Mérési utasítás
Mikrokontroller programozás
2.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 Windowx XP elindult, tegyék teljes képernyőssé. Indítsák el a Double Commandert és lépjenek be a **C:** meghajtó **IARC** könyvtárának **WORK** alkönyvtárába. Indítsák el a **HyperTerminált** az asztalon a **Micro1** ikonra kattintással. Csatlakoztassák a mikrokontroller oktatórendszert az erősáramú hálózathoz. Ellenőrizzék, hogy a mikrokontroller monitorprogramja bejelentkezett-e a képernyőn.

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 felejti 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>

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.

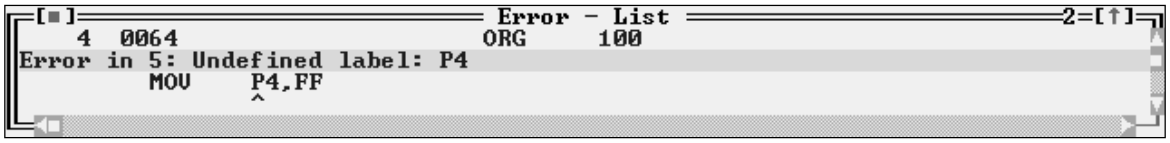
2. Programfejlesztés, hibakeresés
2a. Szövegszerkesztővel (pl. Double Commander Shift+F4 billentyű) hozzák létre **pelda1.s03** névvel az alábbi programot a **WORK** alkönyvtárban (a ;-vel kezdődő comment-eket ne gépeljék be). A programot az alább látható módon, zónákra osztva írják meg. Az egyes zónák elválasztására használják a **tabulátort.**

```
        ORG         100         ; Elhelyezésszámláló, a program kezdőcíme a memórában
        MOV         P4,FF       ; A P4 port bemenet (minden bitje 1)
KEZD:   INC         P1          ; P1 tartalmának növelése
KESL1:  MOV         R7,P4       ; Külső ciklusszámláló kezdőértéke a P4-en beállított
                                ; érték
        MOV         R6,#0       ; Belső ciklusszámláló kezdőértéke 0
KESL2:  NOP
        DJNZ        R6,KESL2    ; Csökkentés 1-gyel, ha nem nulla a KESL2 címre adja
                                ; át a vezérlést
        DJNZ        R7,KESL1    ; Csökkentés 1-gyel, ha nem nulla a KESL1 címre adja
                                ; át a vezérlést
        AJMP        KEZD
        END
```

(Szövegszerkesztésre használhatják a DOS EDIT parancsot ill. az oktatórendszer E paranccsal indítható editorát is. A szerkesztés befejezése után mentsék a programot).

A program (akárcsak az 1. mérés 6. pontja) bináris számlálót valósít meg végtelen ciklusban a P1 porton, de kiegészítettük egy késleltetéssel, amit két egymásba ágyazott ciklussal valósítunk meg. A késleltetés hosszát (a külső ciklusszámláló kezdőértékét) a P4-ről olvassuk be, vagyis változtathatjuk.

2b. Fordítsák le a programot az **ALL pelda1** parancs begépelésével a Double Commander parancssorába.
A fordítás során az alábbi hibaüzenetet kapják:

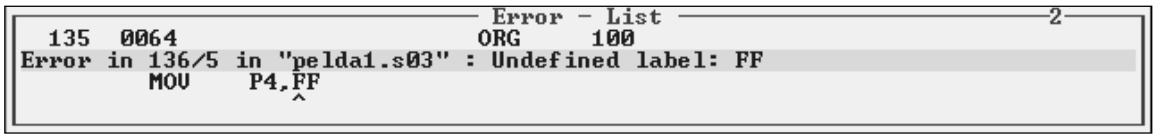


Ennek az az oka, hogy a fordítóprogram nem ”ismeri” a P4 címét. Gépeljék be a program első sora elé az alábbi három sort:

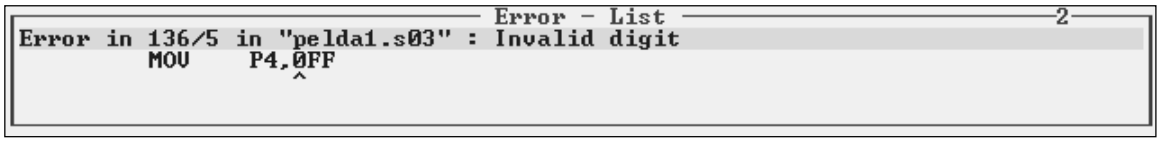
```
LSTOUT-
$C:\IARC\INC\SFR80515.INC
LSTOUT+
```

(Ha a mérés során olyan hibaüzenetet kapnak, ami a mérési utasításban nem szerepel, jelezzék a mérésvezetőnek, mert valószínűleg ”elgépeltek” valamit.)

