

# ***A mikrokontrolleres oktatási gyakorló készülék***

## ***monitor és terminál programja***

**Összeállították: Zalotay Péter főiskolai docens  
Kopják József egy. hallgató**

## Általános leírás

A program a Budapest Műszaki Főiskola, Kandó Kálmán Villamosmérnöki Főiskolai kárán, az Automatika Intézet Elektronika Szakcsoport tanárai által kifejlesztett, Philips 80C552 mikrokontrollerre alapuló fejlesztő és gyakorló rendszerre készült

A monitorprogram, az Intel 8031/51 család tagjaihoz tartozó mikrokontrollerekre fejlesztett program a

- valós idejű tesztelését,
- a mikrogép memóriatartalmának megtekintését, módosítását és
- perifériáinak ellenőrzését

teszti lehetővé.

A program a mikrokontroller aszinkron soros portját használja a *terminál* programot futtató rendszerrel (PC-el) történő kommunikációs kapcsolat fenntartására. Programot a felhasználó *karakterekhez rendelt parancsokkal* vezérelheti.

Az 1. táblázat a gyakorló készülék memória kiosztását szemlélteti. A táblázat baloldali oszlopa tartalmazza a címzési területeket, míg a további két oszlop mutatja, hogy adott címtartományban CODE és/vagy DATA memória található. A táblázat egyes celláiban az olvasható, hogy a mikrokontroller olvasás (RD), írás (WR) vagy az utasítás olvasás (PSEN) vezérlőjelekkel éri el az adott memóriaterületet. A CODE memória oszlopban található piros sáv mutatja, hogy hol került tárolásra fizikailag a monitorprogram.

	CODE	DATA
FFFFh	-	I/O MEMORY RD, WR
C000h		
BFFFh	PSEN, RD	-
8000h		
7FFFh	-	PSEN, RD, WR
4000h		
3FFFh	PSEN	RD, WR
0000h		

1 táblázat

A 0000h - 3fffh címtartományban (első 16 Kbájt) mind EPROM, mind pedig RAM elérhető, de a mikrokontroller eredeti memóriakezelésével. A program, és az adat fizikailag külön egységben lehet.

A 4000h - 7fffh címtartományban (második 16 Kbájt) csak RAM van, amelynek elérése úgy került kialakításra, hogy itt a Neumann-i architektúra érvényesül, mely szerint egyazon fizikai memóriaterületen program, és adat szegmens is lehet.

A **8000h - 0Bfffh** címtartományban (harmadik 16 Kbájt) csak EPROM van, viszont az RD olvasójel bevezetésével a MOVX utasítással is lehet adatot olvasni ( pl. táblázatok).

A **0C000h – 0ffffh** címtartomány (negyedik 16 Kbájt) a memóriába ágyazott I/O illesztők részér fentartott.

A monitorprogram a **CODE** memória **0000h-3FFFh** és **8000h-8FFFh** terjedő címtartományában - EPROM-ba beégetve – helyezkedik el. A monitorprogram programfutás közben három memória részt használ:

- **R3** regiszter bankot,
- XDATA **3E00h-3FFFh** terjedő memóriarészét,
- STACK memóriát

A monitorprogram a lépésenként történő felhasználói program futatásához a mikrokontroller soros portjának (USART) megszakítását (23h) használja, így ezt a megszakítást a felhasználói program nem veheti igénybe. A többi megszakítást a monitor program LJMP utasítás segítségével átirányítja a CODE memória 4000h címe fölé.

### ***A monitorprogram fő részei:***

- Inicializálás.
- Memóriakezelés.
- Visszafordítás, (Unassambly).
- Felhasználói program letöltés és futtatás.
- Lépésenként történő felhasználói programfuttatás.
- Perifériakezelés.

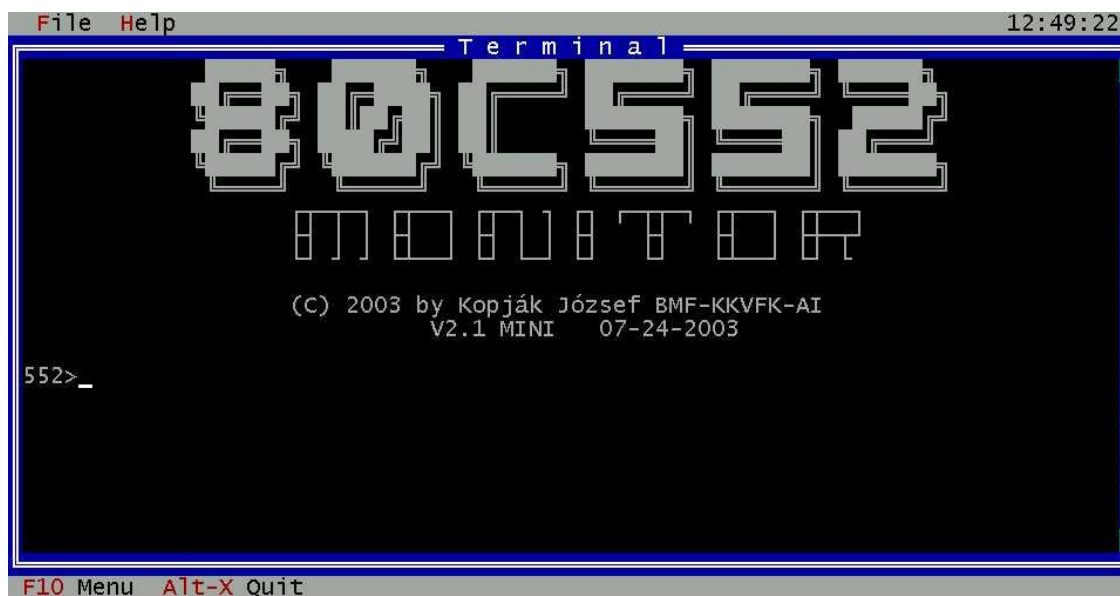
### ***Inicializálás***

Az inicializálás a mikrokontroller **reset**-eléskor, vagy a program memória 0000h címére történő ugrás után történik. Az inicializáló rész:

