

Mérési utasítás

Digitális rendszerek - Labor

3.sz. mérés

Szükséges ismeretanyag:

- IBM PC kezelése, szövegszerkesztés, Double Commander
- A SAB80C515 mikrokontroller felépítése, utasításai
- HyperTerminál (a PC-n futó kommunikációs program)
- A mikrokontroller oktatórendszer monitorprogramja

Az előre elkészített mérési jegyzőkönyv nyomtatványt a mérés során ki kell tölteni és a mérés végén a mérésvezetőnek le kell adni!

Ha a mérési utasítást nem nyomtatott formában használják, akkor olvashatják tabletről, vagy notebookról, de a mérésre használt asztali gépről NEM!

Mérési feladatok:

1. Indítsák el a számítógépet. A Windows7 bejelentkezésekor válasszák a **Hallgató** nevű felhasználót. Keressék meg az asztalon és indítsák el a **Virtuális gép**-et. A virtuális gép bejelentkezése után kattintsanak kettőt a **Virtuális gép - DSP és mikrokontroller méréshez** sorra. Miután a Windows XP elindult, tegyék teljes képernyőssé. Indítsák el a **HyperTerminált** a **Micro1** ikonra kattintással. Csatlakoztassák a mikrokontroller oktatórendszert az erősáramú hálózathoz. Ellenőrizzék, hogy a mikrokontroller monitorprogramja bejelentkezett-e a képernyőn. Nyomják meg az oktatórendszer piros **RESET** gombját és figyeljék a képernyőt. A mérés során használhatják a jegyzetet: **C:\IARC\Mikrokontroller.pdf**

A mikrokontroller külső adatmemóriája egy akkumulátor segítségével akkor is megőrzi a tartalmát, ha nem kap tápfeszültséget. Ezenkívül az előjegyzett töréspontokat sem felejtí el. Ezért annak érdekében, hogy az előző mérőcsoportok által végzett memóriatartalom módosításokat eltüntessük, gépeljék be az alábbi két parancsot:

```
FILLX 0 7fff 00 <Enter>
BK ALL <Enter>
```

Ellenőrizzék a **Segédlet**-ben, hogy mi a fenti parancsok hatása. Állítsák a mikrokontroller kezelőegységének valamennyi kapcsolóját bekapcsolt állapotba (minden LED világít).

A továbbiakban egy-egy monitorparancs kiadása előtt ellenőrizzék annak használatát a **Segédlet**-ben, amennyiben szükséges.

2. Futófény

Szövegszerkesztővel (pl. Double Commander Shift+F4 billentyű) hozzák létre **futof.s03** névvel az alábbi programot a **WORK** alkönyvtárban (a ;-vel kezdődő comment-eket ne gépeljék be). A programot az alább látható módon, zónákra osztva írják meg. Az egyes zónák elválasztására használják a **tabulátort**.

```
LSTOUT-
$C:\IARC\INC\SFR80515.INC
LSTOUT+
      ORG      100H
      MOV      P4,#0FFH
      MOV      A,#1
KEZD:  MOV      P1,A
      RL       A
      MOV      R7,P4
KESL1: MOV      R6,#0
KESL2: NOP
      DJNZ     R6,KESL2
      DJNZ     R7,KESL1
      AJMP    KEZD
      END
```

Mentsék el, fordítsák le a programot, töltsék be a mikrovezérlőbe és futtassák. Változtassák a sebességét a P4 kapcsolóival. Írják át a programot úgy, hogy a P5 kapcsolói vezéreljék a sebességét. Rögzítsék a jegyzőkönyvben a változás(oka)t.

3. Oda-vissza futófény

A Double Commander segítségével másolják át a **WORK** könyvtárba az asztalon található Digit/Peldak-Meres2 mappa tartalmát (Elérése: **C:\Documents and Settings\User\Asztal\Digit\Peldak-Meres2**). Az **F3** billentyűvel jelenítsék meg és ellenőrizzék a **futofov.s03** program listáját:

```
LSTOUT-
$C:\IARC\INC\SFR80515.INC
LSTOUT+
      ORG          100H
      MOV          P4,#0FFH
      MOV          A,#1
BAL:   MOV          P1,A
      RL           A
      JB           P1.7,JOBB
      ACALL        KESL
      SJMP        BAL
JOBB:  RR          A
      MOV          P1,A
      JB           P1.0,BAL
      ACALL        KESL
      SJMP        JOBB
KESL:  MOV          R7,P4
KESL1: MOV          R6,#0
KESL2: NOP
      DJNZ        R6,KESL2
      DJNZ        R7,KESL1
      RET
      END
```

Fordítsák le a programot, töltsék be a mikrovezérlőbe és futtassák. Változtassák a sebességét a P4 kapcsolóival. Írják át a programot úgy, hogy a futófény a P5-ön fusson és a P1 kapcsolói vezéreljék a sebességét. Rögzítsék a jegyzőkönyvben a változás(oka)t.

