

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



MİKROBİLGİSAYAR
LABORATUVARI
DENEY RAPORU

Deney No : 7
Deney Adı : Seri İletişim Uygulamaları
Deney Tarihi : 08.12.2011 Perşembe
Grup : 10

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

Deneyi Yaptıran Araştırma Görevlisi: Hasan Kıvrak

Deneyin Amacı

- Seri İletişim Uygulamaları
 - ✓ Seri iletişimin anlaşılması
 - ✓ Seri iletişimin ayarlarının ve yazmaçlarının anlaşılması
 - ✓ Program beklemeli seri iletişim kavramının anlaşılması
 - ✓ Kesme tabanlı seri iletişim kavramının anlaşılması

Bölüm 1

▪ **Seri iletişim arayüzü kullanarak karakter dizinin gönderilmesi**

Deneyin bu bölümünde, mikrodeneleyici üzerindeki seri iletişim arayüzü kullanılarak, bilgisayar ile iletişim sağlanmıştır. 'init_sci' alt programında bağlantı ayarları 9600 bps (baud) olacak şekilde ayarlanmış, bağlantının 8 veri biti, paritesiz ve 1 dur biti olması için sci kontrol yazmacına uygun değer girilmiştir. 'sendtext' alt programında ise X yazmacında gönderilen adresten başlayarak karakterleri hafızadan okuyup, sci durum yazmacından vericinin gönderime hazır olduğunu kontrol ederek NULL karakteri gelene kadar göndermesi sağlanmıştır.

Bu bölümde gerçekleştirilen kod aşağıdaki gibidir;

```
*****
;* 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
```

```

ifndef _HCS12_SERIALMON
    ORG $3FFF - (RAMEnd - RAMStart)
else
    ORG RAMStart
endif
; Insert here your data definition.
Message DC.B "Ben sana mecburum bilemezsin, Adını mih gibi aklımda tutuyorum",0

; code section
    ORG ROMStart
Entry:
    ; remap the RAM & EEPROM here. See EB386.pdf

ifndef _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:
    JSR initsci
    LDX #Message
    JSR sendtext
11:
    BRA 11

```

initsci:

```
LDAA #52                ; Sci hız ayarı 9600 baud
STAA SCIBDL
LDAA #$00               ; Sci 8 veri biti, paritesiz, 1 dur biti
STAA SCICR1
LDAA #%00001000        ; Sci vericisi aktif
STAA SCICR2
RTS
```

sendtext:

```
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
```

➤ Sci Baud Rate Register (#52) değerinin hesaplanması

SCI baud rate = SCI module clock / (16 * BR)
 $9600 = 8.10^6 / 16 * BR$
BR = #52

Bölüm 2

- **Bilgisayarın seri iletişim arayüzünden gönderdiği karakterlerin program-beklemeli olarak okunup belleğe yazılması**

Deneyin bu bölümünde, seri iletişim arayüzünün vericisi ile birlikte alıcısı da kullanılmıştır. Hyperterminal programına girilen karakter dizisi ENTER tuşuna basılıncaya veya hafızaya ayrılan yer doluncaya kadar okunup hafızaya yazılmıştır.

Bu bölümde gerçekleştirilen kod aşağıdaki gibidir;

```
*****  
;* 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.  
Buffer DS.B 16  
Message DC.B "Karakter dizisini giriniz: ",0  
  
; code section  
ORG ROMStart
```

Entry:

; remap the RAM & 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:

JSR initsci

LDX #Message

JSR sendtext

LDX #Buffer

JSR waittext

LDX #Buffer

JSR sendtext

11:

BRA 11

initsci:

```
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:

```
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
```

waittext:

```
LDAA #$00
STAA PORTA
LDAB #$00
don   LDAA SCISR1  ; Durum yazmacından gelen verinin
      ANDA #$20    ; hazır olup olmadığı kontrol ediliyor.
      BEQ don
      LDAA SCIDRL  ; Hazırsa veri okunuyor.
      CMPB #15     ; Alınan karakter sayısı bellek_boyu -1'den
      BGE Last_Char ; küçük ise belleğe yazılıyor.
      CMPA #$0D     ; Okunan karakter ENTER tuşu ise veya
      BEQ Last_Char ; alınan karakter sayısı bellek_boyu -1 ise
      STAA 00,X     ; son karakter olarak NULL yazılıp programa
      INX           ; geri dönülüyor.
      INCB
      BNE don
```

Last_Char

```
LDAA #$FF
STAA PORTA
STAA 00,X
RTS
```

```
*****
;
;*          Interrupt Vectors          *
;*****
;
ORG $FFFE
DC.W Entry ; Reset Vector
```

Kodları CSM12C32 kartına yüklenerek, Hyperterminal programında bilgisayardan gönderdiğimiz “Karakter dizisini giriniz:” mesajı gözlenmiştir. Daha sonra Hyperterminal programında girilen verilerin ENTER tuşuna basılıncaya veya hafıza ayrılan yer doluncaya kadar okunduğu ve hafızaya yazıldığı görülmüştür.