

# FlexProg, a tanítható programozó 2. rész

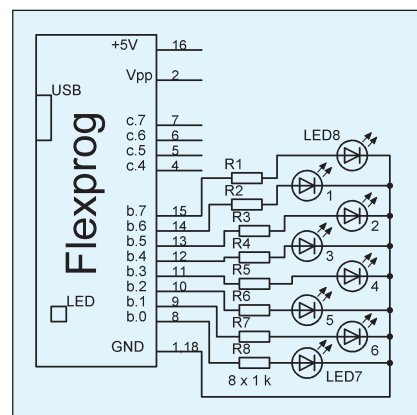
Vörös Tamás tanár, adlerson@freemail.hu

Az 5. ábrán végül elérkeztünk talán a legfontosabb részhez, az egyéni függvény készítésének lehetőségéhez. Ezen a lapon komplett alkatrészfájlt definiálhatunk saját függvényeivel. Létrehozhatunk például egy 27c128.adf fájlt, amelyben külön függvényként elkészítjük az olvasás és az írás folyamatát. Így már csak az elkészített függvény nevét kell az utasítás-sorozatban szerepeltetni, ami ezáltal jelentősen leegyszerűsödik. A függvényszerkesztő bal oldalán a 13 kivezetés egyéni neveit adhatjuk meg, a táblában pedig az adatlapból kiolvasható jelalakokat „szerkeszthetjük meg”.

## Gyakorlati példák

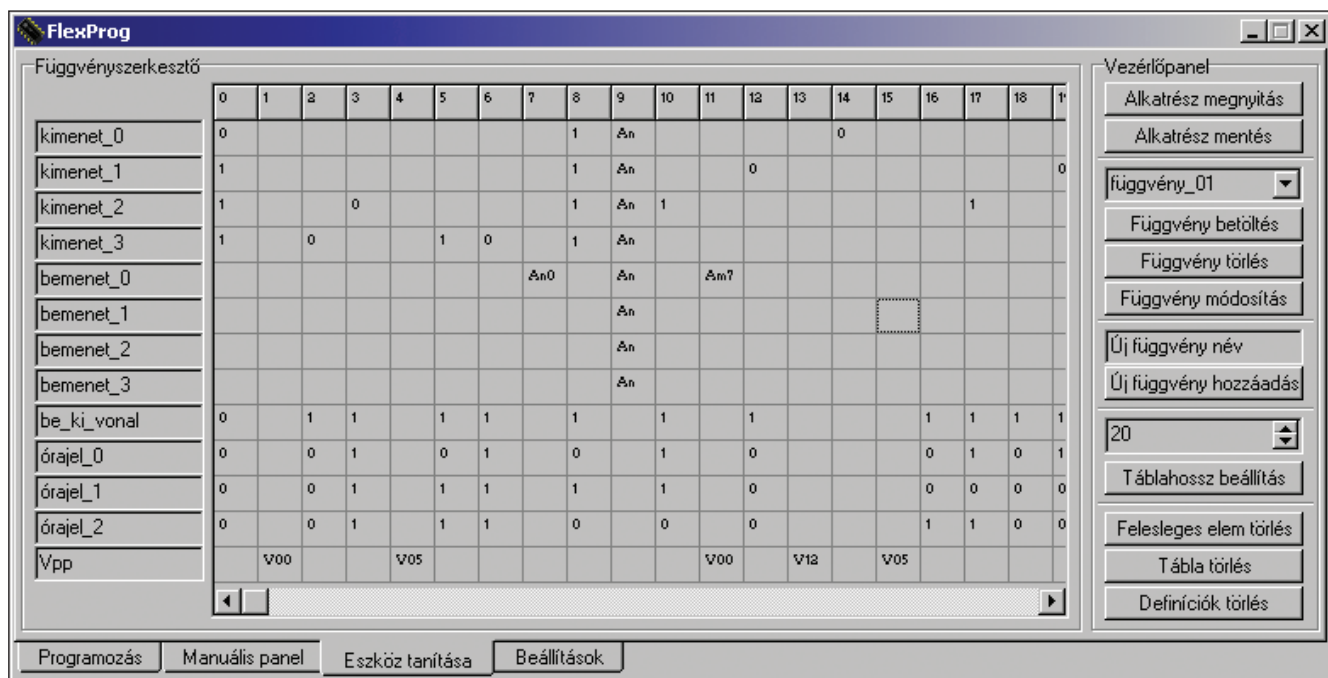
A következőkben két példán keresztül mutatjuk be a FlexProg alkalmazását. Az egyik egy LED-sor vezérlése, a másik pedig egy 27C128 EPROM olvasási művelete. Mindkét esetben minimális mennyiségű külső alkatrész felhasználásával érünk célba. A 6. ábrán nyolc LED-et kapcsolunk 1 kohmos előtétellenálláson kereszt-

