**MSP430 Uyg.20 – WatchDog Timer Uygulaması\_2**

Ferudun GÖKCEGÖZ, 07 Ağustos 2011, Pazar

Merhaba arkadaşlar. 20. dersimizde yine sizlerle birlikteyiz. Bu dersimizdede geçenki derste söylediğimiz gibi Watchdog Timer ile ilgili bir uygulama yapacağız. Geçen dersimizdeki uygulamamızda, WDT birimini watchdog mod da yani wdt periyodu dolduğunda mikrodenetleyiciye reset attırma modunda çalıştırdık. Bu dersimizdeki uygulamamızda ise yine aynı mod da, fakat wdt sayıcısının temizlendiğini bu nedenle hiç reset durumu oluşmadığı bir örnek yapacaktım. Ama Watchdog Timer biriminin “interval mode” isminde bir modu var. Bu mod da, wdt sayıcısında taşma oluştuğunda wdt kesme vektörüne dallanılıyor. Yine aynı şekilde taşma oluşmadan, sayıcı resetlenirse kesme vektörüne dallanılmıyor. Bu dersimizde bu mod ile ilgili uygulama yapacağız. Hadi bakalım başlayalım..

Bu uygulamamızıda geçenki uygulamamız gibi launchpad üzerinde gerçekleştireceğiz. Proteus üzerinde wdt uygulamasında daha önce bir uygulamamızda olduğu gibi problem var. O yüzden launchpad üzerinde uygulamayı yapıp video olarak paylaşmak daha iyi olacağını düşündüm. İsterseniz ilk olarak uygulamaya ait yazılımı verip ardından da anlatıma geçelim.

|  |  |
| --- | --- |
| **[main.c](http://www.mcu-turkey.com/msp430-uyg-20/" \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-20/#codesyntax_1)** **[http://www.mcu-turkey.com/wp-content/plugins/wp-synhighlight/themes/default/images/printer.png](http://www.mcu-turkey.com/msp430-uyg-20/#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"

#define LED0 P1OUT\_bit.P0

void delay(void)

{

unsigned short i=10000;

while(i--);

}

void main( void )

{

WDTCTL = WDT\_ADLY\_1000;

IE1\_bit.WDTIE = 1;

IFG1\_bit.WDTIFG = 0;

DCOCTL=CALDCO\_1MHZ;

BCSCTL1=CALBC1\_1MHZ;

P1OUT = 0x00;

P1DIR = 0x01;

\_BIS\_SR(GIE);

for(;;);

}

#pragma vector=WDT\_VECTOR

\_\_interrupt void WDT\_ISR (void)

{

LED0 = 1; delay();

LED0 = 0; delay();

LED0 = 1; delay();

LED0 = 0;

IFG1\_bit.WDTIFG = 0;

}

Yazılımda ilk olarak, LED değişkeni P1.0 pininde tanımlanmıştır. Daha sonra ise wdt kesme alt programında led e flash yaptırmak için gerekli bir gecikme alt programı yazılmıştır. Daha sonra ise main fonksiyonuna geçilmiştir. Main fonksiyonunda ilk olarak WDT ayarları yapılmıştır.

WDT ayarlarında ise, watchdog timer için clock kaynağı ACLK seçilmiş, Mode olarak “Interval Timer” mod seçilmiştir. Default olarak WDTISx bitleri sıfır olduğu için WDT clock kaynağı 32768 e bölünmektedir. Clock kaynağı ACLK seçili olduğuna ve ACLK clock olarak default durumda 32.768 kHz lik harici kristal seçili olduğu için WDT birimi 32768/32768= 1 sn içerisinde resetlenmez ise interval timer mod da, wdt kesme vektörüne dallanacaktır.

WDT ayarlarına devam edecek olursak, WDT interrupt ı enable edilmiş ve ardından kesme bayrağı temizlenmiştir. Ardından Clock ayarları yapılmıştır. Daha sonra ise port ayarları kısmında LED in bağlı olduğu pin temizlenmiş ve çıkış olarak şartlanmıştır. Son olarakta genel kesmelere izin verilmiş ve sonsuz döngüye girilmiştir. Sonsuz döngü içerisinde yapılacak işlem yoktur. Bu durumda herhangi bir kesme gelmesi durumunda mikrodenetleyici kesme alt programına dallanacak, oradaki kodları icra edecek ve sonrarasında ise boş boş beklemeye devam edecektir.

Bu nedenle mikrodenetleyiciyi hiçbir iş yapmadığı durumlarda güç tasarruf modlarından birine geçilip, harcanan güç minimuma indirilebilir.

Kesme alt programında icra edilen kodları inceleyecek olursak, ilk olarak LED yakılmış. Ardından belirli bir süre gecikme verilip, tekrar söndürülmüştür. Yine aynı gecikme süresince beklenilip, led tekrar yakılmış yine aynı gecikme süresi kadar beklenilip, led tekrar söndürülmüştür. Son olarak wdt kesme bayrağı temizlenmiş ve kesme alt programından çıkılmıştır.

Bu uygulamada Watchdog Timer birimi INTERVAL MODE da çalıştırılıp, Wdt sayıcısı bilerek temizlenmeyip kesmeye gidip gitmeyeceği kesme içerisinde P1.0 pinine bağlı olan led e flash yaptırılarak anlaşılmıştır.

WatchDog Timer birimi genel olarak şu şekilde kullanılır. Projelere başlarken ayak altında dolaşması istenmeyen birşeymiş gibi kapatılır. Proje sona erdikten sonra, WDT birimi açılır ve yazılım içerisinde periyodik olarak resetlenir. Bu şekilde üretilen cihaz sahada eğer yazılımsal ve donanımsal bir problemle karşılaşırda kilitlenirse, bu birim sayesinde cihaz resetlenerek bu durumdan kurtulmuş olunur.

Bu konu hakkındaki benim söyleyeceklerim bu kadar.  İnşallah sizlerede faydalı olabilmişimdir. Bir dahakidersimizde görüşmek üzere. Şimdilik Hoşçakalın…

***Ferudun GÖKCEGÖZ***

***fgokcegoz@yahoo.com***