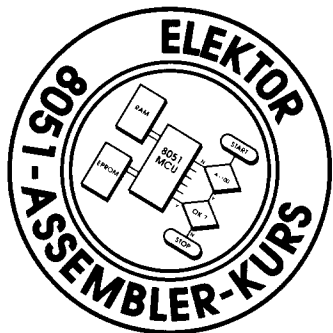


# 8051-es Mikrokontroller- és Assembler-tanfolyam

## 7. (befejező) rész: LC kijelzők csatlakoztatása és vezérlése. Billentyűzet csatlakoztatása, kitekintés



Tanfolyamunk befejező részében először az LC kijelzők csatlakoztatásával foglalkozunk. Az utóbbi időben viszonylag jutányosan beszerezhetővé vált folyadékkristályos kijelzők és a mikrovezérlő együttes alkalmazása kényelmes kijelzési és kezelési lehetőségekkel rendelkező készülékek építését teszi lehetővé. Végül egy pillantást vetünk a tanfolyamunkon már túlmutató programozási segédesszközökre és nagy általánosságban a mikrovezérlők programozásával kapcsolatos alapvető tudnivalókra.

### LC kijelzők

Számos Stand-Alone mikroprocesszor alkalmazás igényel rövid szövegek vagy mérési értékek megjelenítése céljából valamilyen output lehetőséget. Ezen a téren az elmúlt időszakban már viszonylag kedvező áron beszerezhetővé vált alfanumerikus LC kijelzők használata jöhet elősorban szóba. Ezek az LCD-k számos kereskedelmi termékben is széles körű alkalmazásra találtak. Ilyen kijelzővel a felhasználó tájékoztatására szolgáló megoldások is kialakíthatók és komplex rendszerkiszolgálási folyamatok válnak már néhány gomb segítségével megoldhatóvá. Az LC kijelző első üzembe helyezésének lehető legegyszerűbbé tétele céljából a többfunkciós kártyát már elláttuk LC csatlakozási lehetőséggel.

### Összekötések

A bővítőkártya és a (Hitachi HD 44780-as vagy vele ekvivalens vezérlővel működő)

kijelző közötti elektromos összekötések 14-eres lapos szalagkábel útján valósulnak meg. Mivel a különböző típusú kijelzők kivezetéseinek bekötése is eltérő, itt a többfunk-

ciós kártya K1 csatlakozójának érintkező-sorszámát adjuk meg.

A csatlakoztatás módját vázlatosan az 1. ábrán mutatjuk be. A 0C001<sub>16</sub>, illetve 0C009 címre történő beírás (WR = 1) vagy kiolvasási (RD = 1) hozzáférés esetén előáll a kijelző számára az Enable jel. Hogy a kijelzőn belüli adatokhoz való hozzáférésről (RS

= 10A3 = 1, cím= 0C009<sub>16</sub>) van-e szó, vagy utasításokat (RS=10A3=0, cím=0C001<sub>16</sub>) kell-e átadni, azt az 10A3 címvezeték állapota határozza meg. Az LC kijelző R/W vezetéket közvetlenül a Compu-board RD kimenetével kell összekötni.

Parancsok és adatok átadása a kétirányú adatbusz útján történik. Maga a kijelző 4-bites adatátvitellel (0...3-as bit) vagy 8-bites adatátvitellel működhet. Mivel 8-bites mikrovezérlőt használunk és a kijelző az adatbuszra kötve fog üzemelni, kézenfekvőnek látszik a kijelzőt úgy programozni, hogy az 8-bites átvittel működjék. Ha a kijelzőt például egy PORT-on át kívánánk üzemeltetni, akkor a 4-bites átvitel volna célszerűbb, mert az kevesebb portvezetéket vesz igénybe.

A 3-as kivezetésre adott V<sub>0</sub> feszültség a kijelző kontrasztjának beállítására szolgál. A beállítás a helyiség megvilágításához és a rálátás szögéhez illeszthető. A kijelző rákötése előtt a biztonság kedvéért annak adatlapját mindig tüzetesen át kell nézni és valamennyi összekötést ellenőrizni kell.

### LCD utasítások

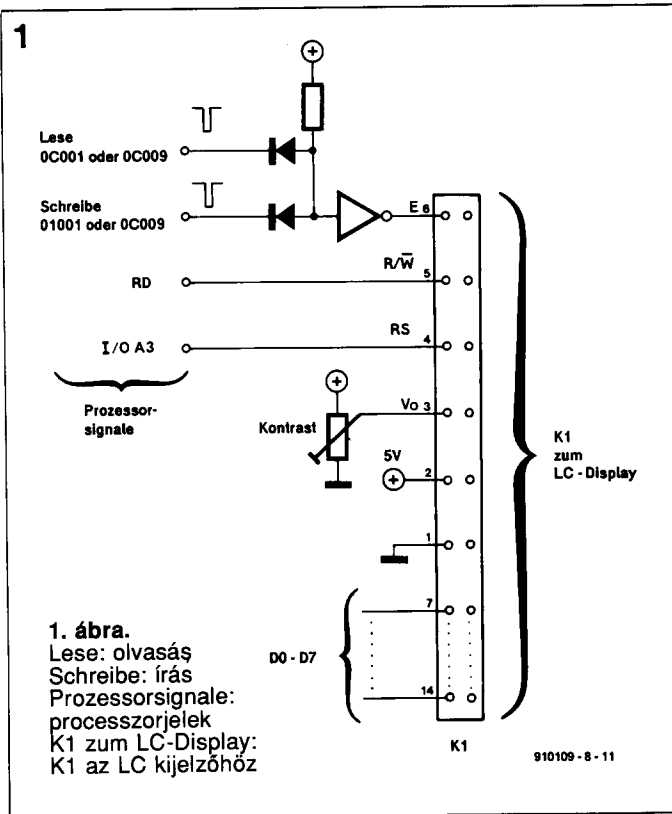
A kijelző és az integrált vezérlő kombinációs lehetőségeinek teljes leírása a Hitachi katalógusában csaknem 30 oldalt vesz igénybe. Mi ezért csak a fontosabb utasításokra szorítkozunk. Akinek ezen túlmenő információkra van szüksége, az a gyártó által közölt adatokhoz nyúlhat vissza.

ASCII karakterek megjelenítésére sok LC kijelző ugyanazt a típusú LCD vezérlőt használja és ennek megfelelően ugyanazzal az utasításkészlettel rendelkezik. Egy komplett utasításkészletet a 2. ábrán mutatunk be. Ez pl. a H 2570, LM 016L és LM 1612A típusú kijelzőkre érvényes.

Mindenekelőtt álljon itt néhány alapvető tudnivaló az ilyen kijelzőnek az integrált vezérlővel együtt történő alkalmazásáról.

A kijelző belső tárolója 80 karakter tárolására alkalmas (Display data RAM = DD-RAM, címei 000<sub>16</sub>-tól 04F<sub>16</sub>-ig terjednek). Egy 16-karakteres, egysoros kijelző balról kezdve a 000-tól 00F<sub>16</sub>-ig terjedő címek alatti karaktereket jeleníti meg. A kijelzett karakterek tehát az összes tárolt karakterből csak egy szletet („ablakot”) képviselnek. Ez a szlet a Display-Shift útján eltolható. A karakterek a DD-RAM-ban ennek során ugyanazon a címen maradnak, csak a kijelzőablak kezdődik a DD-RAM más címen.