- törli a monitor program által használt XDATA memóriát (3E00h-3FFFh),
- törli a megszakítás kezelő regisztereket ( IE0,IE1,IP0,IP1), és
- alaphelyzetbe állítja a memóriába ágyazott perifériákat.

A gyakorlón található LED-sor tesztelése egy futófény segítségével történik. A hétszegmensű kijelzők szegmenseit sorra kiivilágítja, majd a középső g szegmense világít.

Végül inicializálja a soros portot, mely sebességét 9600 Baud-ra állítja. A terminál ablakba egy bejelentkező szöveg segítségével jelzi a felhasználónak, hogy készen áll a **parancsok** fogadására – 1.ábra. A képernyő tetején egy üdvözlő szöveg olvasható, míg lejjebb úgynevezett, prompt jelenik meg, a következő szöveggel: „ **552>** ”.



1.ábra

### ***A monitor-. és a terminál programok***

A PC-ben futó **terminál**-program, és a mikrogépben futó **monitor**-program együttműködése biztosítja azt, hogy a PC klaviatúrájáról vezérelni lehet a mikrogépben futó programot, illetve innen használható a képernyő ASCII karakteres állományok (szövegek, kvázi-grafikus stb. ) megjelenítésére. A két gép közötti kommunikáció az RS232 szabvány szerinti soros vonalon bonyolódik.

A **terminálprogram** – az indítását követően – felváltva figyeli a **klaviatúrát**, illetve a **soros vonalat**. A klaviatúrán leütött gomb scan-kódját a soros vonalra továbbítja. A sor vonalról érkező kódokat pedig a **képernyőn** jeleníti meg.

Az inicializálást követően a monitorprogram csak a soros vonalat figyeli. Az innen érkező egy karakteres **felhasználói parancs** ( 2. táblázat) észlelése után hajtja végre a parancshoz tartozó műveletsort, és küldi vissza válaszokat ugyancsak a soros vonalon.

#### ***Felhasználói parancsok:***

<b><i>Billentyű</i></b>	<b><i>Angol megnevezés</i></b>	<b><i>Parancs leírás</i></b>
<b>G</b>	Go form address	Adott című történő felhasználói programfuttatás
<b>U</b>	Unassembler	Programkód visszafejtő
<b>D</b>	Display memory	Memóriaterület listáztatás
<b>M</b>	Modify memory	Memória tartalom megváltoztatása
<b>C</b>	Copy memory	Memóriaterület másolása
<b>F</b>	Fill memory	Memóriaterület kitöltése
<b>I</b>	Inputs monitoring	Bemenetek folyamatos figyelése
<b>H</b>	Help	Segítség, felhasználói parancsok rövid összefoglalója
<b>S</b>	Step form address	Adott című történő lépésenkénti felhasználói programfuttatás

2.táblázat

A **H** gomb (help) leütése után a képernyőre is kiíródnak a parancsok, illetve a mikrogép memóriakirosztása (2. Ábra).



2.ábra

## G – Go form address – Adott címtől történő felhasználói programfuttatás

„Go form address” menü segítségével felhasználói programot futtathatunk a program-memória általunk megadott kezdőcímről.

- A „**G**” billentyű lenyomása után a képernyőn a következő felirat olvasható: „552>GO FROM ADDRESS:”.
- Ezt követően be kell írunk azt a **memóriacímet** – négy karakteres hexadecimális szám formájában (post-, vagy prefix nélkül) - ahol a felhasználói programunk kezdő utasítása található.
- Hibás beírást a **BACKSPACE** gomb segítségével korrigálhatjuk.
- **ESC** gomb leütésével léphetünk vissza a főmenübe.
- Az **ENTER** gomb megnyomását követően az ugrási parancs végrehajtódik, azaz a monitorprogram átadja a vezérlést az általunk előzőleg begépett memóriacímen található felhasználói programnak.

## U – Unassembler – Programkód visszafejtő

„Unassembler” parancs segítségével a kódmemória tartalmát utasítás mnemonic-os szintaktikákra fejthetjük vissza.

- Az „**U**” billentyű lenyomása után a képernyőn a következő felirat jelenik meg: „552>UNASSEMBLER FROM ADDRESS:”.
- Ezt követően be kell írunk azt a memória kezdőcímet, melyről vissza szeretnénk fejteni a kódmemória.

- Az ENTER gomb megnyomását követően a képernyőn megjelenik az első visszafejtett utasítás.

Az utasítássor a következő adatokat tartalmazza:

- Utasítás kezdőcíme.
- Az utasítást tartalmazó byte-ok hexadecimális formában.
- Az utasítás mnemonik és paraméterk.

Azoknál az utasításoknál melyek SFR byte-okra vagy bit-ekre hivatkoznak nem a hivatkozási cím jelenik meg, hanem azon a memória területre vonatkozó SFR elnevezés.

