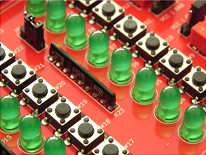
**1-Karaşimşek LED Uygulaması**

Muhammed Fatih İNANÇ, 05 Aralık 2012, Çarşamba



Bu uygulama notunda MSP430 Geliştirme Kiti üzerinde “Karaşimşek” uygulamasının nasıl yapılacağını inceleyeceğiz.

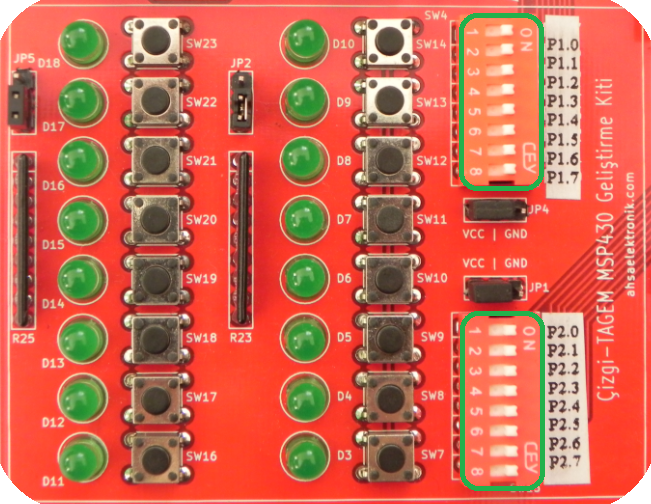
Bu uygulamada gerçekleştirilmesi amaçlanan işlem, kit üzerinde bulunan LED leri MSP430 mikrodenetleyicisi ile kontrol etmektir.

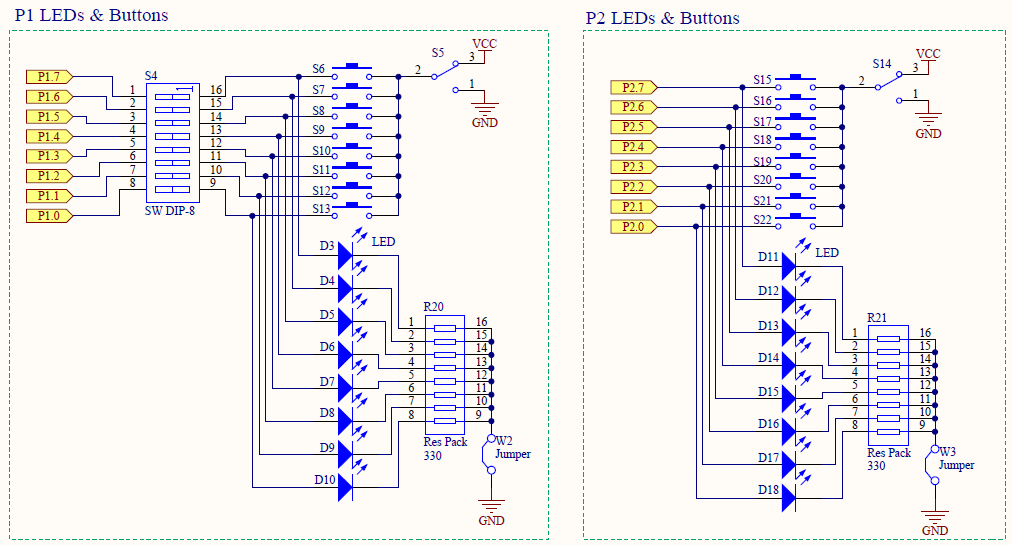
**Portların Ayarlanması**

MSP430 Geliştirme Kiti üzerindeki MSP430 mikrodenetleyicinin pinlerinin tamamını LED ler için kullanacağımız için kit üzerindeki tüm switchler kapalı konumda tutulmalıdır.

Ayarlanması gereken switchler;

* SW4 ve SW15 in tümü açık.
* JP1 ve JP4 **VCC** konumunda
* JP2 ve JP5 **VCC** konumunda
* LaunchPad üzerindeki J3 konnektöründen **TXD**ve **RXD** jumperlerı çıkarılmalıdır.

****

****

**Uygulamanın Çalışması**

“Karaşimşek” uygulaması bildiğiniz gibi arka arkaya dizilmiş LED lerin sıra ile yanması sonucu ortaya çıkan hareketli ve göz alıcı bir görüntü ortaya çıkarak yanıp sönmesidir.

