

FlexProg, a tanítható programozó

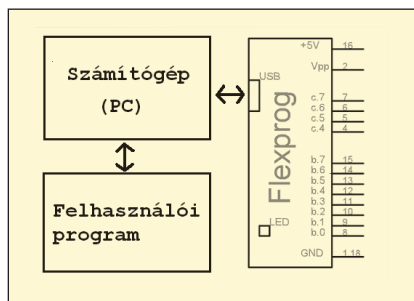
Vörös Tamás tanár, adlerson@freemail.hu

A különféle memória IC-eket, EPROM-okat, mikrovezérlőket programozók számára kifejezetten nagy kihívást jelent egy új típus megjelenése, vagy éppen egy rég elfeledett típus előkerülése. Ilyenkor mindig az okozza a fő problémát, hogy a megszokott, bevált „égetőprogram” nem ismeri az adott IC-t. Itt általában kétféle lehetőség közül választhatunk: vagy előbb-utóbb találunk egy olyan égetőprogramot, amely az adott típust is támogatja, vagy nem. Ekkor azonban jobb esetben magunknak kell megírni a programozórutint, ami általában az adattábla kezeléssel együtt már szép feladatnak mondható. És ekkor még nem beszélünk a hardverről...

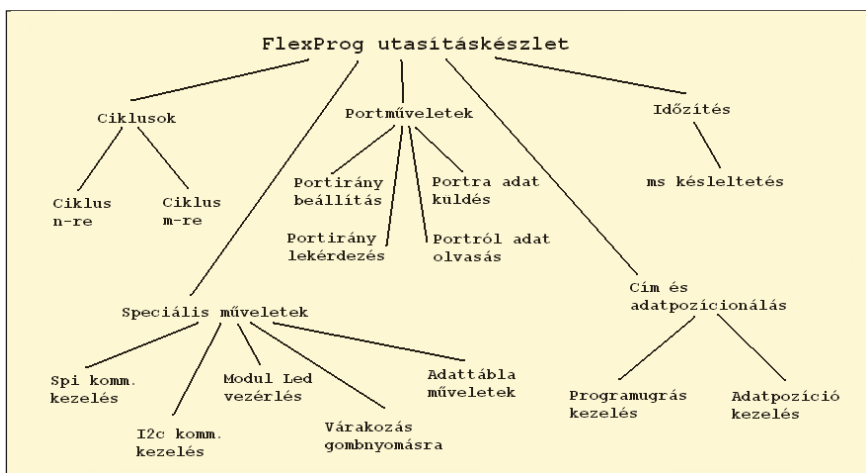
A következőkben egy olyan szoftver-hardver együtttest mutatunk be, amely az előbbieken vázolt problémákra egyszerűen megoldást kínál, nem utolsósorban szinte játékosan elsajátíthatók vele a programozás alapfogalmai. A rendszer alapötletét egy rég elfeledett 82S129 típusú PROM áramkör adta, amellyel egy régi pénzvizsgáló programozása közben sikerült összeakadni. A régi memóriatartalom ugyanis nem tartalmazta a 200 forintos felismeréséhez szükséges adatokat, egy pénzdefiníció azonban még éppen belefért volna. A PROM maradék adatterületének égetéséhez gyakorlatilag meg kellett konstruálni egy célhardvert, és megírni egy tetszetős és felhasználóbarát, Windows alatt futó programozó-szoftvert. A PC és a hardver között „természetesen” csak az USB kapcsolat jöhetett szóba, mivel a megrendelő a PROM-okat egy új notebook segítségével szeretne volna programozni.