Az LM 16255-ös kétsoros kijelzőnél az első sorban a 000<sub>16</sub> című kezdődő karakterek, a második sorban a 040<sub>16</sub>



1. ábra. Lese: olvasás  
Schreibe: írás  
Prozessor-signale: processzorjelek  
K1 zum LC-Display:  
K1 az LC kijelzőhöz

1. ábra. Az LCD csatlakozás vázlatos ábrázolása

2. ábra. Az LCD vezérlő utasításkészletének vázlatos áttekintése

Instruction	Code								Description	Execution Time (when f <sub>clk</sub> or f <sub>osc</sub> is 250KHz)		
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2			DB1	DB0
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears all display and returns the cursor to the home position (Address 0)	82 μs ~ 164ms
Return Home	0	0	0	0	0	0	0	0	0	*	Returns the cursor to the home position (Address 0). Also returns the display being shifted to the original position. DD RAM contents remain unchanged.	40 μs ~ 16ms
Entry Mode Set	0	0	0	0	0	0	0	1	MD	S	Sets the cursor move direction and specifies or not to shift the display. These operations are performed during data write and read.	40 μs
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	Sets ON/OFF of all display (D). Cursor ON/OFF (C) and blink of cursor position character (B).	40 μs
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	*	*	Moves the cursor and shifts the display without changing DD RAM contents.	40 μs
Function Set	0	0	0	0	1	DL	N	F	*	*	Sets interface data length (DL), number of display lines (L), and character font (F).	40 μs
Set CG RAM Address	0	0	0	1	A <sub>CG</sub>					Sets the CG RAM address. CG RAM data is sent and received after this setting.	40 μs	
Set DD RAM Address	0	0	1	A <sub>DD</sub>					Sets the DD RAM address. DD RAM data is sent and received after this setting.	40 μs		
Read Busy Flag & Address	0	1	BF	AC					Reads Busy flag (BF) indicating internal operation is being performed and reads address counter contents.	1 μs		
Write Data to CG or DD RAM	1	0	Write Data								Writes data into DD RAM or CG RAM.	40 μs
Read Data to CG or DD RAM	1	1	Read Data								Reads data from DD RAM or CG RAM.	40 μs
I/D = 1 Increment I/D = 0 Decrement S = 1 Accompanies display shift S/C = 1 Display shift S/C = 0 Cursor move R/L = 1 Shift to the right R/L = 0 Shift to the left DL = 8 bits DL = 4 bits N = 12 lines N = 0 1 line F = 1 5 x 10 dots F = 0 5 x 7 dots BF = 1 Internally operating BF = 0 Can accept instruction												
DD RAM: Display data RAM CG RAM: Character generator RAM A <sub>CG</sub> : CG RAM address A <sub>DD</sub> : DD RAM address Corresponds to cursor address AC: Address counter used for both of DD and CG RAM address												
Execution time changes when frequency changes (Example) When f <sub>clk</sub> or f <sub>osc</sub> is 250 KHz 250 μs = 37μs 270 μs = 37μs												

címtől kezdődő karakterek jelennek meg. Emiatt a két-soros kijelzők kezelése valamivel nehezebb.

Rendelkezésre áll egy kurzor is, mely meghatározza, hogy a következő karakter a DD-RAM-ban melyik helyre kerüljön (a továbbiakban ezt címszámlálónak nevezzük). A kurzor a programozástól függően a kijelzésen akár látható, akár láthatatlan lehet. Villogó kurzor használata is lehetséges. Ezenkívül szoftver úton határozható meg az is, hogy egy karakter átadása után az LC kijelzőn bekövetkezzék-e automatikus léptetés és/vagy kurzorelmozdulás. Ezáltal például futó feliratok valósíthatók meg egyszerű módon.

Végül az LC kijelző még karaktergenerátor RAM-mal (Character-Generator-RAM = CG-RAM) is rendelkezik, melyben a karakterek képelemekből történő felépítése 0-tól 7-ig terjedő ASCII kódokban tárolható. Ennek a CG-RAM-nak a pontos felosztása a gyártó adatlapjából ismerhető meg. A CG-RAM segítségével önmagunk is definiálhatunk saját célra felhasználói karaktereket.

Az LC kijelzőre parancsok úgy küldhetők, hogy a kívánt utasításkódot a 0C001<sub>16</sub> címre adjuk ki (azaz RS alacsony és R/W szinten 0).

A fontosabb utasításkódok a következők (\* az adott bit pozícióján tetszőleges szintet jelent):

**Clear Display**  
0 0 0 0 0 0 1

Ez az utasítás valamennyi DD-RAM tárekeszbe a 20<sub>16</sub> bájt (= üres jel) írja be. Hatására a kurzor a 0 pozícióra kerül és a kijelzőablak szintén a 0 címtől kezdődik, ami egy esetleges megelőző Display-Shift utasítás törléséhez vezet.

**Return Home**  
0 0 0 0 0 0 1\*

Ez a parancs a kurzort a 0 pozícióba viszi és a megelőző Display-Shift-et nullázza. A DD-RAM tartalma változatlan marad.

**Entry Mode Set**  
0 0 0 0 0 1 I/D S

Ez az utasítás annak meghatározására szolgál, hogy egy adatbájt átvitele után mi történjék a kijelzőn. Az I/D (Increment/Decrement) bit azt írja elő, hogy egy jelnek a DD-RAM-ba történő beírása során a belső DD-RAM cím automatikusan növekedjék (I/D = 1) vagy csökkenjen (I/D = 0). E cím értékének tárolása az AC címszámlálón (Address Counter) történik.

Az S léptetőbit (Shift-Bit) azt adja meg, hogy a DD-RAM-ba történő beírásor a I/O bit által előírt irányban a

kijelzés egy lépéssel automatikusan eltolódjék-e. Itt S=1 azt jelenti, hogy léptetni kell, S=0 esetében a kijelzés léptetése elmarad. A léptetés során a kurzor a kijelzőablakban azonos helyen marad.

