MİKROBİLGİSAYAR LABORATUVARI DENEY RAPORU

Deney No : 4
Deney Adı : Gerçek Zamanlı Kesme Uygulamaları
Deney Tarihi : 15.12.2011 Perşembe
Grup : 10

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

Deneyi Yaptıran Araştırma Görevlisi : Gökhan Seçinti

Bölüm 1
Gerçek zamanlı kesme kullanarak CSM12C32 kartı üzerindeki LED'i periyodik yakıp söndürmek

deney4_kisim1.asm kodunda yorum satırları ile açıklananlar adım adım koda dökülmüştür. Deney esnasında bizden istenildiği gibi RTICTL yazmacı 8.192 milisaniyede bir kesme üretecek şekilde koşullanmıştır. Kesme hizmet programında ise bir sayıcı ile 2 saniyenin dolup dolmadığına bakılmış ve dolmuşsa LED in konumu değiştirilmiştir. Bu sayı ise 2sn / 8.192ms = 244 tur. Yaptıklarımızın aşağıdaki kodda daha iyi anlaşılacağını düşünerek satir satır yorumladık.

```
;*****************************************************************
;* This stationery serves as the framework for a                 *
;* user application (single file, absolute assembly application) *
;* For a more comprehensive program that                         *
;* demonstrates the more advanced functionality of this          *
;* processor, please see the demonstration applications          *
;* located in the examples subdirectory of the                   *
;* Freescale CodeWarrior for the HC12 Program directory          *
;*****************************************************************

; export symbols
            XDEF Entry            ; export 'Entry' symbol
            ABSENTRY Entry        ; for absolute assembly: mark this as application entry point

; include derivative specific macros
            INCLUDE 'mc9s12c32.inc'

ROMStart    EQU  $4000  ; absolute address to place my code/constant data

; variable/data section
 ifdef _HCS12_SERIALMON
            ORG $3FFF - (RAMEnd - RAMStart)
 else
            ORG RAMStart
 endif
 ; Insert here your data definition.
Counter     DS.W 1

; code section
            ORG   ROMStart
Entry:
            ; remap the RAM &amp;amp; EEPROM here. See EB386.pdf
 ifdef _HCS12_SERIALMON
            ; set registers at $0000
            CLR   $11                 ; INITRG= $0
            ; set ram to end at $3FFF
            LDAB  #$39
            STAB  $10                 ; INITRM= $39

            ; set eeprom to end at $0FFF
            LDAA  #$9
            STAA  $12                 ; INITEE= $9

            LDS   #$3FFF+1        ; See EB386.pdf, initialize the stack pointer
 else
            LDS   #RAMEnd+1       ; initialize the stack pointer
 endif
            ;CLI                    ; enable interrupts

mainLoop:
            ; PortA'nin tüm bitlerini çikis olacak sekilde kosullandirin
            LDAA #$FF   ;butun bitlere 1 verilerek
                        ;PortaA'nin tum bitleri verici olarak kosullandi
            STAA $0002 ;PortA'nin yonlendiricisi
            ;PortA'ya uygun sayiyi yazarak LED1'i söndürün
            STAA PORTA  ;FF degeri yazilarak LED1 sonduruldu.

            ; Counter adresindeki sayiyi bir yapin
            LDAA #$01
            STAA Counter

            JSR rti_init
            CLI

loop:       BRA loop

rti_init:   ; RTICTL yazmacini 8.192 milisaniyede bir kesme uretecek sekilde
            ; kosullandirin. Osilator hizi 16 MHz.
            LDAA #$71 ;artik 2*2^16 saniyede bir kesme gelsin..
            ;8.192 milisaniyede bir kesme uretecek sekilde
            STAA RTICTL
            ; CRGINT yazmacindan gercek zamanli kesmeyi etkinlestirin
            LDAA #$80 ;7. bite 1 yazilarak gercek zamanli kesme etkinlestirildi.
            STAA CRGINT
            RTS

rti_isr:    ; Counter adresindeki sayiyi bir artirin
            LDAA Counter
            INCA
            STAA Counter
            ; Bu sayiyi 2 saniye gecmesi icin gerekli sayiya ulasip
            ; ulasmadigini kontrol edin
            CMPA #$F4 ;2sn/8.192milis=224
            ; Ulastiysa LED1'in durumunu tumleyin (yaniyorsa sondurun, sonukse yakin)
            BEQ TUMLE
            BRA TUMLEME
TUMLE       LDAA PORTA
            COMA
            STAA PORTA
            ; CRGFLG yazmacinin 7nci bitine 1 yazarak bu biti sifirlayin
TUMLEME     BSET CRGFLG,#$80
            RTI
               ;!!!Interrupt Vectors!!!;
            ORG   $FFFE
            DC.W  Entry           ; Reset Vector

            ORG   $FFF0
            DC.W  rti_isr
```