Az utasítássoron kívül egy segítségssor is láthatóvá válik, mely azokat az alparancsokat tartalmazza, amelyeket az Unassembler parancson belül használhatunk. Ezek a parancsok a következők:

- **ENTER** vagy **N** billentyű – következő utasítás megjelenítése
- **A** billentyű – új kezdőcím megadása
- **ESC** vagy **Q** billentyű – kilépés az Unassembler-ből, visszatérés a főmenübe.

## D – Display memory – Memóriaterület listáztatás

„Display memory” parancs segítségével a mikrokontroller memóriaterületeinek (DATA, XDATA, CODE és SFR) aktuális tartalmát jeleníthetjük meg.

- A „D” billentyű lenyomása után a képernyőn a következő felirat olvasható: „552>DISPLAY MEMORY ([D]-Data, [X]-Xdata, [C]-Code, [S]-SFRs)” .
- Ezt követően a **D**, **X**, **C** vagy az **S** billentyű lenyomásával választhatjuk ki memóriaterületet.
- A memória terület kiválasztása után a program bekéri a „MEMORY FROM ADDRESS:” felirattal a listázandó memória kezdőcímét.
- A kiválasztott memóriaterület fél-laponként, azaz 128 byte-ként jelenik meg.

A megjelenő táblázat három részre bontható: Baloldalon a memória címek olvashatók, középen a memóriacímeken található byte-ok értékei és végül jobb oldalon a bájtok értékéhez rendelt ASCII karakterkészlet betűjele.

Ebben az esetben is egy segítségssor áll a rendelkezésünkre, mely azokat az alparancsokat tartalmazza, melyeket a memórialistázás közben használhatunk. Ezek a parancsok a következők:

- § **ENTER** vagy **N** billentyű – következő fél-lap megjelenítése
- § **A** billentyű – új kezdőcím megadása
- § **ESC** vagy **Q** billentyű – kilépés a memórialistázásból, visszatérés a főmenübe.

Ha az SFR memóriaterületet kilistáztatását választjuk akkor a kezdő memóriacím bekérése nélkül egyszerre jelenik meg a 80C552 mikrokontroller által használt összes, 56 SFR. Az SFR-ek listáztatása után a program automatikusan visszatér a monitorprogram főmenüjébe.

## M – Modify memory – Memóriatartalom megváltoztatása

A „Modify memory” parancs segítségével lehetőségünk nyílik adott memóriacímtől kezdődően a memóriaterület megváltoztatására.

- Az „**M**” billentyű lenyomása után a képernyőn a következő felirat olvasható: „552>MODIFY MEMORY ([D]-Data, [X]-Xdata, [S]-SFRs):”.
- Ezt követően a **D**, **X** vagy az **S** billentyű lenyomásával választhatjuk ki, hogy melyik memóriaterületet szeretnénk módosítani.
- A memória terület kiválasztása után a program bekéri módosítandó memória kezdőcímét.
- Kivételt jelent, ha az SFR memóriaterületet szeretnénk módosítani. Ebben az esetben a következő felirat jelenik meg: „SFR (NAME or ADDRESS):”. Így az SFR nevével, vagy a címével választhatunk. (Nem létező SFR név berásakor, a program „!Error: Invalid SFR name.” üzenet kiírása után visszalép a főmenübe. Abban az esetben, ha Nem létező SFR memóriacímet adunk meg, akkor a címhez felülről legközelebb álló SFR jelenik meg.)
- Ezek után byte-okként van lehetőségünk a memória tartalmának megváltoztatására. Képernyőn megjelenik a módosításra kerülő byte memóriacíme, (SFR esetén a megnevezése is), a jelenlegi értéke, és a „New value:” szövegrészahova írhatjuk az új értéket.
- A módosítás végrehajtása két módon nyugtázható:
  - Az **ENTER** billentyű leütésével beíródik az új érték, és a program felkínálja a következő című memóriahely módosítását. Ezt akkor használhatjuk, ha pl egy táblázatot töltünk ki. Amikor új érték megadása nélkül nyomjuk le az **ENTER** billentyűt, akkor a jelenlegi memóriacím tartalmának módosítás nélkül a következő memóriacímre ugrik a program.
  - A **SPACE** billentyű leütésével beíródik az új érték, de a program ugyanazon című memóriahely további módosítását kínálja fel. Ezt pl. egy adott memóriába ágyazott perifériára történő különböző értékek kiírásakor használjuk.
- Az **ESC** billentyű segítségével térhetünk vissza a főmenübe.

## C – Copy memory – Memóriaterület másolása

„Copy memory” parancs segítségével a mikrogép egy adott memóriaterületét másolhatjuk át egy másik memóriaterületre vagy terminálprogramon keresztül egy külső állományba.

- A „**C**” billentyű lenyomása után a képernyőn a következő felirat olvasható: „552>COPY MEMORY FROM ([D]-Data, [X]-Xdata, [C]-Code):”.
- Ezt követően a **D**, **X** vagy a **C** billentyű lenyomásával választhatjuk ki, melyik memóriaterületről szeretnénk másolni.
- A memória terület kiválasztása után a program bekéri „forrásterület” kezdőcímét, majd a végcímét (A végcímként megadott byte is átmásolásra kerül.).
- A „COPY MEMORY TO ([D]-Data, [X]-Xdata, [F]-File):” szöveg megjelenése után **D**, **X** vagy a **F** billentyű lenyomásával választhatjuk ki, azt a memóriaterületet vagy állományt ahova másolni szeretnénk.