**Display ON/OFF Control**  
0 0 0 0 1 D C B

Ezzel az utasítással a kijelzés, illetve a kurzor a DD-RAM tartalmának megváltoztatása nélkül be-, illetve kikapcsolható. A kijelzőbit (Display-Bit) D = 1 értéke esetén a kijelzés bekapcsolódik, egyébként kikapcsolódik, illetve kikapcsolt állapotban marad. A kurzorbit (Cursor-Bit) C = 1 értéke esetén a kurzor bekapcsolódik, egyébként kikapcsolódik, illetve megtartja kikap-

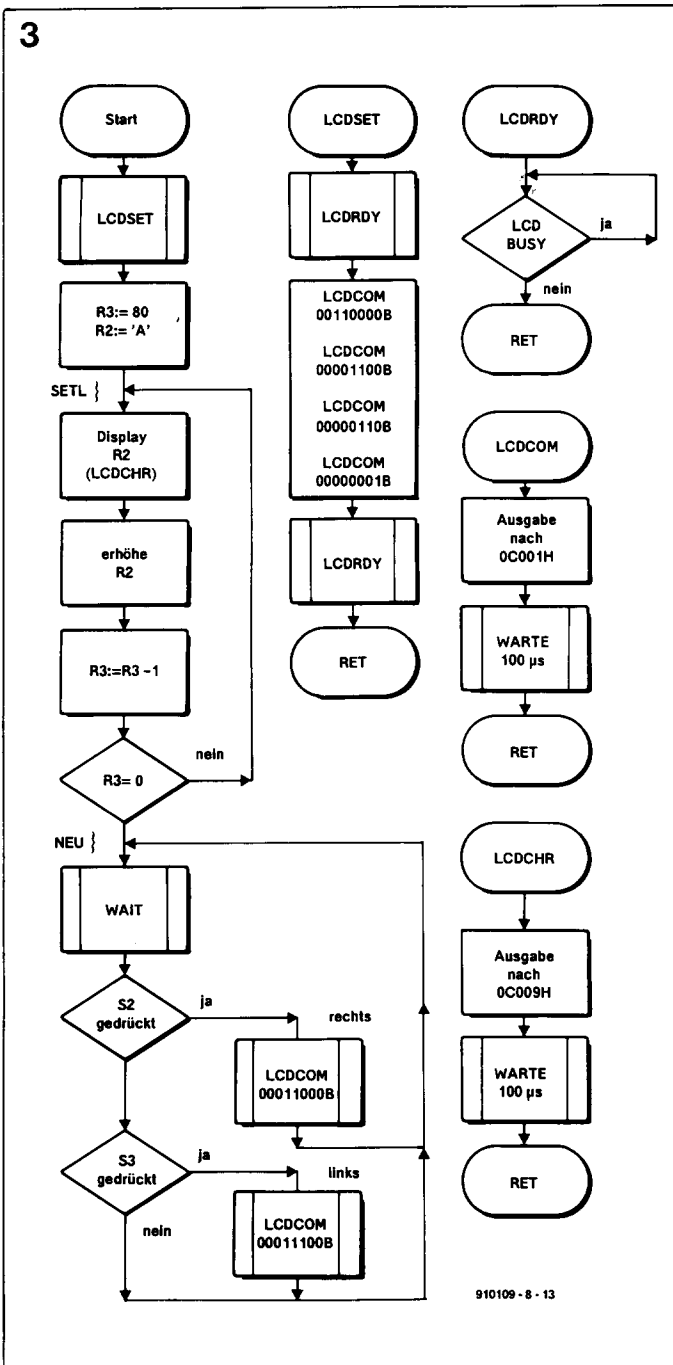
csolt állapotát. A villogási bit (Blink-Bit) B = 1 értéke mellett a kurzor villog, egyébként nem.

**Cursor or Display Shift**  
0 0 0 1 S/C R/L \*\*

Ezzel az utasítással a kurzor mozgatható vagy a kijelzés toltathó el. Az utasítást például futó kijelzések keltésére lehet használni.

S/C R/L  
0 0 kurzor balra  
0 1 kurzor jobbra  
1 0 kijelzés balra,  
a kurzor a kijelzést követi  
1 1 kijelzés jobbra,  
a kurzor a kijelzést követi

3. ábra. Az LC kijelző program folyamatábrája



**Function Set**  
Parancs =  
0 0 1 DL N F \*\*  
DL = 1: 8-bites  
DL = 0: 4-bites interfész  
N = 0: egy sor  
N = 1: két sor  
  
(csak egyes típusoknál:)  
F = 0: 5x7-es pontmátrix  
F = 1: 5x10-es pontmátrix

Ezzel az utasítással történik a kijelzés bekapcsolási reset utáni üzemmódjának meghatározása. A tanfolyamon a kijelző felé 8-bites adatátvitelt alkalmazunk és egysoros kijelzési módban dolgozunk.

**Set CG RAM Address**  
0 1 a5 a4 a3 a2 a1 a0

Ez a parancs a karaktergenerátor RAM felé történő adatátvitel előkészítésére szolgál. A következő átvinni kívánt bájt CG címét határozza meg. A címszámlálón (AC) átvivendő címet az a0-tól a5-ig terjedő bitek képezik.

**Set DD RAM Address**  
1 a6 a5 a4 a3 a2 a1 a0

Ez az utasítás a Display-Dat-RAM-ba történő adatátvitelt készíti elő. A következő átvinni kívánt bájt DD-RAM címét határozza meg. A címszámlálón (AC) átvivendő címet az a0-tól a6-ig terjedő bitek képezik.

**Read Busy Flag**  
Olvasva: BF a6 a5 a4 a3 a2 a1 a0  
(R/W = 1)

BF értéke azt adja meg, hogy a kijelző foglalkozik-e még egy utasítás végrehajtásával vagy készen áll-e további utasítások, illetve adatok fogadására. Egyidejűleg sor kerül a címszámláló értékének kiolvasására is.

**Write DATA to CG or DD Ram**  
Adatok: d7 d6 d5 d4 d3 d2 d1 d0  
(R/W = 0, RS = 1)

Ezzel az utasítással egy bájt kerül átvitelre a CG- vagy DD-RAM-ba, attól függően, hogy ezt megelőzően CG- vagy DD-cím átvitelére került-e sor. Hogy a bájt átvitele után a címszámláló tartalma nő-e vagy

3. ábra.  
erhöhe R2: R2 növelése  
nein: nem  
WAIT: várakozás  
S2 gedrückt: S2 megnyomva  
ja: igen  
S3 gedrückt: S3 megnyomva  
rechts: jobbra  
links: balra  
Ausgabe nach 0C001H: output 0C001-ra  
Warte 100 ms: 100 ms várakozás  
Ausgabe nach 0C009H: output 0C009-ra

csökken, az az ENTRY-MODE utasításban meghatározott üzemmódtól függ.

**Read DATA from CG or DD Ram**

Adatok: d7 d6 d5 d4 d3  
d2 d1 d0  
(R/W = 1, RS = 1)

Ezzel az utasítással bájtok olvashatók ki a CG- vagy DD-RAM-ból. Az utasítás előtt SET CG ADDRESS, illetve SET DD ADDRESS útján a címét át kell vinni.

**A kijelző tesztelése**

Az LCD vezérlő parancsok megtárgyalása után most a kijelző tesztelő programjára térünk át. A program folyamat-ábrája a 3. ábrán látható, a hozzá tartozó programot a 4. ábra assembler listáján mutatjuk be. A program feladata egyszerű: először beírunk a kijelzőbe egy szöveget, majd a kijelző tartalmát két nyomógomb segítségével balra és jobbra tologathatjuk.

A példaprogram olyan segédprogramokat tartalmaz, amelyek a saját készítésű programok alapjául is szolgálhatnak. Ezek képezik az LC kijelzővel történő kapcsolattartás úgynevezett kommunikációs alaprutinjait. Először ezeket beszéljük meg.

