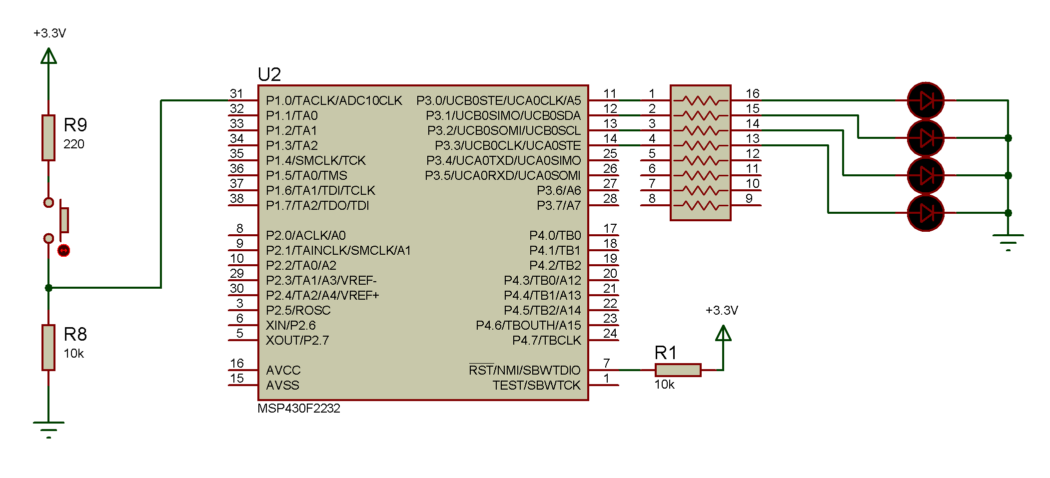
**MSP430 Uyg.5 – Buton ile BCD Sayıcı Uygulaması**

Ferudun GÖKCEGÖZ, 04 Ağustos 2011, Perşembe

Merhaba Arkadaşlar. Yeni bir uygulama ile yine birlikteyiz. Bu uygulamamızda bir adet buton kontrolüyle, bir bcd(binary coded decimal) sayıcı yapacağız. Sayıcının aldığı değerleri görmek için 4 adet led bağladım. Siz isterseniz, proteus kütüphanelerinde bulunan BCD 7 Segment Display koyarak, direk rakam olarak görebilirsiniz.

**İşte devre şemamız…**



Devre şemamızda görüldüğü gibi P1.0 butonuna bağlı bir adet butonumuz ve P3 portuna bağlı BCD sayıcımızın değerini görüntüleyeceğimiz 4 adet ledimiz bulunmaktadır. İsterseniz önce yazılımı verelim, ardındanda yazılımdan bahsetmeye başlayalım.

|  |  |
| --- | --- |
| **[Source code](http://www.mcu-turkey.com/msp430-uyg-5/" \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-5/#codesyntax_1)** **[http://www.mcu-turkey.com/wp-content/plugins/wp-synhighlight/themes/default/images/printer.png](http://www.mcu-turkey.com/msp430-uyg-5/#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 BUTTON P1IN\_bit.P0*

unsigned char num;

void main( void )

{

WDTCTL = WDTPW + WDTHOLD;

DCOCTL=CALDCO\_1MHZ;

BCSCTL1=CALBC1\_1MHZ;

P1DIR = 0x00;

P3OUT = 0x00;

P3DIR = 0x0F;

for(;;)

{

if(BUTTON)

{

num++;

P3OUT = num;

while(BUTTON);

if(num==10) num=0;

}

}

}

Yazılımda ilk olarak butonumuzun bağlı olduğu pini define etmişiz. Daha sonra, “num” adında bir değişken tanımlamışız. Bu değişkenimiz ram de bir byte lık yer kapalayacaktır. Birde şunu söylemeliyimki C de global olarak tanımlanan değişkenlerin ilk değeri sıfırdır. Değişkeniniz main fonksiyonunda veya diğer alt programlarınızda tanımlanmış ise ilk değeri sıfır değildir. Ram de rastgele bir değer almasıs söz konusudur. Bu nedenle global değişkenlerinizi tanımlarken sıfıra eşitlemenize gerek yoktur. Eğer global değilde local değişkeniniz varsa, ilk değer olarak sıfıra eşitlemeniz fayda sağlayacaktır.

Yazılımda daha sonra, WDT ve Clock ayarlamaları yapılmış ardından da, port ayarlarına geçilmiştir. Port ayarlarında da, P1 portunun tamamı giriş olarak şartlanmış, P3 portu temizlenmiş ve  ilk dört biti çıkış olarak şartlanmıştır. Daha önceki uygulamaları bizzat kendiniz yazıp denediyseniz, yavaş yavaş bu ayarlara aşina oluyorsunuz demektir.

Daha sonra sonsuz döngüye girilmiş ve BUTTON umuzun basılı olup olmadığı kontrol edilmektedir. Butona basılı ise global olarak tanımladığımız “num” değişkeni bir artırılmakta ve P3 portuna gönderilmektedir. Daha sonrada butondan elimizin çekilmesi beklenmektedir. Eğer butondan elimizi çekmemiş isek, mikrodenetleyici bu satırda sürekli bekleyecektir. Bu satır butona elimiz sürekli basılı kaldığında, sayının birden fazla artmasını önlemek için konmuştur.  Son olarakta ledlerle göstediğimiz sayının 10 a ulaşıp ulaşmadığı kontrol edilmiş, eğer 10 a ulaşılmışsa “num” değişkenimiz sıfırlanmıştır.

Yazımızın başında da söylediğimiz gibi, sayıcının değerini ledler ile görmek istemezseniz, proteus kütüphanelerinde bulunan bcd 7segment koyarak rakam olarak görebilirsiniz. Ayrıca uygulamayı sizlerde istediğiniz gibi çeşitlendirebilirsiniz. Örneğin belirli bir sayıdan başlatıp geriye saydırabilirsiniz. Yada butona basılı değilken sayıyı artıtıp butona basıldığında durmasını sağlayabilirsiniz.

Bu dersimizinde sonuna geldik. Takip eden arkadaşlara kolaylıklar dilerim.. Bir dahaki dersimizde görüşmek üzere. Şimdilik Hoşçakalın…

***Ferudun GÖKCEGÖZ***

***fgokcegoz@yahoo.com***