Kíséreljék meg ismét a fordítást. Újabb hibaüzenet:



A betűvel kezdődő hexadecimális szám elé nem tettünk nullát! Javítsák ki és fordítsák újra!
Ismét hibaüzenetet kaptunk:



Nem jeleztük a számrendszer típusát. Gépeljenek a 0FF után egy H betűt (0FFH), és fordítsák újra!
A fordítás végre hibátlanul sikerült, több formai hiba nincs a programban. Ellenőrizzék és rögzítsék, hogy milyen fordítási termékek jöttek létre az alkönyvtárukban. Vegyék figyelembe, hogy az általunk begépelt fájl(ok), ill. a hibás fordítás során létrejött biztonsági másolatok **nem** fordítási termékek. Nézzék meg a Double Commander F3 parancsával a fordítás során létrejött **pelda1.lst** fájl tartalmát. Próbálják meg értelmezni.

2c. Készítsenek másolatot a **pelda1.hex** fájlról **pelda1.bak** néven. Töltsék be a mikrokontroller memóriájába a **pelda1.hex** programot a **HyperTerminál** Adatátvitel – Szövegfájl küldése menüpontjának használatával. A HyperTerminál Szövegfájl küldése ablakában a fájltípus sorban válasszák a Minden fájl (*.*) lehetőséget. Kattintsanak a **pelda1.hex** fájlra. Ellenőrizzék az **U** paranccsal (U100), hogy sikerült-e a letöltés és a memória 100-as címén látható-e a program? Próbálják meg kitalálni, hogy miért nem? (l. Ellenőrző kérdések.) Rögzítsék a jegyzőkönyvbe.

2d. Térjenek vissza a szövegszerkesztéshez. Javítsák ki a forrásprogram hibáját: írják be az elhelyezésszámláló direktívába (ORG 100) a számtípust jelző H betűt (ORG 100H). Fordítsák le ismét. Hasonlítsák össze a **pelda1.bak** és a **pelda1.hex** fájlok tartalmát. Rögzítsék a jegyzőkönyvbe, hogy mi a különbség és miért?

2e. Töltsék be ismét a mikrovezérlőbe a **pelda1.hex** programot. Ellenőrizzék az **U** paranccsal (U100), hogy most sikerült-e a letöltés. Futtassák a programot (G 100). Ha nem működik, hibakeresésre van szükség.

2f. A RESET gombbal állítsák le a programot. Jegyezzenek elő töréspontot az AJMP 100 utasításra. Ehhez meg kell nézni a **pelda1.lst** fájlban, hogy az adott utasítás milyen memóriacímre került (010E):

```
BS 010E
```

Ellenőrizzék a töréspontokat a BL paranccsal. Futtassák a programot (G 100). A program futásának a törésponti címen meg kell szakadni, bejelentkezik a monitorprogram és a regiszterek, ill. a memória tartalma a monitorparancsokkal vizsgálható. Ha a monitorprogram nem jelentkezett be, akkor a hiba a programunk korábbi részeiben van, végtelen ciklusba kerültünk.

Nyomják meg a RESET gombot. Töröljék a töréspontot (pl. **BK ALL**, vagy **BK 0**) és jegyezzenek elő egy új töréspontot a **MOV R6,#0** utasítás címére. Ehhez most azt kell megnézni a **pelda1.lst** fájlban, hogy ez az utasítás milyen memóriacímre került (0107):
Állítsanak be a P4 porton a kapcsolókkal 03H értéket. Futtassák a programot, majd amikor a monitorprogram bejelentkezik, vizsgálják meg, hogy a beállított érték bekerült-e az R7 regiszterbe (X parancs). Ha igen, akkor a hiba a két utasítás közötti részben van. Próbálják megkeresni a hiba okát és kijavítani. Rögzítsék a jegyzőkönyvbe a változtatás(oka)t. (Ha nem sikerül megtalálni, lapozzanak az ellenőrző kérdésekhez.)
2g. Töröljék a töréspontot (**BK ALL**), fordítsák le újra a programot, töltsék be és futtassák. Látszólag hibátlanul működik, pedig a program első utasításában még mindig maradt egy hiba, de a körülmények szerencsés összejátszása miatt nincs látható hatása. Keressék meg a hibát, javítsák ki és rögzítsék a jegyzőkönyvben!