**Programozás**

Az RCOM alprogram az LC kijelző állapotát az akkumulátorba olvassa be. Ebből a célból a P2 portba kerül beírásra a kijelző 0C01<sub>16</sub> címéről a nagyobb helyiértékű bájt. A cím kisebb helyiértékű bájtja az R0 regiszterbe kerül betöltésre. Ezután a MOVX utasítás segítségével sor kerül az LC kijelzőből a BUSY-FLAG kiolvasására. Ez képezi az akkumulátor 7-es bitjét. Mint már említettük, a külső tároló közvetett címzése során a cím nagyobb helyiértékű részeként a 2-es portot használjuk.

Az LCDRDY alprogram mindaddig vár, míg a kijelző BUSY-FLAG-je 0 értékű. Ehhez felhasználja az RCOM alprogramot. Ezt az alprogramot az LC kijelző viszonylag hosszú ideig tartó inicializálási utasításainak időtartama alatt arra a várakozásra használjuk fel, mely addig tart, míg az LC kijelző az átvitt utasítást végre nem hajtotta.

A WT1 alprogram 100 mikroszekundumot várakozik. Mivel a legtöbb kijelző-utasítás végrehajtási ideje 100 mikroszekundum alatt van, ez az alprogram a kijelző-utasítások végrehajtásának kivárására jól használható.

Az LCDCOM alprogram egy parancsot (RS = 0) küld a kijelzőre. A MOVX utasításnál ugyanaz a címzés kerül felhasználásra, mint az RCOM alprogramban. A parancs átvii-

tele után megint 100 mikroszekundum szünetet követnek.

Az LCDCHR alprogram egy karaktert küld (RS = 1) a kijelző karaktertárolójába (DD-RAM), majd 100 mikroszekundumot vár. Az LCDCHR behívása előtt adott esetben egy SET RAM ADDRESS utasítással az új DD-RAM címet át kell vinni.

Az LCDSET alprogram az LC kijelző valamennyi fontos funkciójának meghatározására szolgál. Először (LCDRDY behívásával) kivárja valamennyi előző utasítás végrehajtását. Ezután következik (a 37-38-as sorban) a kijelző működési módjának meghatározása: 8-bites, egysoros és 5x7 pontos. Ehhez az LCDCOM alprogram kerül felhasználásra. Következő lépésként a kijelző be- és a kurzor kikapcsolása következik (39-40-es sor), míg a léptetőmódus beállítása a 41-42-es sorokban történik. Ekkor kerül sorra a kijelző törlése. Mivel ez az utasítás akár 1,6 ms-ig is tarthat, utána még végrehajtásra kerül az LCDRDY alprogram.

A figyelmes olvasónak talán már feltűnt az alábbi programozási trükk. Ha az 1-es alprogram a következő utasítás-sorozattal végződik:

LCALL 2-es alprogram  
RET

akkor ehelyett valamivel helytakarékosabb módon a következőképpen programozhatunk:

LJMP 2-es alprogram

és így a 2-es alprogram RET utasítását használhatjuk (a lista 52-es sora). A helytel való takarékoskodás e módját azonban semmi esetre sem szabad túlzásba vinni, mert zavart okozhat. Itt csak példaképpen mutattuk meg mint érdekességet.

**A főprogram**

A kijelző definiált állapotba hozásáért először az LCDSET-et hívja be. Ezt követően beírja az ABCDEFG.....80 karaktert a kijelző RAM-ba. Ezután a NEU végtelen hurok következik. Ebben a hurokban az kerül tesztelésre, hogy az S2 vagy az S3 gomb meg van-e nyomva. Amennyiben igen, úgy a kijelzés egyet lép balra, illetve jobbra. E célból LCDCOM útján a megfelelő utasítás eljut a kijelzőhöz. Annak érdekében, hogy folyamatosan megnyomva tartott gomb esetén a kijelző tartalma csak lassan haladjon, minden hurokvégigjáráskor behívásra kerül a WAIT alprogram, mely mintegy 0,26 másodpercet (255 x 255 x 4 µs-ot) lázas semmittevéssel tölt el.

**Feladat**

A jelenlegi és egyben utolsó programozási feladatunk a

következő: Szöveget és (hexadecimális és decimális) számokat kell az LC kijelzőre kiadni. Az EMON51.LST monitorlistában is utánuk lehet nézni, hogyan oldották meg ott a V24-en át megvalósuló output programozását és hogyan lehet az ott tapasztaltakat az LC kijelző alkalmazási feladataihoz illeszteni.

**Kitekintés a tanfolyamunkon túlmutató lehetőségekre**

Ezzel a tanfolyam végére értünk. Megbeszéltük és nyúl-farknyi programokkal megvilágítottuk a 8051-es programozásának alapjait. Végezetül egy sor, a 8051-essel és utódaival való további foglalkozásra ösztönző megjegyzés és gondolat következik.

**Saját projektek**


Saját projektek esetén már azok megvalósítása előtt, a tervezési fázisban minél több gondolkodás szükséges. Amennyiben ez nem történik meg, úgy a későbbi programozás során kellemetlen meglepetésekre lehet számítani. Előre tisztázandó fontos kérdések például a következők. A projekt mely részei kerülnek hardver- és melyek szoftver-megvalósításra?

Hogyan történjenek a különböző szükséges hardver-bővítések csatlakoztatása (portok vagy busz)? Szükséges-e az adatok telepes pufferelése? Elég-e a processzor teljesítménye ahhoz, hogy a tervezett feladattal megbirkózzék, kell-e esetleg az eseményekre interruptokkal reagálnia? Milyen egyszerű problémákra osztható a teljes program és a projekt?

Ezeknek a kérdéseknek a megválaszolása gyakran olyan tapasztalatot igényel, amelyet csak sok-sok gyakorlással lehet megszerezni. Ezért először az egészen egyszerű problémákkal célszerű a programozási tevékenységet elkezdni. Gyakran fordul elő, hogy először egyszerűnek látszó feladat később nehezebb bizonyul. Ezért kezdetben gyakran túlterheljük magunkat. Az első saját lépésekre az olyan egyszerű készülékek látszanak alkalmasnak, melyek néhány TTL IC-vel épülnek fel. Ilyenek a dobókocka, a digitális óra, a morzejeladó stb. Ezek funkcióinak szoftver úton történő megvalósítása gyakorlás és tapasztalat szerzése céljából kiváló lehetőség.

**Billentyűzet-interfészek**

Az adatbevitel a legtöbb esetben saját projekteknél sem kerülhető el. A problémákör



**1067 Budapest, Szondi u. 5-7.**  
(Béke Szálló oldalával szemben)  
**Nyitva: hétfő-péntek 10-18-ig.**  
**Ebédidő: 13-14-ig**  
**Telefon/Fax: 132-7480**

**A legnagyobb HIFI-videó választék Magyarországon!**  
Quad, Mission, Cyrus, Carver, Bang & Olufsen, Denon, Nakamichi, Quadral, Sony, Panasonic stb.  
Hangfalbemutató minden nap előzetes bejelentkezésre.  
Megrendeléseket elfogadunk bármely HAMA fotó-videó-hifi termékre.  
Megnyílt videó-utómunkálatti stúdióink. Kedvező áron bérelhető VHS, S-VHS, Video 8, Hi 8 anyagok vágására, keverésére.  
Megérkezett a Denon DTU-2000 DSR (digitális műholdas rádióadás) vevő. Ára: 79 000 Ft.

rövid áttekintése céljából beszéljük most meg a billentyűk állapota (megnyomott ill. meg nem nyomott) lekérdezésének 6 lehetőségét. Ennek során látni fogjuk, hogy a kapcsolások hardver és szoftver ráfordítás szempontjából különbözőek, és minden esetben vannak előnyeik és hátrányaik is.