Bölüm 2
Belirli bir sayıdan ileriye sayan zamanlayıcı yapılması

Bu deneyde ise 0' dan başlayarak belirlediğimiz sayıya kadar 2''şer 2'şer sayan ve istenilen sayıya ulaşınca terminalde belli bir mesajı gösteren program yazılmıştır. Belirlediğimiz sayı 16'dir. Bir önceki deneydekinin aksine her 2 saniye geçtiğinde LED'in söndürülmesi yerine o anki saniye değeri Hyperterminal bağlantısı ile ekrana sunulmuştur. Baud rate 9600 bps olacak şekilde önceki deneydeki gibi ayarlanmış, ayrıca *sendtext* altprogramı aynen kullanılmıştır. Kodlar ve her satirin işlevi yanlarında yorumlanmıştır.

```
;*****************************************************************
;* This stationery serves as the framework for a                 *
;* user application (single file, absolute assembly application) *
;* For a more comprehensive program that                         *
;* demonstrates the more advanced functionality of this          *
;* processor, please see the demonstration applications          *
;* located in the examples subdirectory of the                   *
;* Freescale CodeWarrior for the HC12 Program directory          *
;*****************************************************************

; export symbols
            XDEF Entry            ; export 'Entry' symbol
            ABSENTRY Entry        ; for absolute assembly: mark this as application entry point

; include derivative specific macros
            INCLUDE 'mc9s12c32.inc'

ROMStart    EQU  $4000  ; absolute address to place my code/constant data

; variable/data section
 ifdef _HCS12_SERIALMON
            ORG $3FFF - (RAMEnd - RAMStart)
 else
            ORG RAMStart
 endif
 ; Insert here your data definition.
Counter     DS.W 1
Number      DS.B 1
NumStr      DS.B 4
Message     DC.B " saniye kaldi",10,13,0
Message2    DC.B "Bingo!!!!",0

; code section
            ORG   ROMStart
Entry:
            ; remap the RAM &amp;amp; EEPROM here. See EB386.pdf
 ifdef _HCS12_SERIALMON
            ; set registers at $0000
            CLR   $11                 ; INITRG= $0
            ; set ram to end at $3FFF
            LDAB  #$39
            STAB  $10                 ; INITRM= $39

            ; set eeprom to end at $0FFF
            LDAA  #$9
            STAA  $12                 ; INITEE= $9

            LDS   #$3FFF+1        ; See EB386.pdf, initialize the stack pointer
 else
            LDS   #RAMEnd+1       ; initialize the stack pointer
 endif
            ;CLI                    ; enable interrupts

mainLoop:   ;PortA'nin tüm bitlerini çikis olacak sekilde kosullandirin
            LDAA #$FF ; //butun bitlere 1 verilerek PortaA'nin tum bitleri verici olarak kosullandi
            STAA $0002 ;PortA'nin yonlendiricisi
            ;PortA'ya uygun sayiyi yazarak LED1'i söndürün
            STAA PORTA  ;FF degeri yazilarak LED1 sonduruldu.
            ; Counter adresindeki sayiyi bir yapin
            LDAA #$01
            STAA Counter

            LDAA #0  ; 0. saniyeden 16 ya kadar ileri sayalim
            STAA Number
            JSR init_sci
            JSR rti_init
            CLI

loop:       BRA loop  ;ana program sonsuz dongude kesme bekliyor
num2str:    LDX #NumStr   ;index registeri stringin basini gostersin.
            LDAA #$30
            STAA 00,X    ;ilk sifir yazildi
            LDAA Number
            CMPA #$0A
            BLT ADANKUCUK  ;A'DAN kucukse dallan, buyukse sayiyi onar
;A'DANBUYUK
;ornegin NumStr de E yazsin
            SUBA #$0A   ;E-A=4 ==>birler basamagi iki
            LDAB #1     ;onlar basamagi ise 1
            ORAA #$30   ;ASCII kodlari bulundu