3. Futófény

A Double Commander F4 billentyűjével nyissák meg szerkesztésre a **pelda1.s03** programot és azonnal mentsék el a **Fájl** menü **Mentés másként** parancsával **futof.s03** névvel. Szerkesszék át a programlistát az alábbira:

```
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.

4. 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.

5. 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          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

;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
```

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. Ellenőrizzék az eredményt oszcilloszkóppal.

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

Az F3 billentyűvel jelenítsék meg és ellenőrizzék 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. Ellenőrizzék, 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.

7. 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 beallitott erteket kiirja a terminalra binarisan
; a P3.2 kapcsoloval vezerelve

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 bitcime a VEZ névhez rendelve

       SETB        VEZ         ;P3.2 bemenet
       MOV         A,#CR       ;Kocsi vissza kodjanak
       ACALL       KIIR        ;kuldese a soros vonalra
       MOV         A,#LF       ;Soremeles kodjanak
       ACALL       KIIR        ;kuldese a soros vonalra
VAR1:  JB          VEZ,VAR1
VAR2:  JNB         VEZ,VAR2
       MOV         A,P1
       MOV         R7,#8       ;Az ACC legfelso bitjet 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 kodja, ha C=1
       SJMP        SOROS
NULL:  MOV         A,#'0'      ;kulonben a 0 ASCII kodja.
SOROS: ACALL       KIIR
       POP         ACC
       DJNZ        R7,CIKL
       MOV         A,#'B'      ;A B betu ASCII kodjanak
       ACALL       KIIR        ;kuldese a soros vonalra
       MOV         A,CR        ;Kocsi vissza kodjanak
       ACALL       KIIR        ;kuldese a soros vonalra
       SJMP        VAR1        ;Uj ciklus

KIIR:  CLR         TI
       MOV         SBUF,A      ;Az A tartalma a soros adasi bufferbe
SVAR:  JNB         TI,SVAR     ;Megvarja a soros adas befejezeset
       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.

8. 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 elölről a ciklust

9. 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!

10. 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. Leállításkor válasszák a Power Off lehetőséget. Állítsák le a **Windows**-t. **Húzzák ki a mikrovezérlő dugóját a konnektorból!** Adják le a mérési jegyzőkönyvet.

<u>Ellenőrző kérdések:</u>

A kérdések a válaszokat is tartalmazzák. A cél annak megértése és tisztázása, hogy miért pont ezek a válaszok.

- Milyen kiterjesztéssel kell megadni a forrásprogram nevét a fordításkor a 2b. pontban?
Fordításkor a kiterjesztés kiírása **TILOS!** A program lefordítása az **ALL pelda1** paranccsal lehetséges.

- Mi történik a korábbi fordítási termékekkel ismételt fordításkor?
A korábbi fordítási termékek felülíródnak. Ha bármelyikre később is szükségünk lehet, biztonsági másolatot kell róla készíteni.

- Miért nem sikerülhet a 2c. pontban a letöltött programot megtalálni a memóriában a 100 címen?
A forrásprogramban használt számok esetén meg kell adni a számtípust is:

100	decimálisan megadott szám	értéke	64H	ill.	100 decimálisan
100H	hexadecimálisan megadott szám	értéke	100H	ill.	256 decimálisan
101100B	binárisan megadott szám	értéke	2CH	ill.	44 decimálisan

A lefordított program tehát a 64H címre töltődött le, ezért az U100 paranccsal nem tudjuk listázni.

- Hol követtünk el logikai hibát a pelda1.s03 programban?
A **`KESL1:`** címkét nem a megfelelő utasítás elé írtuk. Töröljék a címkét a **`MOV R7,P4`** sor elől és írják be a **`MOV R6,#0`** sor elé.

- Hogyan kell használni a játékprogramot?
Az első játékos beállít egy kitalálandó 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 kitalálandó, 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.