#### a) Közvetlen portcsatlakozás

Az 5a. ábrán bemutatott csatlakozási lehetőség bizonyosan a legegyszerűbb. Minden egyes lenyomott billentyű a testre köt egy portvezeték. Mivel a 8051-ben a felhúzóellenállások már beintegrálva megtalálhatók, ezeket kívülről nem kell alkalmazni. Az egyes

kapcsolók állapotának lekérdezésére szolgáló szoftver is érthetően egyszerű, ha bitvizsgáló utasítást (JB-t vagy JNB-t) használunk. A kapcsolat hátrányát a szükséges portvezetékek nagy száma jelenti: az ASCII billentyűzet mintegy 60 billentyűjének bekötéséhez ugyanennyi portvezeték kell. Ebből a szempontból a b változat máris kedvezőbbnek látszik.

#### b) Matrix-portcsatlakozás

Ennél a csatlakozási módatnál (5b. ábra) a billentyűk az oszlop- és sorvezetékek által alkotott mátrix egyes pozícióiban helyezkednek el. Egy kapcsoló állása úgy kérdezhető le, hogy a sorok vezetékeire egymás után alacsony

szintű jelet adunk, miközben a többi sor magas szinten van. A megnyomott billentyű pozíciója ezután az oszlopok lekérdezése útján határozható meg. Ennek a kapcsolásnak az az előnye, hogy sok billentyű kérdezhető le viszonylag kevés vezeték útján, hiszen pl.  $8 \times 8 = 64$  billentyűhöz már 16 vezeték elegendő. Az egyszerű csatlakozás fő hátránya az, hogy több billentyűt egyidejűleg soha nem szabad megnyomni. Ez a hátrány azonban leválasztó diódák formájában némi hardverkiegészítés útján kiküszöbölhető. Megjegyzendő, hogy az 5b. változathoz szükséges szoftver kérés nélkül meglehetősen bonyolult.

#### c) Léptetőregiszteres csatlakozás

Különösen kevés vezetékkel beéri az 5c. ábra szerinti csatlakozás. A billentyűk állapota egy töltőimpulzus útján kerül a léptetőregiszterbe betöltés

#### 4. ábra.

fuer hoehwertige Adresse: magasabb helyiértékű címekhez  
MSB der LCD Adresse C001H bzw C009H: a C001H, illetve C009H LCD címek MSB-je  
LCD niederwertige Byte der Adresse mit RS=0: LCD cím alacsonyabb helyiértékű bájtja RS=0 esetén  
LCD niederwertige Byte der Adresse mit RS=1: LCD cím alacsonyabb helyiértékű bájtja RS=1 esetén  
initialisiere LCD: az LCD inicializálása  
80 Zeichen: 80 karakter  
mit A beginnend: A-val kezdődő  
auf LCD ausgeben: kiadása az LCD-re  
naechstes Zeichen: következő jel  
wiederholen: ismétlés  
Warten: várakozás  
Tasten einlesen: billentyűk beolvasása  
Bits 7 und 6 testen: 7-es és 6-os bitek tesztelése  
nichts zu tun: semmit sem kell tenni  
shift Display S/C=1 R/L=0: kijelző léptetés (S/C=1, R/L=0)  
Warte ein Weilchen: rövid várakozás  
ca 65500\*4 Mikrosekunden: kb. 65500\*4 mikrosekondum  
LCD Treiber Routinen: LCD meghajtórutinok  
Beendigung des letzten Kommandos abwarten: az utolsó parancs befejezésének kivárása  
eine Zeile: egy sor  
Als Kommand: parancsként  
Display an, Cursor, Blinken: kijelző be, kurzor, villogás  
Increment mit Display shift: inkrementálás kijelzőléptetés-sel  
Reset Display: kijelző nullázása  
Warte bis LCD fertig: várakozás, míg az LCD kész  
Bit 7 ist Busy-Flag: 7-es bit Busy-Flag  
Kommando an LCD ausgeben: parancs kiadása az LCD-re  
RS=low <=> Kommando: RS=alacsony <=> parancs  
Ausgabe auf Adresse p2, R0: outputja a p2, R0 címre  
Warten: várakozás  
Zeichen via LCD ausgeben: karakter kiadása az LCD-re  
hoehwertige Adresse: magasabb helyiértékű cím  
RS=high <=> Daten: RS=magas <=> adatok  
ausgeben auf Adresse p2, R0: outputja a p2, R0 címre  
100 Mikrosekunden: 100 mikrosekondum  
Warten: várakozás  
Lese Zustand von LCD: LCD állapot kiolvasása  
RS=low: RS=alacsony  
holen: átadás

