

20 MART 2017

BBM418 GÖMÜLÜ SİSTEMLER LABORATUVARI

LAB4 LABORATUVAR RAPORU

ENVER KESMEN
MEHMET AKİF ÖZDEMİR

1.AMAÇ

Mikroişlemci yardımıyla breadboard üzerinde kurulan düzenekteki LED'leri belirli bir süre ve belirli bir sırayla yakmayı (mavi-0.5sn, kırmızı 1sn, yeşil 1.5sn) ve breadboard üzerine yerleştirilen butona her basıldığında kesmen kullanarak bu sıralamayı tersine döndürüp kaldığı yerden devam ettirmeyi amaçladık.

2.TEORİK ALTYAPI

İlk olarak LED'lerin bağlanacağı pinlere karar verildi (PA3,PA4,PA5) ve çıkış pini, buton(PA3) ise giriş pini olarak ayarlandı.

2.1 Kullanılacak global değişkenler

Değişken türü	Değişken adı	Açıklama
unsigned long	Counts	LED'in kaç timer kesmesi boyunca yandığını sayıyor.
unsigned long	Switch	LEDlerin yanma yönünü belirliyor(1 veya 0)
unsigned long	TempPortA_Data	portA' daki datayı geçici olarak saklamak için kullanılıyor
unsigned long	WhichLed	Hangi ledin yanacağını belirliyor