4. Oda-vissza futófény + megszakítás kiszolgálás

Az **F3** billentyűvel jelenítsék meg és ellenőrizzék a **futofint.s03** program listáját:

```
; A foprogram azonos az oda-vissza futofennyel
; A Timer0 megszakitas kiszolgalasa 100 msec periodusideju 1:1 kitoltesu
; negyszogjelet allit elo a P3.5-on.
```

```
LSTOUT-
$C:\IARC\INC\SFR80515.INC
LSTOUT+
SZAML EQU          50000

      ORG          0BH                      ;Timer0 megszakitas belepesi cime
      LJMP        TKIS                      ;Kiszolgalo rutinra ugras

      ORG          400H
      MOV          TMOD,#00000001B         ;Timer0 uzemmodbeallitas
      MOV          TL0,#LOW(65536-SZAML)    ;16 bites szamlalo kezdoertek also bajt
      MOV          TH0,#HIGH(65536-SZAML)  ;16 bites szamlalo kezdoertek felso bajt
      SETB        TR0                      ;Timer0 indul
      SETB        ET0                      ;Timer0 megszakitas engedelyezese
      SETB        EAL                      ;Globalis megszakitas engedelyezes
```

```

;Az eredeti oda-vissza futofeny program
MOV      P4,#0FFH
MOV      A,#1
BAL:    MOV      P1,A
        RL
        JB      P1.7,JOBB
        ACALL  KESL
        SJMP
JOB:    RR
        MOV      P1,A
        JB      P1.0,BAL
        ACALL  KESL
        SJMP
KESL:   MOV      R7,P4
KESL1:  MOV      R6,#0
KESL2:  NOP
        DJNZ
        DJNZ
        RET

;Tenyleges megszakitas kiszolgalas
TKIS:   CLR      TR0 ;Timer0 leáll
        CPL      P3.5 ;Invertalja a P3.5-ot
        MOV      TL0,#LOW(65536-SZAML) ;16 bites szamlalo kezdoertek also bajt
        MOV      TH0,#HIGH(65536-SZAML) ;16 bites szamlalo kezdoertek felso baj
        SETB     TR0 ;Timer0 ujraindul
        RETI     ;Visszateres a megszakitasbol

        END

```

Ellenőrizték a jegyzet segítségével, hogyan programozzuk fel a Timer0-át, és mi történik a megszakítás kiszolgálás során. Fordítsák le a programot, töltsék be a mikrovezérlőbe és futtassák. Figyeljenek oda, hogy ennek **nem 100H a kezdőcíme!!** Ráadásul két ORG direktíva is szerepel a programban, vajon melyik tartalmazza a valódi kezdőcímet? Változtassák a sebességét a P4 kapcsolóival. Mérjék meg oszcilloszkóppal a négyszögjel periódusidejét. Írják át a programot úgy, hogy a Timer0 megszakítása 50 msec periódusidejű négyszögjelet állítson elő a P3.5-ön. Rögzítsék a jegyzőkönyvben a változtatás(oka)t. Ellenőrizték az eredményt oszcilloszkóppal. Az oszcilloszkóp kezeléséhez kérjenek segítséget a mérésvezetőtől.

5. Egy 8 bites adat (ASCII-kód) vétele a soros vonalról

Az F3 billentyűvel jelenítsék meg és ellenőrizték a **sorvesz.s03** program listáját:

```

; A program a soros vonalon vett adatot kiirja a P1 portra
LSTOUT-
$C:\IARC\INC\SFR80515.INC
LSTOUT+
        ORG      100H
CIKL:   CLR      RI
SVAR:   JNB      RI,SVAR ;Megvarja, amig a soros veteli buffer megtelik
        MOV      P1,SBUF ;A soros veteli buffer tartalmat a P1-re masolja
        SJMP     CIKL    ;Uj ciklus
        END

```

A soros portot nem kell felprogramoznunk, mert ezt a monitorprogram már megtette. A P1 porton azonnal megjelenik az adat, amint egy billentyűt lenyomunk a klaviatúrán. A kiírt adat az adott billentyűhöz tartozó karakter ASCII-kódja. Fordítsák le és futtassák a programot. Ellenőrizték, hogy mi az ASCII kódja a kis a és nagy A karaktereknek, mennyi köztük a különbség. Rögzítsék a jegyzőkönyvbe. Rögzítsék további 4 tetszőleges karakter ASCII-kódját.