#### 4. ábra Az LC kijelző program listája

```

4
***** LISTING of EASM51 (BSP12) *****
LINE LOC OBJ T SOURCE
1 0000 ; ***** DATEI BSP12.A51 *****
2 0000 P2 EQU OAOH ; fuer hoehwertige Adresse
3 0000 ACC EQU OEOH
4 0000 ADDRhigh EQU OCOH ; MSB der LCD Adresse C001H bzw C009H
5 0000 adrRS0 EQU 001H ; LCD niederwertiges Byte der Adresse mit RS=0
6 0000 adrRS1 EQU 009H ; LCD niederwertiges Byte der Adresse mit RS=1
7 0000 ;
8 0000 ; ORG 4100H
9 4100 31 2F [2] START ACALL LCDSET ; initialisiere LCD
10 4102 7B 50 [1] MOV R3,#80 ; 80 Zeichen
11 4104 7A 41 [1] MOV R2,#A' ; mit A beginnend
12 4106 EA [1] SETL MOV A,R2
13 4107 31 4F [2] ACALL LCDCHR ; auf LCD ausgeben
14 4109 0A [1] INC R2 ; naechstes Zeichen
15 410A DB FA [2] DJNZ R3,SETL ; wiederholen
16 410C 31 24 [2] NEU ACALL WAIT ; Warten
17 410E 90 C0 00 [2] MOV DPTR,#OC000H ; Tasten einlesen
18 4111 E0 [2] MOVX A,@DPTR
19 4112 30 E7 05 [2] JNB ACC.7,RECHTS ; Bits 7 und 6 testen
20 4115 30 E6 08 [2] JNB ACC.6,LINKS ; Bits 7 und 6 testen
21 4118 80 F2 [2] SJMP NEU ; nichts zu tun
22 411A 74 18 [1] RECHTS MOV A,#00011000B ; shift Display S/C=1 R/L=0
23 411C 31 47 [2] RAUS ACALL LCDCOM ; als LCD Kommando schicken
24 411E 80 EC [2] SJMP NEU
25 4120 74 1C [1] LINKS MOV A,#00011100B ; shift Display S/C=1 R/L=1
26 4122 80 F8 [2] SJMP RAUS
27 4124 ;
28 4124 78 FF [1] WAIT MOV R0,#255 ; Warte ein Weilchen
29 4126 79 FF [1] WAIT1 MOV R1,#255
30 4128 00 [1] WAIT2 NOP ; 255*4 Mikrosec
31 4129 00 [1] NOP
32 412A D9 FC [2] DJNZ R1,WAIT2
33 412C D8 F8 [2] DJNZ R0,WAIT1 ; * 255
34 412E 22 [2] RET ; ca 65500*4 Mikrosekunden
35 412F ; ; LCD Treiber Routinen
36 412F 31 41 [2] LCDSET ACALL LCDRDY ; Beendigung des letzten Kommandos abwarten
37 4131 74 30 [1] MOV A,#00110000B ; DL=1 N=0 F=0 : 8Bit , eine Zeile , 5*7
Dots
38 4133 31 47 [2] ACALL LCDCOM ; Als Kommando
39 4135 74 0C [1] MOV A,#00001100B ; D=1 C=0 B=0 ,Display an, Cursor, Blinken
aus
40 4137 31 47 [2] ACALL LCDCOM
41 4139 74 06 [1] MOV A,#00000110B ; I/D=1 S=0 : Increment mit Display shift
42 413B 31 47 [2] ACALL LCDCOM
43 413D 74 01 [1] MOV A,#00000001B ; Reset Display
44 413F 31 47 [2] ACALL LCDCOM
45 4141 31 5A [2] LCDRDY ACALL RCOM ; Warte bis LCD fertig
46 4143 20 E7 FB [2] JB ACC.7,LCDRDY ; Bit 7 ist BUSY-Flag
47 4146 22 [2] RET
48 4147 ;
49 4147 75 A0 C0 [2] LCDCOM MOV P2,#ADDRhigh ; Kommando an LCD ausgeben
50 414A 78 01 [1] MOV R0,#adrRS0 ; RS=low <=> Kommando
51 414C F2 [2] MOVX @R0,A ; Ausgabe auf Adresse P2,R0
52 414D 80 06 [2] SJMP WT1 ; Warten
53 414F ;
54 414F LCDCHR EQU $ ; Zeichen via LCD ausgeben
55 414F 75 A0 C0 [2] MOV P2,#ADDRhigh ; hoehwertige Adresse
56 4152 78 09 [1] MOV R0,#adrRS1 ; RS=high <=> Daten
57 4154 F2 [2] MOVX @R0,A ; ausgeben auf Adresse P2,R0
58 4155 78 32 [1] WT1 MOV R0,#50 ; 100 Mikrosekunden
59 4157 D8 FE [2] WT2 DJNZ R0,WT2 ; warten
60 4159 22 [2] RET
61 415A ;
62 415A 75 A0 C0 [2] RCOM MOV P2,#ADDRhigh ; Lese Zustand von LCD
63 415D 78 01 [1] MOV R0,#adrRS0 ; RS=low
64 415F E2 [2] MOVX A,@R0 ; holen
65 4160 22 [2] RET
66 4161 END
***** SYMBOLTABLE (21 symbols) *****
P2 :00A0 ACC :00E0 ADDRhigh :00C0 adrRS0 :0001
adrRS1 :0009 START :4100 SETL :4106 NEU :410C
RECHTS :411A RAUS :411C LINKS :4120 WAIT :4124
WAIT1 :4126 WAIT2 :4128 LCDSET :412F LCDRDY :4141
LCDCOM :4147 LCDCHR :414F WT1 :4155 WT2 :4157
RCOM :415A

```

910109 - 8 - 14

tésre és P1.2-n keresztül 8 óra impulzus alatt toódik be a 8051-be. Ehhez mindössze három portvezeték szükséges és a szoftver is egyszerű. További billentyűk rákapcsolása céljából több léptetőregisztert lehet kaszkádba kapcsolni. Ezzel együtt viszont természetesen a kiolvasási idő is megnő.

**d) Multiplexelt csatlakozás**

Az 5d. ábra szerinti változat négy portvezetékét használja. A P1.0...P1.2 portvezetékek egy 8 az 1-hez multiplexer útján választanak ki egyet az S1...S8 billentyűk közül és továbbítják annak jelét a P1.3-ra. A szoftver ennél a kapcsol-

lásnál is egyszerű és több billentyű beiktatása céljából a kapcsolás itt is kaszkád elrendezésben használható.

**e) Buszos csatlakozás**

Az 5e. ábra szerinti kapcsolás használata olyan esetekben célszerű, amikor szabad portvezeték nem áll rendelkezésre, az adatbusz azonban használható. A használt cím kiválasztása címdekód útján történik és a billentyű jele a processzor Read vezetékeivel kombinálva jut a Tri-State-Puffer útján a buszra. Az ehhez szükséges szoftver igen egyszerű, a hardver azonban meglehetősen sok huzalozással jár.

**f) Multivibrátoros interfész**

Az 5f. ábra kapcsolási változatához mindössze egy portvezeték szükséges. A két invertekből felépített oscillátor frekvenciája minden billentyű megnyomásakor valamilyen mértékben megváltozik. A program a P1.0 ponton megjelenő négyzögjel frekvenciáját méri és ebből határozza meg, hogy melyik gomb van megnyomva. A szükséges portvezetékek száma tekintetében e változat előnye aligha vitatható, a szoftver viszont ennek megfelelően terjedelmes. A kapcsolás lényegesen több billentyűre sem nagyon bővíthető, mert akkor a frekvenciák eltérései már túl kicsivé válnának.

Az előbbieken vázolt példák világosan megmutatják, hogy egy látszólag egyszerű feladat is milyen sokféleképpen oldható meg. A kreativitás ezen a téren szinte határtalanul kibontakozhat. Egyes esetekben érdekes kihívásnak számít egy adott feladat minimális ráfordítással történő megoldása akkor, ha például már csak két szabad portvezeték áll rendelkezésre és az adatbuszt nem célszerű használni, mert az a panel huzalozását túl bonyolulttá tenné. Itt is a gyakorlat teszi a mestert és sok könnyvben található izgalmas feladatok. Egész sor jó ötlet található a 6502-es és 8080-as időkből származó, kissé régebbi számítógépes irodalomban is. Ezek közül sok a 8051-es projekteknel is kiindulásként szolgálhat.

