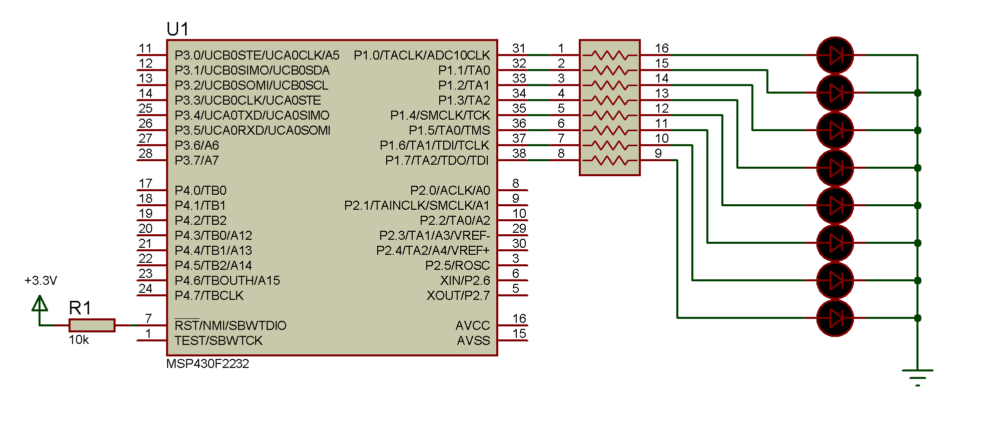
**MSP430 Uyg.3 – Sıralı Led Uygulaması**

Ferudun GÖKCEGÖZ, 04 Ağustos 2011, Perşembe

Merhaba arkadaşlar. 3 dersimizle yine sizlerle birlikteyiz. Bu dersimizde ise P1 portuna bağlı 8 adet ledimizi sıra ile yakıp söndüreceğiz. İsterseniz LaunchPad lerinizde de bu uygulamayı deneyebilirsiniz. P1.0 dan P1.7 ye  kadar olan tüm pinlere 220 ohm gibi bir direnç üzerinden Led bağlarsanız rahatlıkla deneme imkanınız olacaktır.

**Devre şemasını verelim ilk olarak…**



Devremizde P1 portuna bağlı 8adet led bulunmaktadır. Yazılımımızda LED lerimizi sıra ile tümünü yakıp ardındanda sıral ile tümünü söndüreceğiz.  Bu uygulamayı sizde anladıktan sonra istediğiniz gibi çeşitlendirebilirsiniz. Yazılımımıza geçersek;

|  |  |
| --- | --- |
| **[Source code](http://www.mcu-turkey.com/msp430-uyg-3/" \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-3/#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 P1\_0 P1OUT\_bit.P0*

*#define P1\_1 P1OUT\_bit.P1*

*#define P1\_2 P1OUT\_bit.P2*

*#define P1\_3 P1OUT\_bit.P3*

*#define P1\_4 P1OUT\_bit.P4*

*#define P1\_5 P1OUT\_bit.P5*

*#define P1\_6 P1OUT\_bit.P6*

*#define P1\_7 P1OUT\_bit.P7*

void delay (void)

{

unsigned int j;

for(j=50000;j;j--);

}

void main( void )

{

WDTCTL = WDTPW + WDTHOLD;

DCOCTL=CALDCO\_1MHZ;

BCSCTL1=CALBC1\_1MHZ;

P1OUT = 0x00;

P1DIR = 0xFF;

for(;;)

{

P1\_0 = ~P1\_0;

delay();

P1\_1 = ~P1\_1;

delay();

P1\_2 = ~P1\_2;

delay();

P1\_3 = ~P1\_3;

delay();

P1\_4 = ~P1\_4;

delay();

P1\_5 = ~P1\_5;

delay();

P1\_6 = ~P1\_6;

delay();

P1\_7 = ~P1\_7;

delay();

}

}

Yazılımı incelediğinizde anlaşılması gayet sade olduğunu sizde göreceksiniz. İlk olarak ledlerimizin bağlı olduğu P1 portu temizlenerek, çıkış olarak şartlandırılmıştır. Ardından da sonsuz söngüye girilerek, P1 portuna bağlı tüm pinler sıra ile terslenmektedir.

Satır aralarında ise beklemeleri sağlamak adına “delay();” fonksiyonu çağrılmıştır. Bu fonksiyonumuz main fonksiyonunun üst kısmındadır. Bu nedenle ayrıca global kısımda, fonksiyon prototip ine ihtiyaç yoktur. Ama eğer siz fonksiyonunuzu main fonksiyonunun altında yazmak isterseniz, global kısımda yani main fonksiyondan daha önce, fonksiyon prototipini yazmanız gerekecektir. Bunun nedeni, derleyici programı derlerken, en üst kısımdan başlar ve aşağı doğru iner. Main fonksiyonuna geldiğinde, main fonksiyonunda çağırılan bir programla hiç karşılaşmamışsa onu tanıyamayacaktır ve hata verecektir. Eğer fonksiyon main in altında ise, main den önce de prototipi varsa, main e geldiğinde o fonksiyon çağrıldığında o fonksiyonu tanıyıp hata ile karşılaşılmayacaktır.

Bu arada P1 portuna ait tüm pinler main den önce define edilmiştir. Eğer böyle olmasaydı her satırda her pin için uzun uzun pin ismini yazmak gerekecekti. #define işlemi bu sayede bize kolaylık sağlamıştır.

|  |  |
| --- | --- |
| **[Source code](http://www.mcu-turkey.com/msp430-uyg-3/" \l "codesyntax_2" \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-3/#codesyntax_2)** **[http://www.mcu-turkey.com/wp-content/plugins/wp-synhighlight/themes/default/images/printer.png](http://www.mcu-turkey.com/msp430-uyg-3/#codesyntax_2)** **[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)** |

*#define P1\_0 P1OUT\_bit.P0*

*#define P1\_1 P1OUT\_bit.P1*

*#define P1\_2 P1OUT\_bit.P2*

*#define P1\_3 P1OUT\_bit.P3*

*#define P1\_4 P1OUT\_bit.P4*

*#define P1\_5 P1OUT\_bit.P5*

*#define P1\_6 P1OUT\_bit.P6*

*#define P1\_7 P1OUT\_bit.P7*

Geldik bu yazımızında sonuna. İsteyen arkadaşlar aynı devre üzerinden çeşitli uygulamalarda yapabilirler. Örneğin P1 portuna 0×01 veya 0×80 gibi değerler yükleyip sonsuz döngünün içinde her seferinde bir kaydırarak led lerin sıra ile yandığını görebilirsiniz.

Yada bir önceki dersimizdeki buton uygulaması ile birleştirip buton kontrolünde, ledlerin sıra ile yanmasını deneyebilirsiniz. İlerleyen derslerimizde de benzer uygulamalar yapacağız zaten..

Bir sonraki yazımızda görüşmek üzere. Şimdilik Hoşçakalın..

***Ferudun GÖKCEGÖZ***

***fgokcegoz@yahoo.com***