A programozó tömbvázlata az 1. ábrán látható. A PC segítségével tartja a kapcsolatot egymással az USB csatlakozóval rendelkező modul és a felhasználói program. Lássuk először az ábra jobb oldalán található modul jellemzőit. A hardver lelke természetesen egy mikrovezérlő, amely a felhasználói programtól kapott utasításokat hajtja végre. A modul csatlakozója egy 18 pólusú hüvelyszív, amelybe tűskesort dugaszolva a programozandó áramkörrel a kapcsolat létre jön. Az USB port +5 V-os feszültsége táplálja a modult, de a csatlakozósorra is ki lett vezetve. Legfeljebb 100-200 mA-rel terhelve gyakorlatilag bármilyen memória vagy programozandó IC tápellátását biztosítani tudja. A port a számítógépek nagy részénél rövidzárvédett, így a gépben nem teszünk kárt egy esetleges malőr alkalmával, legfeljebb egy újraindulást vagyunk kénytelenek megvárni (tapasztalat...). A modul 13 db felhasználói kivezetéssel rendelkezik. Ebből 12 lábat tetszőlegesen használhatunk ki-, illetve bemenetként, ráadásul portként is kezelhetjük azokat. A „b”-vel jelölt porton 8 kivezetés, míg a „c” porton 4 kivezetéssel találkozhatunk. Ez így önmagában még nem nagy dolog, a 13. láb – ami a modul 2. számú kivezetése – teszi azonban igazi programozóvá a modult. A panelen helyet kapott egy kb. 20 mA-rel terhelhető +12 V-ot adó feszültségforrás is. Erre a bizonyos 13. lábra szoftveresen kapcsolhatunk +12 V-ot, +5 V-ot, illetve GND-t. Ezzel az EPROM-ok, flash memóriák, mikrovezérlők, azaz a nagyobb programozási feszültséget, ún. Vpp-t igénylő

áramkörök kezelése is lehetséges. Nevezhetjük ezt a lábat háromállapotúnak, igaz nem a klasszikus értelemben. A nagy kimenőfeszültség ára, hogy ezen kivezetés csupán kimenetként vezérelhető, adatot fogadni nem tud. A modul a háromállapotú láb kivételével a megszokott +5 V-os logikát követi a szokásos jelszintekkel. A használat során ügyeljünk arra, hogy 5,6 V-nál nagyobb feszültséget a bemenetek ne kaphassanak, mert ekkor a mikrovezérlő védelmei aktiválódnak. Egy esetleges GND-vel történő kimeneti rövidzár esetében néhányszor 10 mA áramot tapasztalhatunk a védelmeknek köszönhetően. A panelen lévő IC-kben lévő védelmek hosszú ideig fennálló hazardok esetében persze nem nyújtanak biztos védelmet, de a rövid idejű túlterhelésekkel szemben ellenállóak. A kimenetek mindegyike hozzávetőlegesen 20 mA-es árammal terhelhető, tehát egy 220-240 ohmos terhelést hosszútávon meg tud hajtani. A modulon helyet kapott egy állapotjelző LED, amely bekapcsoláskor három felvillanással jelzi, hogy a hardver használatra kész, ezenkívül minden PC-től fogadott utasítást rövid felvillanással jelez. A rendszer sebességére jellemző, hogy egyetlen másodperc alatt többszáz utasítást képes végrehajtani. A tipikus utasításideje 4 ms körül van. Egyszerű portvezérlés esetében természetesen kevesebb, míg spi vagy i2c kommunikáció esetében 6-7 ms-ra is felmehet.