**Szoftver-  
építészekrény**

Az egyszerű programoknál és projekteknel felgyűlt tapasztalatok kihasználása céljából

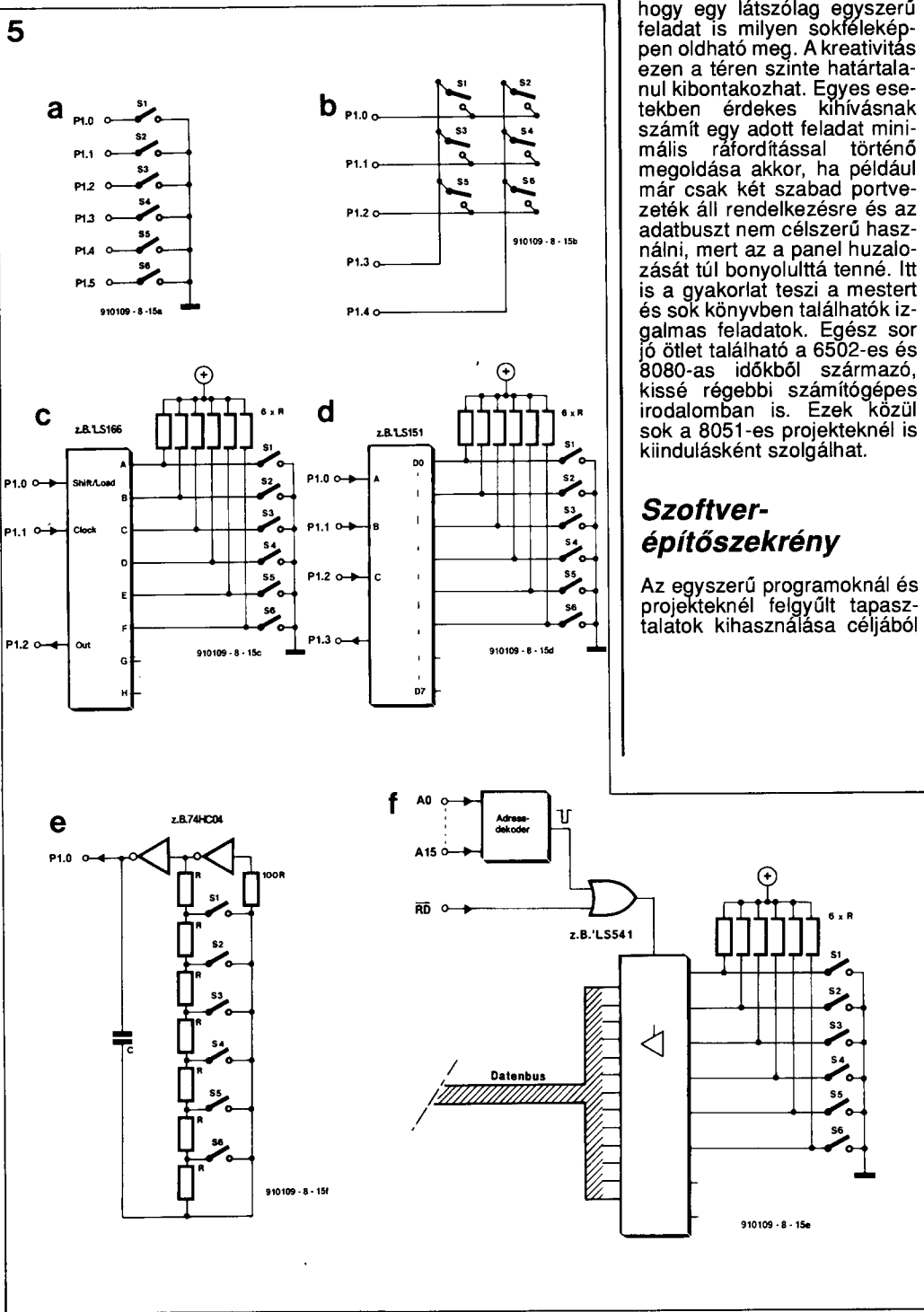
meg kell szokni, hogy a programrészeket úgy programozzuk, hogy azokat később újra fel tudjuk használni. Idővel a hasznos alprogramokból olyan alap halmazódik fel, melyből könnyedén összeállíthatók nagyobb programok is. Az alprogramok ismételt felhasználhatósága természetesen a hozzájuk csatolt dokumentáció részletességétől is függ.

A széleskörűen elterjedt véleménnyel ellentétben az assembler-programozásnál nem valamiféle trükkprogramozásról van szó. Éppen ellenkezőleg. A szerző véleménye szerint assemblerben ugyanolyan strukturáltan és rendezetten lehet programozni mint, minden más programnyelven, mert a program felépítése itt is lényegében a programozó gondolkodásmódjának kérdése. Sok ötlet ismerhető meg a mások által készített programokból is. Érdemes ebből a célból az elérhető szakkönyveket is módszeresen átnézni. E különböző ötletforrások ellenére az assemblerben történő programozás önfegyelm szempontjából magas követelményeket támaszt és nem tekinthető könnyű feladatnak. Ez életet némileg megkönnyítik azok a segéd-eszközök, melyekről a következőkben szót ejtünk.

**Makro-assembler**

Tanfolyamunkon assemblálás céljára egy igen egyszerű assemblert használtunk, mely azonban kezdőknek tökéletesen elegendő. A profi felhasználó, akinek több ezer márkát sem számít, természetesen jobb eszközöket fog használni. A piacon kapható assemblerek igen széles teljesítményspektrumot kínálnak. Sok esetben alkalmasak úgynevezett makrók feldolgozására is. Makróknak nevezik a hosszabb, gyakran azonos vagy csaknem azonos formában visszatérő utasítássorozatokra (vagy más szövegekre) vonatkozó rövidített írásmód bevezetésének lehetőségét. Az assembler ilyenkor az adott rövidítést minden esetben a makróhoz rendelt szöveggel helyettesíti. Emellett a lehetőség mellett sok assembler az olyan különböző programrészek külön-külön történő assemblálását is lehetővé teszi, melyek később egy úgynevezett linker útján kerülnek összeépítésre egyetlen programmá. Ennek során még alprogram-könyvtárak is befűzhetők. Nagy programoknál az ilyen eljárás igen hatékony lehet. A hobbista programozónak azonban sok esetben ilyen lehetőségek nélkül is meg kell lennie. Mielőtt tehát az ember 500...1500 márkát investálna egy assemblerbe, gondosan meg kell vizsgálnia, hogy saját igényeinek kielégítése céljából indo-

5. ábra. A billentyűzetek vagy nyomógombok lekérdezésének különböző lehetőségei





kolt-e a beszerzés. A kiemelt árkategóriába tartozó programfejlesztő segédeszközök alternatívájaként az elmúlt időszakban olcsóbb, de makroutasításokra is alkalmas assemblerek is megjelentek a piacon. Erről a szaklapok hirdetéseinek gyors áttekintése útján is könnyen meggyőződhetünk.

Akinek az assemblerben történő programozás túl fáradtságosnak tűnik, az a 8051-esen is használhat magasabb szintű programnyelvet. A 8051-hez például C-Compiler és a Forth-Interpreter is kapható. De ezek sem igazán olcsók.

## 8051-es szimulátorok

Programteszteléshez 8051-es szimulátorprogramok állnak rendelkezésre.