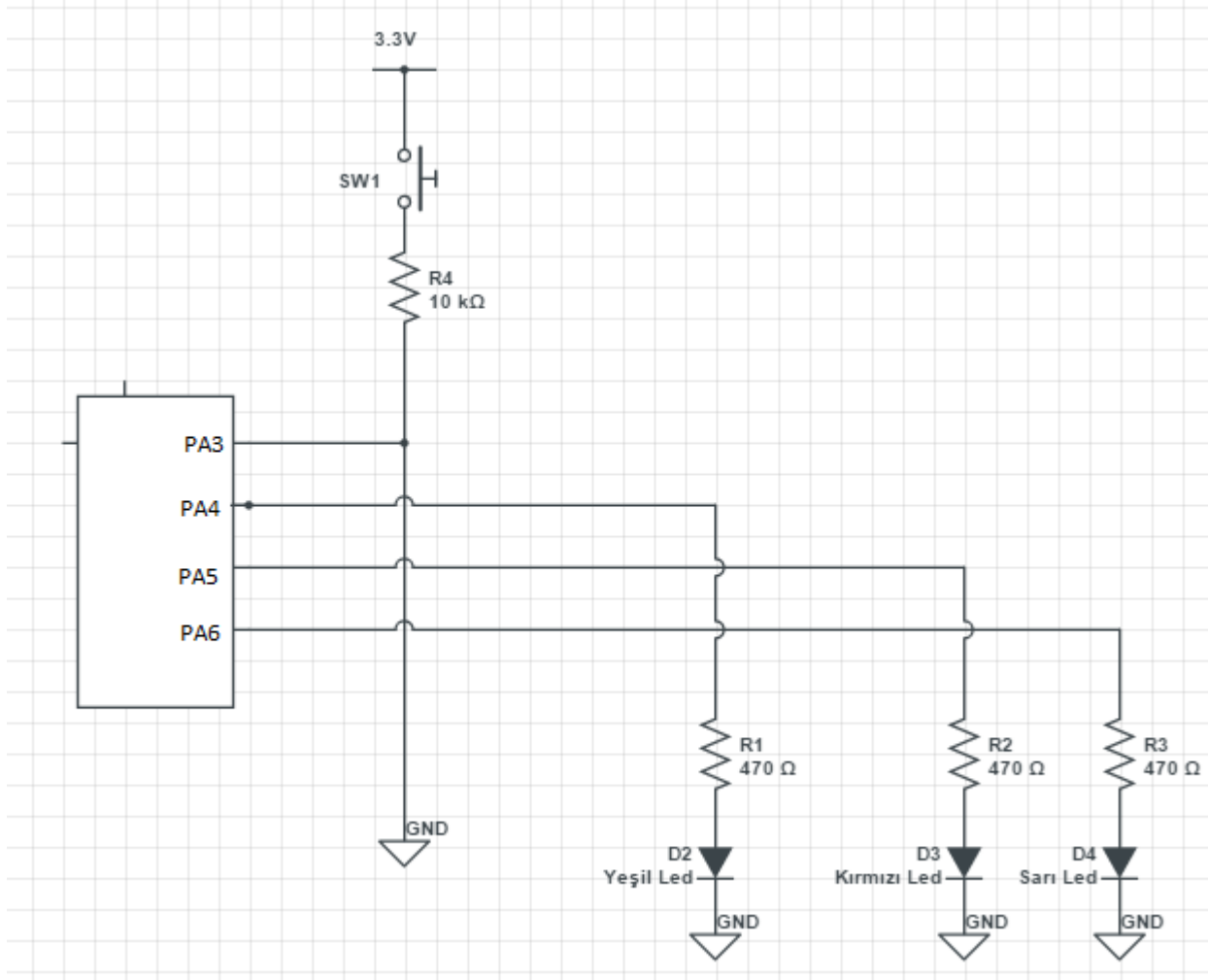
2.2 Kullanılacak Fonksiyonlar

Fonk. Adı	Aldığı değerler	Fonk. Türü	Açıklama
SysTick_Init	-	void	SysTick timer ayarlamalarını yapıyor
EnableInterrupts	-	void	Kesmeleri aktif hale getiriyor
ChangeLed	unsigned long x	void	Kendisine gelen değere göre yeni LED'i değiştiriyor.
PortA_Init	-	void	A portunun kurulumunu gerçekleştiriyor.
GPIOPortA_Handler	-	void	A portunda ayarlanan kesmeleri işliyor
SysTick_Handler	-	void	Systick timerda ayarlanan kesmeleri işliyor

3.GERÇEKLEŞTİRİM VE SONUÇLAR

3.1 Gerçekleştirim

3.1.1 Sistemin tasarlanması



3.1.2 LED'lerin yanma sürelerinin hesaplanması

Bu süreleri hesaplamak için SysTick zamanlayıcısının kesmelerini kullandık. Kesmenin her 0,5 saniyede bir olması için zamanlayıcıyı 8000000 değeri ile kurduk. Daha sonra Counts değişkeni ile kaç kesme gerçekleştiğini saydık. Örn: Yeşil LED'in 1.5 saniye yanması için Counts 2'ye set ediliyor. Her timer kesmesinde bir azaltıldığı için 3. Kesmede yani 1.5 saniye sonra 0 değerine ulaşılıyor ve 0 olduğu kontrol edildikten sonra timer kesmesi içerisinde sıradaki LED yakılıyor.

```

//SysTick timer interruptu her 0.5 saniyede bir kez aktif oluyor
void SysTick_Handler(void){

    if(Counts == 0)//o anki ledde yanmasi gereken sure tamamlandi ise isliyor
    {
        if(Switch==0x00) //Switch 0 ise WhichLed degiskenini arttiriyor degilse azaltiyor
            WhichLed++;
        else
            WhichLed--;

        ChangeLed(WhichLed); //Ledi degistir.
    }
    else //sure tamamlanmadi ise bir azaltarak 0.5 saniye daha bekliyor
        Counts--;
}