1. ábra



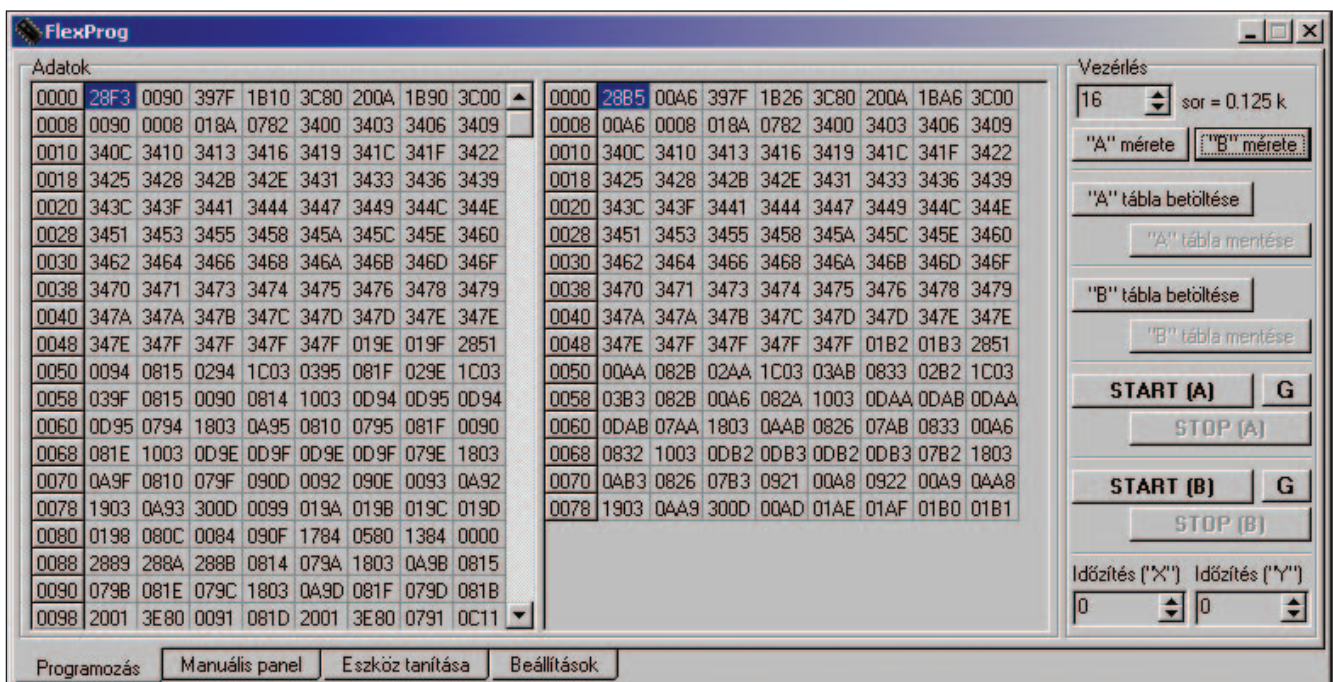
2. ábra

A rendszer működésének alapelve a következő: a FlexProg moduljának utasításkészletéből a felhasználó építi fel az aktuálisan kezelendő IC által igényelt utasítássorozatot. Ezt a tulajdonképpeni programvezérlést egy fájl formájában akár el is lehet menteni a számítógépre „pcf” kiterjesztéssel. Ezt a fájlt a későbbiekben megnyitva újra elvégezhetjük az adott alkatrészben a kívánt műveletet. Nézzük meg ezt egy példán keresztül: tegyük fel, hogy egy 27C128-as EPROM-ot szeretnénk felprogramozni. Az EPROM adatlapjában fellelhető a pontos progra-