Ezek a 8051-est egy PC-n vagy más számítógépen képezik le. Egy program így akár lépésről lépésre is végrehajtható a „host” számítógépen. Ennek során valamennyi regiszter stb. kijelzésre kerül. A szimulátorok kiválóan alkalmasak egyértelmű szoftverproblémák és hibák felismerésére és elhárítására. Kevésbé hasznosak akkor, ha a probléma abból adódik, hogy a hardver és a szoftver együttműködése nem megfelelő. Mivel a mikrovezérlők üzemeltetése természetesen általában külső hardverbővítésekkel történik, szimulátorok használata is csak feltételes gyógyszernek tekinthető. Mivel azonban a szimulátorokat ma már sokszor igen olcsón kínálják, ezek saját megfontolásaink során mindenesetre számolni kell. A szimulátorok a kezdő programozóknak is hasznos segítséget jelentenek, mert az egyes utasítások hatása a PC képernyőjén pontosabban követhető. Saját készítésű programok tényleges használat során történő tesztelésére azonban inkább az EPROM szimulátor felel meg.

## EPROM szimulátor

A tanfolyam során valamennyi program végrehajtása az EMON51 monitorprogram vezérlése mellett történt és a programok a RAM-ba kerültek betöltésre. Gyakran kerül azonban sor olyan projektek megvalósítására, melyeknél költségkihatása miatt a RAM-ról teljesen lemondanak. Előfordulhat, hogy egy ilyen rendszerben még V24-es interfész sem áll rendelkezésre. A program valóságos feltételek közötti tesztelhetősége ennek ellenére természetesen mégis abszolút követelmény. Ilyen feladatok megoldása során egy EPROM szimulátor nagy segítséget jelent. A program mindenkori aktuális változata ebbe kerül betöltésre és itt történik annak tesztelése.

## In-Circuit emulátorok

Még nehezebb a helyzet akkor, ha a 8051-est egy belső ROM-mal rendelkező projektben kell alkalmazni. Itt a tesztelésnél az EPROM szimulátor már nem segít, hiszen EPROM egyáltalán nincs is jelen. Ahhoz, hogy a programokat mégis a végleges kapcsolatban lehessen tesztelni, az In-Circuit emulátorok állnak rendelkezésre. Ezek a meglehetősen drága „kasznik” a 8051-es CPU viselkedését előre megadható programtartalom mellett szimulálják. Segítségükkel egy sor olyan hardverhiba is megtalálható, mely például időzíti problémára vezethető vissza.

## A 8051-es utódai: 80535-ös és társai

A tanfolyam során a 8051-es sorozat kissé régebbi, olyan modelljeire koncentráltunk, melyek mindamelllett még ma is ipari szabványt jelentenek. Aki azonban a 8051-es sorozat mikrovezérlővel készül projekteket megvalósítani, annak az utódmodelleket (pl. Siemens SAB 80C535-öt, SAB 80C537-et, OKI MSM 80C154-et, Valvo PCB 83C552-t stb.) is szem előtt kell tartania. Ezek lényegében felfelé kompatibilisek, ami azt jelenti, hogy mindazt tudják, amit a 8051-es tud, mégpedig azonos programozás mellett.

Olyan plusz lehetőségekkel is rendelkeznek azonban, mint például további időzítők, integrált A/D-átalakító, integrált Watchdog-Timer a csipen, járulékos portok, gyorsabb aritmetikai egységek stb. A kiegészítő lehetőségek további SFR-eket használnak. Ennek alapján az Elektor tanfolyam assemblerre ezekre a kontroller típusokra is alkalmazható. A Compuboard 80535-ös dugaszolható kártyával való kiépítése már kifejlesztésre került, ezzel kapcsolatban a tanfolyam kiegészítéseként a 80535-öst is részletesen ismertetni fogjuk.

A fenti megjegyzésekkel valóban a tanfolyam végére értünk ugyan, a 8051 alapjaival azonban a témát messzemenően nem merítettük ki. A 80535-re történő bővítést már említettük. Ahogy az már az Elektorhoz illik, további igen érdekes mikrovezérlő-alkalmazásokkal fogunk a jövőben is jelentkezni (többek között a 80535-ösre épülő projektekkel is). A tanfolyamon szerzett ismeretek az Elektor számos olvasójánál bizonyára saját alkalmazások kidolgozásához is vezetnek majd. Az assembler, a Compuboard bővítés és az EPROM szimulátor ugyanis olyan komplett fejlesztési környezetet képez, melyben mindenki számára lehetőség nyílik a 8051-es mikrovezérlőre alapuló saját projektek megvalósítására. ■

# GoldStar

magyarországi képviselő:

**RAPAS Kft.**

1191 Budapest, Üllői út 200. Tel./Fax: 127-0863

## OSZCILLOSKÓPOK

20/40/60/100 MHz-es sávzélesség, 1 mV érzékenység  
Digitális kijelzés, auto-fókusz, tv-szinkron, 8 x 10 cm-es képernyő, késleltetett időzítés, beépített funkciógenerátor

### CSAK NOVEMBER VÉGÉIG!!!

**OS 9020 analóg oszcilloszkóp 45 000 Ft (+ ÁFA!)**  
2 csatorna, 20 MHz-es sávzélesség, auto-fókusz,  
tv-szinkron, 8 x 10 cm képernyő, 5 mV osztás érzékenység

## DIGITÁLIS MULTIMÉTEREK

3<sup>1</sup>/<sub>2</sub>, 4<sup>1</sup>/<sub>2</sub> digités kijelzés, analóg oszlopdiagramos kijelzés.

Az alapméréseken kívül kapacitás-, frekvencia-, hőmérséklet-, folytonosság-, félvezetőmérések. Automatikus és kézi méréshatárváltás, hibás csatlakozás esetén hangjelzés. Elektronikus védelem minden méréshatárban.

### CSAK NOVEMBER VÉGÉIG!!!

**DM-9185 digitális multiméter: 8700 Ft + ÁFA**  
AC/DC feszültség- és árammérés, ellenállás, kapacitás-,  
frekvencia-, hőmérséklet-, folytonosság-, félvezetőmérés.  
Egyéb funkciók: valódi RMS mérés, Max, Min és  
átlagértékmérés, 24 órás mérésadatgyűjtés,  
relatívérték-mérés, %-os mérés és kijelzés, lóg,  
nullgalvanométernek is használható oszlopdiagramos  
kijelző, Jó/Nem jó válogatási lehetőség.  
Gumirozott rázásálló tokozás.

**Árusítás: a fenti címen vagy CALDERONI bolt,  
VII., István u. 47.**

## „PUSKÁS TIVADAR” MŰSZER- ÉS GÉPIPARI SZÖVETKEZET

1138 Budapest, Topolya u. 4-8.

Tel.: 129-6250

Fax: 140-2942

Telex: 22-6428

- Környezetvédelmi műszerek  
vízminőségmérés: zavarosság, pH, oldott oxigén, vezetőképesség, hőmérséklet
- Ipari műszerek  
folyadék szintmérés és -szabályozás
- Transzformátorok 100 VA-ig
- Forrasztástechnikai eszközök  
pákák, forrasztóállomások
- Tömegelemek  
kábelsaruk, érvéghüvelyek, biztosítéktartók, távtartók
- Híradástechnikai berendezések  
adatok és információ átvitele erősáramú kábeleken és légvezetéseken
- Alkonyatkapcsolók, elektromos ajtózárok,  
időkapcsoló relék, lépcsőházi automaták,  
műszerdobozok fémből és műanyagból