- Ha memóriaterületre másolunk, akkor a program bekéri a „célterület” kezdőcímét.
- Ha célterületként állományt választunk ki, akkor a program először a célállomány nevét kéri be, mely maximálisan 8+3 karakter hosszúságú lehet, majd a monitorprogram átadja a vezérlést a terminálprogramnak. Sikeres másolás esetén a monitorprogram „OK.” felirattal tér vissza a főmenübe.

### **Hibalehetőségek:**

Ha forrásterület megadása esetén nagyobb kezdőcímet adunk meg mind végcímet, akkor a monitorprogram a következő hibaüzenettel tér vissza: „!Error : Start address is greater then stop address.”

Ha kiválasztott forrásterület nagyobb, mint a fennmaradó célterület nagysága akkor a program a következő hibaüzenettel tér vissza: „!Error : Source is bigger then target space.”

Abban az esetben, ha célterületnek állományt választunk, és a megjelenő terminálprogram dialógusablakában a CANCEL gombra kattintva leállítjuk a másolást, akkor „Canceled.” felirattal tér vissza a monitorprogram a főmenübe.

## **F – Fill memory – Memóriaterület kitöltése**

A „Fill memory” parancs segítségével lehetőségünk nyílik adott memóriacímtől kezdődően a memóriaterületet egy adattal feltölteni.

- Az „F” billentyű lenyomása után a képernyőn a következő felirat olvasható: „552>FILL MEMORY ([D]-Data, [X]-Xdata):”.
- Ezt követően a **D** vagy az **X** billentyű lenyomásával választhatjuk ki, memóriaterületet.
- A memória terület kiválasztása után a program bekéri „forrásterület” kezdőcímét, majd a végcímet (A végcímként megadott byte is módosításra kerül.).
- A „FILL MEMORY WITH CHAR:” szövegrész megjelenése után kell feltöltő adat értékét megadni

Sikeres feltöltés befejeztével a monitorprogram „OK.” felirattal tér vissza a főmenübe. Ha memóriaterület megadása esetén nagyobb kezdőcímet adunk meg mind végcímet, akkor a monitorprogram a következő hibaüzenettel tér vissza: „!Error : Start address is greater then stop address.”

## **I – Inputs monitoring – Bemenetek folyamatos figyelése**

„Inputs monitoring” parancs a mikrokontroller egy kiválasztott port-jára, vagy analóg bemenetére, illetve egy memóriába ágyazott bemeneti perifériára jutó jel folyamatos megfigyelésére alkalmas.

- Az „I” billentyű lenyomása után a képernyőn a következő felirat jelenik meg: „552>MONITOR ([P]-Ports, [A]-Analog inputs, [X]-Xdata peripheras):”.
- Ezt követően a **P**, **A** vagy az **X** billentyű lenyomásával választhatjuk ki, a folyamatosan figyelni kívánt bemenet-típust.



- **Port** kiválasztásakor a program a következő szöveggel: „DISPLAY PORT WITH NUMBER (0-5):” kéri a port számát.
  - Ezt követően megjelenik a bemeneti érték bináris és hexadecimális formában.
- **analóg bemenet** választásakor, akkor a következő sorban megjelennek az analóg bemenetek nevei, és alattuk pedig az aktuális 10 bites értékének hexadecimális (0h – 3FFh tartomány) formában.
- Memóriába ágyazott periféria választásakor a „DISPLAY BYTE IN XDATA MEMORY ON ADDRESS:” szöveggel kéri a periféria címét.
  - Ezt követően megjelenik a bemeneti érték bináris és hexadecimális formában.
- A kiválasztott bemenet értékének folyamatos kijelzése **ESC** vagy a **Q** billentyű lenyomásáig tart. Ennek hatására a program visszalép a főmenübe.

## H – Help – Segítség, felhasználói parancsok rövid összefoglalója

Az „H” billentyű lenyomása után a képernyőn egy táblázat jelenik meg, mely két fő részből tevődik össze. A táblázat bal oldalán a mikrokontrolleres gyakorló memória kiosztása található meg, míg a táblázat jobb oldalán a főmenüben használható parancsok rövid összefoglalását olvashatjuk.

## S – Step form address – Adott címtől történő lépésenkénti felhasználói programfuttatás

„Step form address” menü segítségével felhasználói programot futtathatunk lépésenként egy általunk megadott programmemória kezdőcíméről. Lépésenkénti programfuttatás esetén lehetőségünk van a memória teljes áttekintésére, megváltoztatására, végrehajtásra kerülő utasítások böngészésére, töréspont elhelyezésére.

- A „S” billentyű lenyomása után a képernyőn a következő felirat olvasható: „552>STEP FROM ADDRESS:”.
- A kettőspont után be kell írunk a felhasználói program ellenőrzésének kezdő címét.
- A kezdőcím begépelése után a monitorprogram az általunk előbb meghatározott címre ugrik, és egy teljes képernyős területet biztosít a felhasználó számára a a lépésenkénti, illetve töréspontos futtatáshoz közbeni memória, periféria ellenőrzésekhez.