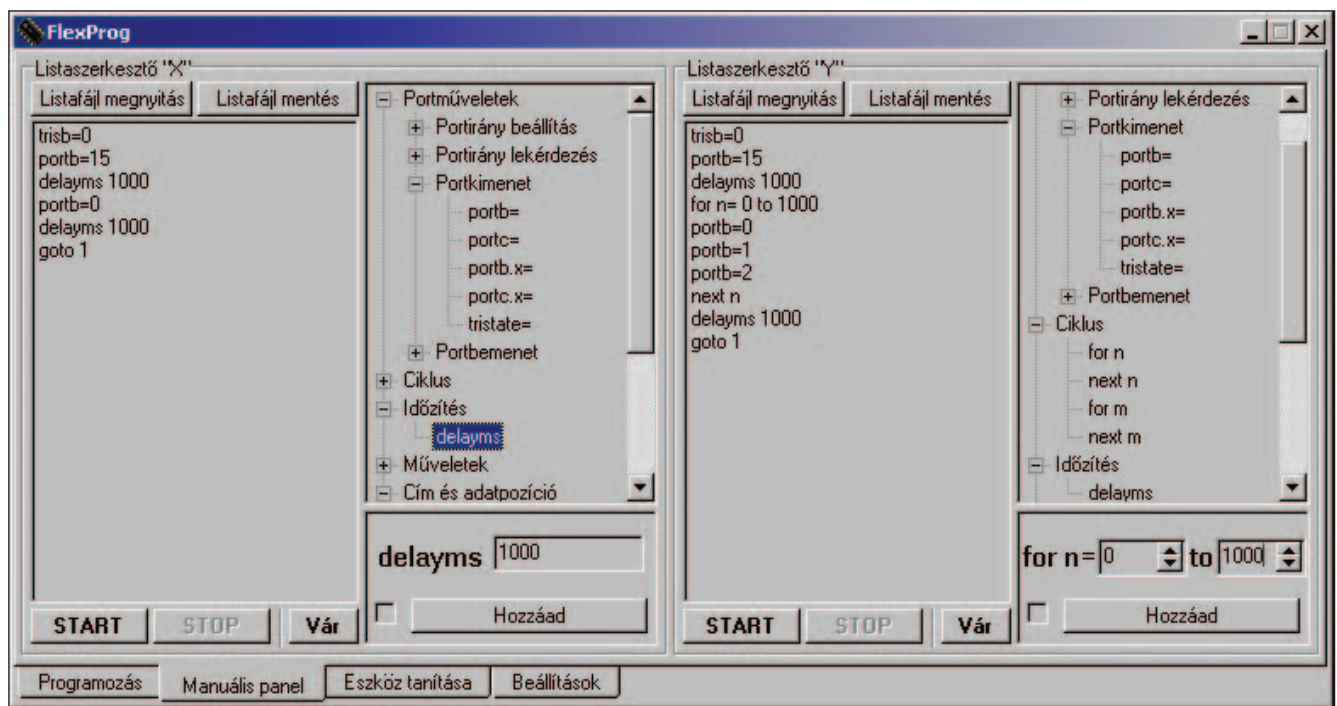
mozási procedúra folyamata, így a FlexProg modul kivezetéseit a megfelelő sorrendben vezérelve egy adattábla tartalma az eszközbe tölthető. Az elkészített és tesztelt utasítássorozatot elmentjük egy fájlba. Amikor legközelebb egy ilyen 27C128-ast kellene felprogramozni, csak megnyitjuk az utasításfájlt, majd betöltjük a felhasználói programba a beégetendő adatfájlt, a programozási folyamatot elindítjuk, és az adat betöltődik az EPROM-ba. Hasonlóképpen tehetünk az adattartalom kiolvasásánál is. Mivel a különféle eszközök programozási művele-

te eléggé eltérő lehet, az éppen használt eszközhöz használatos „pcf” fájl megnyitása után már programozható is az adott IC. Ezzel a módszerrel csaknem bármely manapság előforduló programozható eszköz, memória, mikrovezérlő kezelése megoldható.

A 2. ábrán a FlexProg utasításkészlete látható. Az utasítássorozat elkészítése mindösszesen öt-féle típusra vezethető vissza. A legfontosabb a „portműveletek” csoportja, mivel ezek segítségével kezelhetjük a FlexProg modul kivezetéseit. A portműveleteket négy alcsoportba sorolhatjuk: az egyes kivezetések adatáramlási irányát felhasználás előtt be kell állítani. A háromlappalós láb kivételével a kivezetések akár egyesével is konfigurálhatók bemenetnek vagy kimenetnek. A portirány természetesen lekérdezhető is. Az irány beállítása után akár már adatot is küldhetünk a kiválasztott portra vagy kivezetésre. Az utasítások többszöri végrehajtását célszerű ciklussal megoldani. Például egy 2 kilobájt adatterülettel rendelkező bájt szervezésű memória kiolvasását érdemes egy 0-tól 2047-ig futó ciklussal megoldani, miközben a ciklusváltozó pozíció-



