

İ.T.Ü.
Bilgisayar ve Bilişim Fakültesi
Bilgisayar Mühendisliği Bölümü



MİKROBİLGİSAYAR LABORATUVARI DENEY RAPORU

Deney No : 5
Deney Adı : Code Warrior Yazılımının Tanıtılması
Deney Tarihi : 24.11.2011 Perşembe
Grup : 10

Deneyi Yapanlar : 040080153 Serkan Güler
040080322 Osman Boyacı
040090533 Abdullah Aydeğer

Deneyi Yaptıran Araştırma Görevlisi: Hasan Kıvrak

Deneyin Amacı

- Code Warrior yazılımının tanıtımı
 - ✓ ASM projelerinin oluşturulması
 - ✓ Gerekli kodun eklenmesi ve uygun platform için derlenmesi
 - ✓ Oluşturulan kodun cihaz üzerine yüklenmesi ve çalıştırılması

Code Warrior Kullanımı

▪ Assembly projesinin oluşturulması

Code Warrior programı çalıştırılıp ‘HC(S)12 New Project Wizard’ seçilip yeni bir proje oluşturulmuştur. Daha sonra listeden kart üzerinde bulunan mikrodenetimci olan MC9S12C32 seçilmiştir.

Proje oluşturulduktan sonra ninovada verilen ‘main.asm’ dosyası editör vasıtası ile projeye aktarılmıştır. F7 ile kod derlenmiş, herhangi bir hatayla karşılaşılmamıştır. Daha sonra F5 tuşuna basarak oluşturulan kod karta yüklenmiştir. Yükleme ile birlikte ‘real-time debugger’ programı çalışmaya başlamış olup, programın üzerindeki yeşil ok düğmesine basılarak karta yüklenen programın çalışması sağlanmıştır.

▪ Main.asm

```
; export symbols
      XDEF Entry ; export 'Entry' symbol
      ABSENTRY Entry ; for assembly entry point

; include derivative specific macros
      INCLUDE 'mc9s12c32.inc'

; variable/data section
      ORG $3800 ; RAMStart defined as $3800
; in the file 'mc9s12c32.inc'
Counter1  DC.W $ffff ; X register count number
Counter2  DC.W $0010 ; Y register count number
StackSP   DS.W $0006 ; memory space for stack

; code section
StackST ; $3810 here, 16 bytes from $3800
Entry
```

```

        LDS #StackST ; initialize the stack pointer
        LDAA #$01 ; LED 1 at PORTA bit 0
        STAA $0002 ; set PORTA bit 0 as output
        LDAA #$10 ; LED 2 at PORTB bit 4
        STAA $0003 ; set PORTB bit 4 as output

mainLoop
        LDAA #$10 ; Turn off LED 2 at PORTB4
        STAA PORTB
        LDAA #$00 ; Turn on LED 1 at PORTA0
        STAA PORTA
        JSR delay1sec
        LDAA #$00 ; Turn on LED 2 at PORTB4
        STAA PORTB
        LDAA #$01 ; Turn off LED 2 at PORTA0
        STAA PORTA
        JSR delay1sec

mainLoop2
        LDAA PORTE ; read SW1 at PORTE0
        CMPA #$1
        BEQ mainLoop
        BRA mainLoop2

; subroutine section
delay1sec
        LDY Counter2 ; long delay by

dly1Loop
        JSR delayMS ; Y * delayMS
        DEY
        BNE dly1Loop
        RTS

delayMS
        LDX Counter1 ; short delay
dlyMSLoop NOP ; X * NOP
        DEX
        BNE dlyMSLoop
        RTS

```

▪ **Assmbly Programı Açıklama**

Main.asm dosyasındaki programda öncelikle Counter1 ve Counter2 değişkenleri tanımlanıyor. Ayrıca PORTA'nın 0.biti ve PORTB'nin 4.biti verici olarak kullanılıyor. 'mainLoop' ile birlikte çalışmaya başlayan program 1 saniye aralıkla LED'lerin birini yakıp birini söndürüyor. 'mainLoop2' de SW1 tuşuna basıp basılmadığı kontrol ediliyor. Eğer SW1 tuşuna basıldıysa LED'ler tuşa basmayı bırakana kadar en son halini koruyor.