6. Egy 8 bites adat kiírása a terminálképernyőre binárisan

Az F3 billentyűvel jelenítsék meg és ellenőrizzék a **p1binse.s03** program listáját:

```
; A program a P1 porton beállított értéket kiírja a terminalra binárisan
; a P3.2 kapcsolóval vezérelve

LSTOUT-
$C:\IARC\INC\SFR80515.INC
LSTOUT+

      ORG      100H
CR     EQU     13          ;Kocsi vissza (Carriage Return) ASCII kódja a
                          ;CR névhez rendelve
LF     EQU     10          ;Soremelés (LineFeed) ASCII kódja az LF névhez rendelve
VEZ    EQU     P3.2       ;P3.2 bitcíme a VEZ névhez rendelve

      SETB     VEZ        ;P3.2 bemenet
      MOV      A,#CR      ;Kocsi vissza kódjának
      ACALL    KIIR       ;küldése a soros vonalra
      MOV      A,#LF      ;Soremeles kódjának
      ACALL    KIIR       ;küldése a soros vonalra
VAR1:  JB      VEZ,VAR1
VAR2:  JNB     VEZ,VAR2
      MOV      A,P1
      MOV      R7,#8      ;Az ACC legfelső bitjét ciklikusan a Carry-be
CIKL:  RLC     A           ;forgatjuk (8-szor)
      PUSH    ACC         ;ACC a verembe
      JNC     NULL
      MOV     A,#'1'      ;Az A-ba az 1 ASCII kódja, ha C=1
      SJMP   SOROS
NULL:  MOV     A,#'0'      ;különben a 0 ASCII kódja.
SOROS: ACALL   KIIR
      POP     ACC
      DJNZ   R7,CIKL
      MOV     A,#'B'      ;A B betű ASCII kódjának
      ACALL  KIIR         ;küldése a soros vonalra
      MOV     A,CR        ;Kocsi vissza kódjának
      ACALL  KIIR         ;küldése a soros vonalra
      SJMP   VAR1        ;Uj ciklus

KIIR:  CLR     TI
      MOV     SBUF,A      ;Az A tartalma a soros adási bufferbe
SVAR:  JNB    TI,SVAR     ;Megvárja a soros adás befejezést
      RET
      END
```

A soros portot most sem kell felprogramoznunk, mert ezt a monitorprogram már megtette. A program a P1-en beállított érték minden egyes bitjét (0 vagy 1) az adott szám 8 bites ASCII kódjával helyettesíti és ez küldi a soros vonalra, vagyis a terminálképernyőre. A vezérlés a P3.2 kapcsoló le- majd felkapcsolásával történik. Az új érték mindig a képernyő ugyanazon sorának első 8 karakterhelyére kerül. Fordítsák le és futtassák a programot. Láthatják, hogy hibásan működik, a kocsi-vissza vezérlést nem hajtja végre. Keressék meg és javítsák ki a program hibáját (Kocsi vissza kód küldése a soros vonalra kétszer is szerepel a programban!). Rögzítsék a jegyzőkönyvben. A most már helyesen működő programmal írassák ki a P1 négy különböző értékét és rögzítsék a jegyzőkönyvben.

7. Saját program írása

Írjanak programot, ami a P1 alsó négy bitjét a P4 alsó négy bitjére, felső négy bitjét pedig a P5 alsó négy bitjére írja ki végtelen ciklusban. Rögzítsék a jegyzőkönyvben a programlistát.

Algoritmus:

- P1 legyen bemenet (minden bitje 1, ezt csak egyszer kell végrehajtani a programnak)
- P1-et másoljuk az A-ba (itt kezdődik a ciklus)
- A-ról készítsünk biztonsági másolatot (pl. a B-be)
- Maszkoljuk az A felső négy bitjét úgy, hogy azok mind 0-ák legyenek, míg az alsó négy bit maradjon változatlan (ANL utasítás)
- Másoljuk az A-t a P4-be
- Hozzuk vissza az A-ba az eredeti értékét
- Maszkoljuk az A alsó négy bitjét úgy, hogy azok mind 0-ák legyenek, míg az felső négy bit maradjon változatlan (ANL utasítás)
- Cseréljük meg az A alsó és felső 4-4 bitjét (SWAP utasítás)
- Másoljuk az A-t a P5-be
- Kezdjük előlről a ciklust

8. Jutalom: játékprogram

Az F3 billentyűvel jelenítsék meg és ellenőrizzék a **jatek.s03** program listáját. Próbálják meg kitalálni a programlista alapján, hogyan működik. (Ha nem sikerül kitalálni, lapozzanak az ellenőrző kérdésekhez.) Fordítsák le és futtassák a programot. A játékhoz jó szórakozást!

9. Töröljék a WORK könyvtár tartalmát és ürítsék ki a lomtárat is! Ha szükségesnek látják, törlés előtt a munkájuk eredményét (a Work könyvtár tartalmát) adathordozóra (pen-drive) másolhatják. Állítsák le a virtuális gépet. Leállításakor válasszák a Power Off lehetőséget. Állítsák le a **Windows-t. Húzzák ki a mikrovezérlő dugóját a konnektorból!** Adják le a mérési jegyzőkönyvet.

Ellenőrző kérdések:

- Hogyan kell használni a játékprogramot?

Az első játékos beállít egy kitalálendő számot a P1-en és a P3.2 le- majd felkapcsolásával beviszi. Ezután a P1 minden kapcsolóját felfelé kapcsolja. A második játékos is beállít egy számot (találgatás) és beviszi. Ha az ő száma kisebb, mint a kitalálendő, akkor a P4 alsó négy LED-je világít, ha nagyobb, akkor a felső négy és jöhet a következő találgatás. Ha végre pontosan eltalálta, akkor a P5 valamennyi LED-je 10-szer villog és új játék kezdődik.