Felhasználói felület alkotóelemei (3.ábra):

- A képernyő tetején egy segítségssor látható, mely a főbb parancsokat foglalja össze.
- Következő sorban az akkumlátor, B, DPTR regiszter, és a PSW regiszter bitekre lebontva látható.
- Alatta lévő képernyőrész bal oldalán a futtatandó program utasításai láthatók, míg jobb oldalon R0 – R7 regiszterek, és a PC értéke.
- A képernyő alsó felét egy multifunkcionális ablak foglalja el, mely bejelentkezéskor a lépésenkénti futtatásnál használható parancsok összefoglalását

tartalmazza. Ezen a képernyőterületen van lehetőségünk megjeleníteni a mikrokontrolleres gyakorló memóriaterületeit.

- Legelső sorban található a parancsbeviteli mező. Ebben a sorban lehet megadni az egyes parancsok paramétereit.

```

File Help 13:34:30
Terminal
Help: [N] or [Enter]-Next [G]-GoTo [Q] or [Esc]-Quit [H]-List of commands

ACC:00 B:4D DPTR:4000 CY:0 AC:0 F0:0 BANK:0 OV:0 F1:0 P:0
34BB jmp @A+DPTR R0:44 R1:02 R2:04 R3:00
4000 > ljmp 400Bh R4:00 R5:00 R6:05 R7:00
4003 mov R7, A
4004 mov R7, A PC:4000
4005 mov R7, A

Help
[Enter] or [N] - Next step [M] - Modify memory
[Esc] or [Q] - Quit [A] - new Address (*)
[G] - Goto to address [U] - page Up memory (*)
[S] - display SFRs [O] - page dOwn memory (*)
[D] - display DATA memory [W] - page doWn dissassembly
[X] - display XDATA memory [R] - Reset dissassembly address
[C] - display CODE memory [H] - this Help

552> (debug) _
F10 Menu Alt-X Quit
  
```

3.ábra.

### Lépésenkénti futtatás parancsai:

Billentyű	Angol megnevezés	Parancs leírás
ENTER vagy N	Next step	Következő utasítás
ESC vagy Q	Quit	Vissza a főmenübe
G	Goto to address	Memóriacímig történő futtatás
S	display SFRs	SFR regiszterek megjelenítése
D	display DATA memory	DATA memória megjelenítése
X	display XDATA memory	XDATA memória megjelenítése
C	display CODE memory	CODE memória megjelenítése
M	Modify memory	Memóriaterület értékének megváltoztatása
A	„display memory” Address	aktuális memóriaterület kezdőcímének megadás
U	page Up memory	aktuális memóriaterület felfelé lapozása
O	page dOwn memory	aktuális memóriaterület lefelé lapozása
W	page doWn dissassembly	futtatandó program utasításainak felfelé lapozása
R	Reset dissassembly address	futtatandó program utasításlistájának alaphelyzetbe állítása
H	this Help	segítségablak megjelenítése

## Lépésenként (utasításokként) történő programvégrehajtás

**ENTER** vagy **N** billentyű lenyomását követően a monitorprogram visszaadja a vezérlést a felhasználói programnak. Az éppen soron következő utasítás végrehajtása után ismét a monitorprogramhoz kerül a vezérlés. A képernyőn lévő adatok minden egyes lépés után frissítésre kerülnek.

## Vissza a főmenübe

**ESC** vagy **Q** (Quit) billentyű lenyomásával a felhasználói programfuttatás megszakad. A monitorprogram visszatér a főmenübe.

## Memóriacímig – töréspontig - történő futtatás

A lépésenkénti futtatás közben mód van egy megadott című utasításig (töréspont) folytonosan futtatni a felhasználói programot, majd innen folytatni az utasításenkénti működtetést. Ezt a **G** – Goto to address – parancs valósítja meg.

- **G** billentyű lenyomása után a parancsbeviteli mezőben a „GO TO ADDRESS:” szöveg jelenik meg.
- majd be kell írunk, hogy melyik memóriacímig szeretnénk a felhasználói programunkat egyfolytában futtatni. Ezt követően a megadott memóriacímig fog a felhasználói programunk folyamatosan futni, majd a törésponti cím elérésénél újból visszaveszi a vezérlést a monitorprogram.

Ha töréspont előtt meg szeretnénk szakítani a programunk futását, akkor **ESC** gomb megnyomásával bármikor visszakerülhetünk a monitorprogramba.

## Memóriaterületek aktuális értékeinek megtekintése

Minden lépés végrehajtása után megtekinthetők a különböző memóriaterületeken tárolt aktuális értékek. Így ellenőrizhetők, hogy a programunk valóban végrehajtja-e az esetleges módosításokat. Ezt szolgálják

- az **S**, **D**, **X**, **C** – display SFRs or DATA, XDATA, CODE memory – parancsok.

A programellenőrzés közben bármelyik memóriaértéket megváltoztathatjuk az **M** – Modify memory – paranccsal

- **M** billentyű lenyomását követően a parancsbeviteli mezőben a következő szöveg válik olvashatóvá: „MODIFY ([D]-Data, [X]-Xdata, [S]-SFR, [R]-Rx, [P]-PC):”.
- Ezt követően a **D**, **X**, **S**, **R** vagy a **P** billentyű segítségével adhatjuk meg, hogy a DATA vagy XDATA memóriaterületen lévő byte-t, SFR regisztert, R0-R7 regiszter valamelyikét vagy a PROGRAM COUNTER értékét szeretnénk megváltoztatni.
  - Ha a DATA vagy az XDATA memóriaterületet választjuk, akkor meg kell adnunk a módosítani kívánt byte memóriacímét, és azt követően az új értékét.
  - Ha SFR-t szeretnénk módosítani, akkor meg kell adnunk a módosítani kívánt SFR nevét vagy memóriacímét, és ezt követően az új értékét.
  - Ha Rx regisztercsaládból szeretnénk egy regisztert módosítani, akkor meg kell adnunk a módosítani kívánt regiszter számát és az új értékét.