3. ábra



4. ábra

nálja az adattáblában a kiolvasott információt. A kétdimenziós adatelérés biztosítása érdekében (sor-oszlop szerveződés) két különböző ciklust is egymásba ágyazhatunk, egyiket „n”, a másikat pedig „m” ciklusváltozóval léptetve. Ciklusok felhasználása nélkül is létrehozhatjuk az utasítássorozat ciklikus működését, méghozzá a programugrások használatával. Az éppen aktuális utasítás után egy programugrással a program bármely másik sorban folytatható. Érdekes azonban a programugrásokkal vigyázni, mert könnyen végtelenített programfutás érhető el. Az adattáblában éppen érvényes pozíciót is módosíthatjuk, azaz bárhol olvashatunk és bárhová írhatunk a táblán belül, nem szükséges az olvasott vagy írandó adatokat sorban elhelyezni. Ez praktikus az „n” és „m” adatpozíciót meghatározó változók értékének szinte tetszőleges megadását jelenti. Ahhoz, hogy a program futását láthatóvá tehesük, le kell azt lassítani. Ezt tehetjük meg a késleltető utasítás kiadásával. Ez nem okoz a modul és a felhasználói program között adatforgalmat, az időzítést teljes egészében a PC fel-

ügyleti. Akkor is szükség lehet a késleltetésre, ha az adott feladathoz a FlexProg sebessége túl nagy a bizonyul. 20-30 ms-os késleltetésekkel az egyes utasítások között a program futása szépen lassítható.

Aki az eddigieket áttanulmányozta, annak úgy tűnhet, hogy a FlexProg csupán a párhuzamos elérésű eszközöket támogatja. A kis és közepes sebességű adatátvitelben a soros elérésű eszközök szinte egyeduralmuk a kevés lábszám és az egyszerű áramköri környezet igénye miatt. Gondoljunk csak az I2C, vagy a SPI EEPROM-okra, vagy akár az SD kártyára, de ide sorolhatjuk a mikrovezérlők jó részét, többek között a PIC-eket is. A FlexProg utasításkészlete tartalmaz összetettebb parancsokat is, mint például az i2c, illetve a spi buszra írás vagy arról olvasás. Ezekkel a magas szintű utasításokkal már nagy sebességű PIC programozót is készíthetünk mindösszesen két darab külső ellenállás felhasználásával (ne feledjük, a +12 V elérése adott a modulon!). A speciális műveletek között megtaláljuk még a panelen lévő LED be- és kikapcsolásához szükséges utasításokat

is. Ezek általában összetettebb programok esetében használhatók inkább, visszajelzési céllal. A program futását akár szakaszokra is bonthatjuk, a továbbhaladást pedig a felhasználói program egy gombjára kattintáshoz köthetjük. Ezt a funkciót is csak a PC felülegyleti, adatforgalmat nem generál a modul és a számítógép között.

A továbbiakban lássuk a PC-n futó Windows-alapú felhasználói programot. A program optimális futtatásához szükséges egy jó P4-es konfiguráció legalább 1,5 GHz-es processzorral, 256 Mbyte RAM-mal. A szoftver tesztelése XP-n, illetve Win2k-n történt. A feltelepített FlexProg.exe-t elindítva a 3. ábrához hasonló képernyőképet kapunk, természetesen üres adattáblákkal. A program kezelőfelülete nagyon egyszerű, használatát könnyű elsajátítani. Az egyetlen ablak bal alsó részén látható fülek segítségével navigálhatunk az egyes lapok között. A Programozás fül tartalmazza az adattáblákat, a Manuális panelen az utasítássorozatot építhetjük fel az egyszerű parancsokból, az Eszköz tanítása fülön pedig a modul egyes kivezetéseinek viselkedését állíthatjuk be a progra-