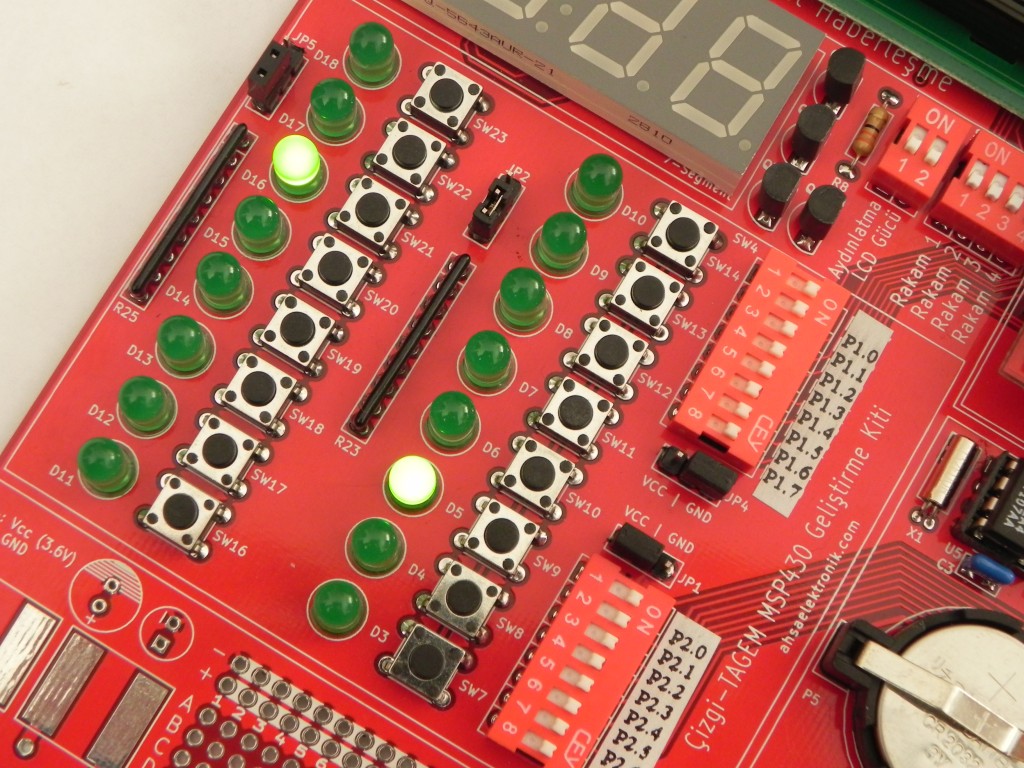
MSP430 Geliştirme Kitinde ise P1 ve P2 portlarında bulunan toplam 16 adet LED in karşılıklı yanması sonucu “Karaşimşek” uygulaması gerçekleştirilmiştir.

P1 portunda yukarıdan aşağıya, P2 portunda ise aşağıdan yukarıya olarak sıralı bir şekilde LED ler yakılmaktadır.

**Uygulama Kodu**

|  |  |
| --- | --- |
| main.c | |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37 | //  //LaunchPad Üzerindeki TXD ve RXD Jumplerlarını Çıkarınız    #include "msp430.h"    void delay\_ms(unsigned int ms)  {    while(ms--)      \_\_delay\_cycles(1000);  }    void main( void )  {    int a=1,b=128;      WDTCTL = WDTPW + WDTHOLD;      BCSCTL1 = CALBC1\_1MHZ;    DCOCTL  = CALDCO\_1MHZ;    \_\_delay\_cycles(1000000);      P1DIR = 0xFF;    P2DIR = 0xFF;    P2SEL = 0x00;    P2SEL2= 0x00;      while(1)    {      P1OUT = a;      a<<=1;      P2OUT = b;      b>>=1;        delay\_ms(500);        if(a > 128 && b < 1)      {        a=1;        b=128;      }    }  } |

* İlk olarak 14. satırda *tamsayı(int)*olarak **a**ve**b**adında iki değişken tanımlanıyor. Bu değişkenler en son LED hareketlerini gerçekleştirmek üzere port saklayıcılarına yüklenecek.
* 16 ve 17.satırlarda flash hafızada bulunan RC osilatör kalibrasyon değerleri okunarak CPU frekansı **1MHz**‘e ayarlanıyor. Ve arkasından(18.satır) kalibre edilmiş frekansın stabilitesi için 100 milisaniye bekleniyor.
* 22 ve 25.satırlar arasında ise Port tanımlamaları yapılıyor. Ve tüm pinler GPIO olarak seçiliyor.
* 27.satırdan itibaren bir sonsuz döngüye giriliyor. Sonsuz döngü içerisinde gerçekleştirilen işlem aslında “karaşimşek” hareketlerini gerçekleştiriyor.
* 29.satıda P1OUT sakayıcısına **a** değişkeni yüklenerek 1.LED in yanması sağlanıyor.
* 30.satırda ise **a** değişkeninin içeriği 1 bit sola kaydırılıyor.
* 31 ve 32.satırlarda ise **b** değişkeninin içeriği P2OUT saklayıcısına yüklenerek **b** değişkeni 1 bit sağa kaydırılıyor.
* 36 ve 40.satırlar arasında ise bit kaydırma işleminin taşması kontrol edilerek taşma durumu gerçekleştirildiğinde ilk değerler tekrar değişkenlere yükleniyor.

****