- A PC módosítása esetében meg kell adnunk a programszámláló négyjegyű új értékét.

A képernyőn csak kisebb terület jeleníthető meg a vizsgált memóriából. A lapozást segítik a következő parancsok, amelyek csak kilistázott memóriaértékeknél hatásosak:

**A** – „display memory” Address – aktuális memóriaterület kezdőcímének megadás

A „display memory” Address parancs segítségével megadhatjuk egy újabb kezdőcím, től történő listázást.

**U, O** – page Up or dOwn memory - aktuális memóriaterület felfelé vagy lefelé lapozása

Az **U, O** billentyűvel felfelé vagy lefelé lapozhatunk az éppen a kilistázott memóriába.

**W** – page doWn dissassembly - futtatandó program utasításainak felfelé lapozása

A **W** billentyű segítségével a képernyő bal-felső részén található programlistában lapozhatunk előre.

**R** – Reset dissassembly address - futtatandó program utasításlistájának alaphelyzetbe állítása

Az **R** billentyű segítségével a képernyő bal-felső részén található programlistát állíthatjuk alaphelyzetbe.

**H** – this Help – Segítő-ablak megjelenítése

### Lépésenkénti futtatás feltételei.

A monitorprogram software-s lépésenkénti futtatást alkalmaz, mivel a 80C552 mikrokontroller nem rendelkezik hardware-s töréspont funkcióval. A program a mikrokontroller soros megszakítását használja a lépésenkénti futtatás lebonyolítására. Ebből adódóan a felhasználó által írt programot csak bizonyos megkötések mellett képes a monitorprogram lépésenként futtatni.

Mielőtt a monitorprogram elindítaná a felhasználó által írt programot, 1-be állítja a soros port TI bit-jét, a soros port megszakításának prioritás bit-jét ( $PS0 = 1$ ), a soros megszakítás engedélyező bit-jét ( $ES0 = 1$ ), és végül a globális megszakítás engedélyező bit-et ( $EA = 1$ ).

Ha ezeket bit-eket a felhasználói program megváltoztatja, abban az esetben a monitorprogram elvesztheti a felügyeletet a futtatott program felett. Ilyen esetben csak a futatott program utasításai fognak végrehajtódni, így nincs lehetőségünk visszatérni a monitorprogramba csak reset feltétel kiadásával.

A tesztelésre szánt program nem használhatja a **hármasszámú registerbank**-ot, illetve az **XDATA memória 3E00 – 3FFF** területét, mert ellenkező esetben instabil futás történhet. Amennyiben az általunk futtatott program hármasszámú registerbank-ra vált, akkor a monitorprogram egy villogó felkiáltójellel jelzi ezt a felhasználó számára az aktuális registerbank érték mellett.

Ügyelni kell arra, hogy a futatott program és a monitorprogram ugyanazt a vermet használja. A relokálható szegmens-megadásoknál biztosítani kell, hogy a verem (stack) a bank regiszterek és a használt belső memória feletti címen kezdődjék. Ennek egyik módja, hogy a verem részére legalább 20h memóriaterületet figlaltaunk le.

Töréspontos futtatás esetén a monitorprogram minden egyes utasítás végrehajtása után visszatér a monitorprogramba, és ellenőrzi hogy a futatott program el ért-e a töréspontot vagy nem. Ez program futásának a lassulását okozza. Fontos tudni, hogy töréspontos

programfuttatás alatt nem valós idejű a program végrehajtása. Ilyenkor a program futása nagyságrendekkel hosszabb időt vesz igénybe, mint tiszta hardware-s futás esetén.

Mivel a lépésenkénti ellenőrzés megszakításos működtetéssel dolgozik, ezért megszakításokat is használó programok valós idejű tesztelése ily módon nem ajánlott. Az ellenőrzés különböző beállításokkal lehetséges, de az oktatásban erre nem térünk ki, s ezért itt sem adunk használati utasítást erre.

## Memóriatartalom letöltése a mikrokontrollerbe

Memóriarész letöltése a mikrokontrollerbe a monitorprogram és a terminálprogram együttműködésével történik. Letöltéshez használt protokoll INTEL HEXA adatformátumon alapul. A letöltés „handshaking”, azaz kézfogásos kommunikációval történik. A terminálprogram leküld a monitorprogramnak egy teljes INTEL HEXA sort (kezdő „:” karakterrel együtt). A monitorprogram értelmezi a beérkezett sor, majd visszaválaszol a terminálprogramnak. Három féle válasz lehetséges:

- „O” karakter: minden rendben, a monitorprogram készen áll a következő sor vételére (OK.).
- „E” karakter: LRC (longitudinal redundancy check – hosszirányú-redundancia ellenőrzés) hiba.
- „T” karakter: Ismeretlen rekord típus.

Mind a három válaszkarakter után a monitorprogram készen áll a következő sor vételére. Kivételt jelenet ez alól az „állomány vége” típusú rekordot, mely után a monitorprogram „OK.” kírással visszalép a főmenübe.

Fontos még megjegyezni, hogy a főmenüben történő „:” karakter lenyomását a monitorprogram letöltési procedúra kezdetének értelmezi. Ebben az esetben a felhasználó számára, visszajelzés hiányában, úgy tűnhet, hogy megállt reagálni a monitorprogram.