            ORAB #$30   ;ASCII kodlari bulundu
            STAB 01,X   ;onlar basamagi yazildi(bu ornekte 1)
            STAA 02,X   ;birler basamagi yazildi(bu ornekte 4)
            LDAA #$00
            STAA 03,X   ;terminating karakter NULL yazildi
            RTS         ;alt programdan don;

;ornegin NumStr de 8 yazsin
ADANKUCUK   LDAB #0     ;onlar basamgi sifir,birler basamagi ise direk ayni sayi
            ORAA #$30   ;ASCII kodlari bulundu
            ORAB #$30   ;ASCII kodlari bulundu
            STAB 01,X   ;onlar basamagi yazildi(bu ornekte 1)
            STAA 02,X   ;birler basamagi yazildi(bu ornekte 4)
            LDAA #$00
            STAA 03,X   ;terminating karakter NULL yazildi
            RTS         ;alt programdan don;
```

```
rti_init:   ; RTICTL yazmacini 8.192 milisaniyede bir kesme uretecek sekilde
            ; kosullandirin. Osilator hizi 16 MHz.
            LDAA #$71 ;2*2^16 saniyede bir kesme gelsin..
            STAA RTICTL  ;2^16

            ; CRGINT yazmacindan gercek zamanli kesmeyi etkinlestirin
            LDAA #$80 ;7. bite 1 yazilarak gercek zamanli kesme etkinlestirildi.
            STAA CRGINT
            RTS

rti_isr:    ; Counter adresindeki sayiyi bir artirin
            LDAA Counter
            INCA
            STAA Counter
            ; Bu sayiyi 2 saniye gecmesi icin gerekli sayiya ulasip
            ; ulasmadigini kontrol edin
            CMPA #$F4 ;(2sn/8.192milis) = 244
            ; Ulastiysa Number adresindeki sayinin degerini bir artirin
            BEQ YAZDIR
            BRA YENIKESME

YAZDIR      LDY #NumStr
            JSR num2str
            LDX #NumStr
            JSR sendtext   ; kalan saniyeyi ekrana yazdir
            LDX #Message
            JSR sendtext   ; "saniye kaldi mesajini ekrana yazdir"
            LDAA Number
            CMPA #$10      ;16 ya ulastikmi?
            BEQ BITIR   ;ulastiysak programi bitir, yeni interrupt da gelmesin.
            INCA        ;ulasmamisiz, artmaya devam et.
            INCA
            STAA Number

            ; CRGFLG yazmacinin 7nci bitine 1 yazarak bu biti sifirlayin
YENIKESME   BSET CRGFLG,#$80    ;boylelikle 8.192milisaniye sonra yenikesme gelebilir.
            RTI                 ;kesmeden don ve main de yeni kesme bekle

BITIR       LDX #Message2
            JSR sendtext   ; "BINGOOO!!!"
            ;artık kesme gelmeyecek cunku CRGFLG yazmacinin 7nci bitini sifirlanmadi.
            RTI             ;kesmeden don ve yeni kesme bekleme.

initsci:    ;eski deneydeki kodlar aynen kullanildi
            LDAA #52 ; Sci hız ayarı 9600 baud
            STAA SCIBDL
            LDAA #$00 ; Sci 8 veri biti, paritesiz, 1 dur biti
            STAA SCICR1
            LDAA #%00001100 ; Sci vericisi ve alıcısı aktif
            STAA SCICR2
            RTS

sendtext:   ;eski deneydeki kodlar aynen kullanildi
geri:       LDAA SCISR1 ; Sci vericinin yeni veri için hazır olup olmadığı
            ANDA #$80 ; kontrol ediliyor.
            BEQ geri
            LDAB 00,X ; X'in gösterdiği adresten başlayarak
            STAB SCIDRL ; NULL olan karaktere kadar
            INX ; karakterleri hafızadan okuyup
            CMPB #$00 ; vericiye gönderiliyor.
            BNE geri
RTS
;***************************************************************
;*                 Interrupt Vectors                           *
;***************************************************************
            ORG   $FFFE
            DC.W  Entry           ; Reset Vector
            ORG   $FFF0
            DC.W  rti_isr
```