı yerlere örnek verecek olursak, DC-DC Dönüştürücülerde (buck-boost-buck/boost converter vb.) yine bu tür sinyallerle bobin anahtarlanarak çıkış gerilimi değiştirilmektedir. Yani bu tür sinyallerin birçok kullanım yeri vardır. Endüstriyel uygulamalarda birçok yerde kullanılırlar.

Geçelim MSP430 larda PWM sinyallerinin nasıl üretileceğine. Aşağıdaki grafikleri inceleyecek olursak, TimerA biriminin CCR0 kaydedicisine yüklenen değer ile PWM sinyalinin frekansını, CCR1 kaydedicisine yüklenen değer ilede doluluk oranını ayarlamaktayız. Bu arada aşağıdaki grafik Up Mode içindir. Ayrıca aşağıdaki grafikte çeşitli mod lar bulunmaktadır. Uygulamamıza göre istediğimiz şekilde sinyal elde edebiliriz. PWM sinyali için uygun olan mod Mode6 veya Mod7 dir.



Şimdide uygulamaya ait yazılımı verip üzerinde konuşmaya başlayalım isterseniz.

|  |  |
| --- | --- |
| **[main.c](http://www.mcu-turkey.com/msp430-uyg-21/" \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-21/#codesyntax_1)** **[http://www.mcu-turkey.com/wp-content/plugins/wp-synhighlight/themes/default/images/printer.png](http://www.mcu-turkey.com/msp430-uyg-21/#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"

void main( void )

{

WDTCTL = WDTPW + WDTHOLD;

DCOCTL=CALDCO\_1MHZ;

BCSCTL1=CALBC1\_1MHZ;

P1DIR = BIT2;

P1SEL = BIT2;

TACCR0 = 1000;

TACCR1 = 600;

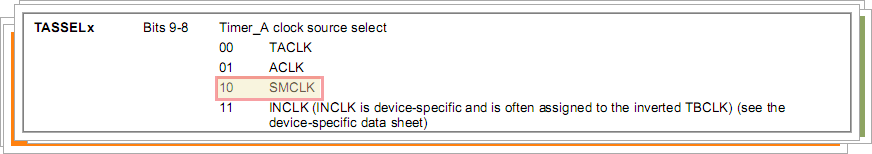
TACCTL1 = OUTMOD\_7;

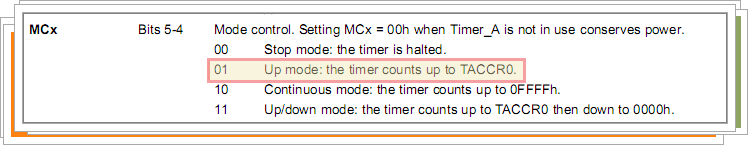
TACTL = TASSEL\_2 + MC\_1 + ID\_0;

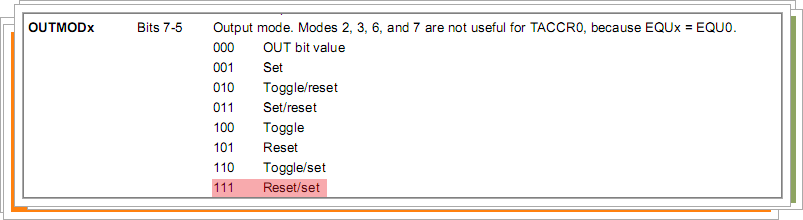
for(;;);

}

Yazılımımız gayet sade. Sadece PWM sinyalleri için gerekli timer ayarlarını yapıp mikrodenetleyiciyi sonsuz döngüde bekletiyoruz. Timer ayarlarından bahsedecek olursak;





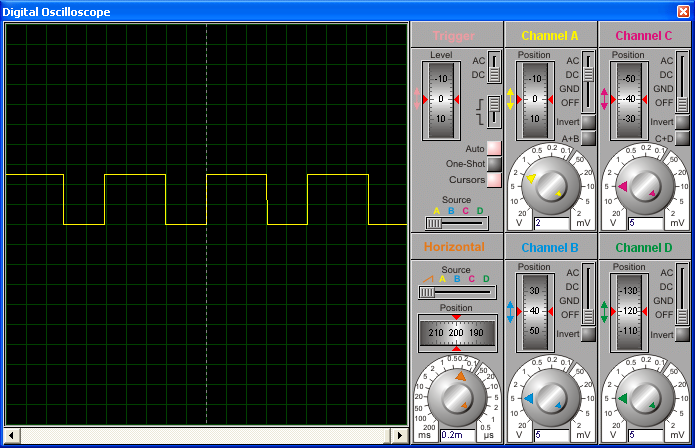


TimerA gelen clock darbe kaynağı olarak SMCLK seçilmiştir. Buna göre timer a gelen darbelerin periyodu 1 μsn dir. TACCR0 kaydedicisine yükleyeceğimiz değer ile PWM sinyallerinin periyodunu belirlediğimizi söylemiştik. Buna göre TACCR0 kaydedicisine 1000 değeri yüklenmiş.

Yani T\_pwm = 1000\* 1μsn = 1 msn , F\_pwm = 1/1msn = 1kHz eder.

Darbe genişliğini ise TACCR1 kaydedicisine yüklenen değer ile belirleyeceğimizi söylemiştik. TACCR1 e yüklenen değer ise 600 dür. Buna göre 600 \* 1 μsn = 0.6 msn darbe genişliğine sahibiz. Yani doluluk oranı %60 tır.

Son olarak ise TimerA , UP Mode da çalıştırılmış ve OutputMod7 seçilmiştir. Buna göre çıkış sinyallerimizi osiloskop üzerinden gözlersek;



Bu uygulamayı LauncPad üzerinde de kolayca deneyebilirsiniz. Ancak bu konuda size bir tavsiyem olacak. TimerA için clock kaynağını SMCLK seçmeyin. LaunchPad üzerine kristali lehimlediyseniz, ACLK olarak seçin. Kristal gayet stabil darbeler ürettiği için çok daha iyi sonuç alabilirsiniz. Benim bu konu hakkında söyleyeceklerim bu kadar. Bir dahaki dersimizdede yine bir PWM uygulaması yapacağız. Bir dahaki dersimizde görüşmek üzere. Şimdilik Hoşçakalın..

***Ferudun GÖKCEGÖZ***

***fgokcegoz@yahoo.com***