## Az Intel hexa formátum (3.táblázat)

Az Intel hexa formátumnál az adatok rekordokba szerveződnek. Egy rekord változó hosszúságú lehet.

	1.	2.	3.	4.	5.	6.
Rekord megnevezése	Record Mark Field	Record Length Field	Load Address Field	Record Type Field	Data Field	Checksum Field
Hosszúság	1 byte	2 byte	4 byte	2 byte	függ a 2. mező tartalmától	2 byte
Példa	:	10	2000	00	455A542045544454D455320564F4C54	53

3.táblázat

A rekordban az első mező a **rekordjelző mező (Record Mark Field)**. Ez a mező jelzi a rekord kezdetét. A rekordjelző mező egy ASCII kettőspont karakterből áll.

Második mező a **rekordhossz mező (Record Length Field)**. Ez a mező két ASCII karaktert tartalmaz, melyek jelzik a rekordban lévő adatbájtok számát. Az adatbájtok számát két ASCII karakterre konvertálva adódik ki a mező két karaktere, mely egy hexadecimális szám. A magasabb helyértékű számjegy szerepel előbb. Ebből adódik, hogy egy rekord maximálisan 255 adatbajt hosszúságú lehet.

Harmadik mező a **betöltési cím mező (Load Address Field)**. Ez a mező négy ASCII karaktert tartalmaz. A négy karakter hexadecimális szám formájában tartalmazza az adott rekord betöltési kezdőcímét. A állomány vége rekord esetén a címrekord vagy csak nullákat tartalmaz, vagy a program kezdőcímét.

Negyedik mező a **rekord típus mező (Record Type Field)**. A mező két karakterből áll, mely egy hexadecimális számot határoz meg. Az Intel hexa alapformátum, csak két rekordtípust különböztet meg:

- 00 – Adat rekord (Data record)
- 01 – Állomány vége rekord (End Of File record)

Ötödik mező az **adat mező (Data Field)**. Ez a mező tartalmazza a betöltendő adatokat két-két ASCII karakteré konvertálva, melyek hexadecimális szám formájában mutatják a betöltendő byte-ok értékeit. Állomány vége rekord esetén nincs adat mező.

Az utolsó mező az **ellenőrző összeg mező (Checksum Field)**. Az ellenőrző összeg a második, harmadik, negyedik és az ötödik mező hexadecimális byte-jainak nyolc bit-re csonkított összegének kettes komplemente (LRC). Az így kapott hexadecimális összeget két ASCII karakterre konvertálva tartalmazza a mező.

A leírásból is látható, hogy egy Intel hexa formátumú állomány egy szöveges ASCII állomány, így például egy szövegszerkesztő vagy listázó program segítségével megvizsgálhatjuk a tartalmát. Következő táblázat egy Intel hexa formátumú sort értelmez.

## Memóriatartalom küldése a terminálprogramnak

A monitorprogram adat küldési protokoll segítségével teszi lehetővé, hogy egy memóriaterületet le lehessen menteni számítógépen keresztül háttértárolóra. Ezt a lehetőséget a „Copy memory” parancs biztosítja a felhasználó számára. Az adatküldési protokoll egyoldalú, az-az nincs visszajelzés a terminálprogram oldaláról, hogy sikeres volt az adatfogadás vagy nem. Az adatküldési protokoll kimondottan oktatási célra lett kifejlesztve. A protokoll egyszerűsége végett könnyű a protokollt lekezelő program elkészítése, de pont a protokoll könnyűsége miatt zajérzékeny. Feltételezve, hogy a monitorprogram oktatási intézményben kerül felhasználásra, ezért a zajból keletkezett hibák szinte kizárhatók. Sebesség problémák fellépésétől is nyugodtan el tekinthetünk, hisz a fogadó oldalon lévő PC sebessége többszöröse a küldő oldalon szereplő mikrokontroller sebességének.

A protokollt két részre bontható. Az első rész definiálja az adtafolyam típusát, majd a második rész a küldeni kívánt adatokat tartalmazza.

Az protokoll egy ESC (←), azaz 1Bh sorszámú ASCII karakterrel kezdődik, majd egy nagy „S” betűvel folytatódik. E két karakter jelzi, a terminálprogramnak, hogy adat letöltés fog következni.

Második rész az állománynév megadására szolgál. Idézőjelek (22h ASCII számú karakter) közé besúrva áll módunkban megadni a terminálprogramnak, hogy milyen állománynévvel történjen az adatfolyamunk mentése. Az állománynév megadása nem kötelező, csak választható opció.

Következő mező az adatfolyam típusát tartalmazza. Két adatfolyam típust definiál a protokoll:

- „B” – Byte – Az adatfolyam bármilyen karaktert tartalmazhat, nincs kiemelt stop karakter. Az adatfolyam hosszúságát közvetlenül a „B” karakter után kell megadni,

melyet „H” karakterrel lezárt ASCII karakterből álló hexadecimális számsor határozza meg.

- „T” – Text – Az adatfolyamnak nincs előre definiált hossza. Az adatok végét egy EOT (End Of Text), azaz 04h karakter jelzi.