```

3.1.2 LED yanarken butona sadece ilk basmanın algılanması

Bu sorunu çözmek için TempPortA_Data değişkeninden faydalandık. Buton kesmesi geldiği zaman o anki PortA'daki veri ile TempPortA_Data'da tutulan veriyi kıyaslıyoruz eğer farklı ise butona o renkte ilk kez basıldığını anlıyoruz ve Switch değişkeninin tersini aldıktan sonra TempPortA_Data=PortA komutunu çalıştırıyoruz. Tekrar aynı LED'de butona basılır ise TempPortA_Data=PortA olacağı için herhangi bir değişiklik olmuyor. Burada bir sorunla daha karşılaştık Mavi LED yanarken butona basıldığını kabul edelim TempPortA_Data da mavi LED yanarken oluşan veriler olacak. LED'ler bir tur döndükten sonra tekrar mavi LED'de butona basılırsa TempPortA_Data=PortA olacağı için sanki ikinci basma gibi algılayıp herhangi bir işlem yapmıyor. Bu sorunu çözmek için ise her yanan LED'i değiştirdiğimizde TempPortA_Data = 0xFF şeklinde bir atama yaptık.

```

void GPIOPortA_Handler(void){

    GPIO_PORTA_ICR_R = 0x08; //PA3 clear flag

    if(TempPortA_Data != GPIO_PORTA_DATA_R)
    {
        Switch = !Switch;
        TempPortA_Data = GPIO_PORTA_DATA_R;
    }
}

```

3.1.3 LED'lerin yanma sıralamasının hangi yönde olacağına karar verilmesi

Bu problemin çözümü için Switch değişkenini kullandık. Buton kesmesi gerçekleştiğinde eğer o an yanan LED süresince butona ilk kez basılmış ise Switch değişkeninin tersini aldık. Daha sonra timer kesmesi içerisinde LED değişikliğine karar verilmiş ise (Counts==0) Switch değişkeninin değerine göre WhichLed değişkenini arttırdık yada azalttık.

```
//SysTick timer interruptu her 0.5 saniyede bir kez aktif oluyor
void SysTick_Handler(void) {

    if(Counts == 0) //o anki ledde yanması gereken süre tamamlandı ise isliyoruz
    {
        if(Switch==0x00) //Switch 0 ise WhichLed degiskenini arttiriyor degilse azaltiyor
            WhichLed++;
        else
            WhichLed--;

        ChangeLed(WhichLed); //Ledi degistir.
    }
    else //sure tamamlanmadi ise bir azaltarak 0.5 saniye daha bekliyor
        Counts--;
}
```

3.1.4 Hangi LED'in yanacağına karar verilmesi

LED'leri 0(PA4),1(PA5),2(PA6) şeklinde numaralandırdık. ChangeLed fonksiyonu kendisine gelen LED numarasını yakmasını sağladık.

```
//Gelen degiskene göre hangi ledin yanacagina karar veriyor
void ChangeLed(unsigned long x)
{
    x%=0x3;
    switch(x)
    {

        case 0:
            GPIO_PORTA_DATA_R = 0x10; //PA4'e bagli ledi yakiyor
            Counts = 2; //Toplamda 3 timer kesmesi suresi boyunca yanik kalmasi icin 2'ye set ediliyor
            break;

        case 1:
            GPIO_PORTA_DATA_R = 0x20; //PA5'e bagli ledi yakiyor
            Counts=1; //Toplamda 2 timer kesmesi suresi boyunca yanik kalmasi icin 1'ye set ediliyor
            break;

        case 2:
            GPIO_PORTA_DATA_R = 0x40; //PA6'ya bagli ledi yakiyor
            Counts=0; //bir timer kesmesi boyunca yanik kalmasi icin 0'a set ediliyor.
            break;
    }
    TempPortA_Data = 0xFF; //Led degistigi icin hafizda tuttugumuz porta verisini resetliyoruz
}
```

3.2 Sonular

3.2.1 Deęerlendirme

LAB3 ve LAB2 ile kıyaslırsak en saęlıklı alıřan yntemin bu olduęu kanaatindeyiz. Kodlarken nceki LAB derslerine gre daha fazla zorlandık kesmelerin ayarlanması yapılandırılması bizi ciddi derecede yıprattı. Kodlama bittikten sonra olayı ok daha iyi kavradık bu haftadan sonraki LAB derslerinde kesme kodlanması gerekirse ok fazla zorlanacaęımızı dřnmyoruz.

3.2.2 Deneyin YouTube linki

<https://youtu.be/MGQCxqZgaQU>