Példák:

- Előre nem definiált adathossz mentése proba.txt állományba: ←S”proba.txt”T
- 32 byte hosszúságú adattömb fejléce. Mentés proba2.txt állományba.: ←S”proba2.txt”B20H
- Előre nem definiált adathossz. Állománynév nincs megadva.: ←ST
- 128 byte hosszúságú adattömb fejléce. Állománynév nincs megadva.: ←SB80H

A fejléc elküldése után a terminálprogram visszaválaszol a kérésre. Két válaszlehetőség lehetséges:

- „O” – Minden rendben, jöhetnek az adatok.
- „C” – A felhasználó visszautasította a kérést. Nem lehet adatokat menteni.

Pozitív válaszadás esetén a terminálprogram készen áll az adatsor fogadására, a mikrokontroller által küldött adatok mentésre kerülnek a háttértárolóra.

## Az ANSI képernyő formázó kódok használata

A terminálprogram az ANSI kódokat értelmezi, és ennek megfelelően vezérli a képernyőt.

### ANSI terminálutasítások leírása

<i>Utasítás szintaktika</i>	<i>Utasítás megnevezése</i>	<i>Utasítás leírás</i>
<b>ESC[PL;PcH</b>	Kurzor pozíció beállítás	Kurzor helyének beállítása koordináták alapján. Ha nincs paraméter megadva, kurzor a jobb felső sarokba kerül
<b>ESC[PL;Pcf</b>	Kurzor pozíció beállítás	Teljes mértékben hasonlóképpen működik mint az előző parancs.
<b>ESC[PnA</b>	Kurzor fel	Felfelé mozgatja a kurzort, a paraméterben megadott sorok számával, úgy hogy a kurzor koordinátájának oszlopszámát nem változtatja meg. Ha a kurzor a legfelső sorban van, a terminál figyelmen kívül hagyja parancsot.
<b>ESC[PnB</b>	Kurzor le	Lefelé mozgatja a kurzort, a paraméterben megadott sorok számával, úgy hogy a kurzor koordinátájának oszlopszámát nem változtatja meg. Ha a kurzor a legalsó sorban van, a terminál figyelmen kívül hagyja parancsot.
<b>ESC[PnC</b>	Kurzor előre	Jobbra mozgatja a kurzort, a paraméterben megadott oszlopok számával, úgy hogy a kurzor koordinátájának sor számát nem változtatja meg. Ha a kurzor a képernyő jobb oldalán van, a terminál figyelmen kívül hagyja parancsot.
<b>ESC[PnD</b>	Kurzor hátra	Balra mozgatja a kurzort, a paraméterben megadott oszlopok számával, úgy hogy a kurzor koordinátájának sor számát nem változtatja meg. Ha a kurzor a képernyő bal oldalán van, a terminál figyelmen kívül hagyja parancsot.
<b>ESC[s</b>	Kurzor pozíció mentése	Elmentik a kurzor jelenlegi állását. A "Kurzor pozíció visszaállítása" parancssal lehet a kurzort az elmentett pozícióra mozgatni.
<b>ESC[u</b>	Kurzor pozíció visszaállítása	A kurzor a "Kurzor pozíció mentése" parancs által elmentett kurzorpozícióra ugrik.
<b>ESC[2J</b>	Képernyőtörlés	Törli a képernyőt, és a kurzort a jobb felső pozícióba (sor=0; oszlop=0) mozgatja.
<b>ESC[K</b>	Sortörlés	Törli az összes karaktereket a kurzor pozíciótól számítva a sor végéig, beleértve a kurzor pozícióján lévő karaktert is.
<b>ESC[Ps;...;Psm</b>	Grafikai üzemmód beállítása	Beállítja a szöveg megjelenésének típusát, színét, és háttérszínét. Beállítás a következő grafikai üzemmód beállításáig változatlan marad. A paraméterek értelmezését a következő három táblázat tartalmazza

Minden ANSI parancs ESC (1Bh) karakterrel kezdődik.



### ANSI Paraméterek

<i>Megnevezés</i>	<i>Leírás</i>
<b>Pn</b>	Numerikus paraméter tízes számrendszerben értelmezve.
<b>Ps</b>	Funkcióválasztó számparaméter, mely tízes számrendszerben van értelmezve. Több funkcióválasztó paraméter is megadható egymás után, melyeket pontosvesszővel (;) kell elválasztani egymástól.
<b>PL</b>	Sor paraméter. Kurzor sor paramétere, mely tízes számrendszerben van értelmezve.
<b>Pc</b>	Oszlop paraméter. Kurzor oszlop paramétere, mely tízes számrendszerben van értelmezve.

### Szöveg tulajdonságok

<i>Ps</i>	<i>Leírás</i>
<b>0</b>	Minden tulajdonság kikapcsolása
<b>1</b>	Magas intenzitás be
<b>4</b>	Aláhúzás be (csak fekete-fehér képernyő esetén)*
<b>5</b>	Villogás be
<b>6</b>	Inverz szín be
<b>7</b>	Rejtett be*

\* A terminál program nem támogatja

#### Szöveg szín

<i>Ps</i>	<i>Szín</i>
<b>30</b>	Fekete
<b>31</b>	Vörös
<b>32</b>	Zöld
<b>33</b>	Sárga
<b>34</b>	Kék
<b>35</b>	Bíbor
<b>36</b>	Kékeszöld
<b>37</b>	Fehér

#### Háttér szín

<i>Ps</i>	<i>Szín</i>
<b>40</b>	Fekete
<b>41</b>	Vörös
<b>42</b>	Zöld
<b>43</b>	Sárga
<b>44</b>	Kék
<b>45</b>	Bíbor
<b>46</b>	Kékeszöld
<b>47</b>	Fehér