tül a FlexProg modul „b” portjára. A LED-ek praktikusán piros, zöld vagy sárga színűek legyenek, mert ezek nyitófeszültsége 1,5-2 V körüli. Az előtéték használatával az egyes portlábakon 3-4 mA erősségű áram fog folyni, bekapcsolt állapotban. A LED-eket működtető programot kétféle változatban készítettük el. A 7. ábra bal oldalán lévő programban nem használunk adattáblát. Az első sorban a „b” port adatáramlásának irányát állítjuk be, a teljes portot kimenetnek konfiguráljuk (trisb=0). Ezután elindítunk egy ciklust n-re úgy, hogy n értéke 0-tól 7-ig változzon egyesével növekedve. A ciklus lefutása után a program megáll. A ciklusmagban mindösszesen két utasítás található: az elsőben a „b” portra kiküldjük az éppen aktuális ciklusváltozó értékét, azaz 0...7. Ez értelemszerűen az USB vonalon keresztül el is jut a FlexProg modulhoz, ott pedig kigyújtja a megfelelő LED(ek)-et. Mivel egy ilyen egyszerű utasítást a modul kb. 4 ms alatt elvéggez, a programban szükség van némi késleltetésre is, ha a folyamatot látni is szeretnénk. Er-

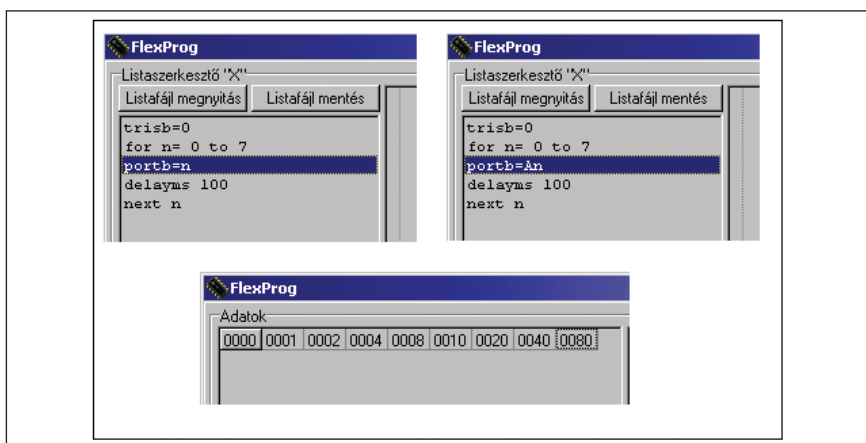


6. ábra

re szolgál a delaysms 100 utasítás, amely 100 ms késleltetést visz a program futásába minden egyes ciklusmag futáskor. Így összesen  $4 + (4 + 100) * 8 = 836$  ms alatt fog lefutni a teljes program. Ha a listafájl mentése gombra kattintunk, akár el is menthetjük a programot pcf kiterjesztéssel. Mivel a fényjárték a programban volt kódolva, ezért módosítása nehézkes, ha például másféle felvillanási sorrendet szeretnénk beállítani. A 7. ábra jobb oldalán található kódban csu-



5. ábra



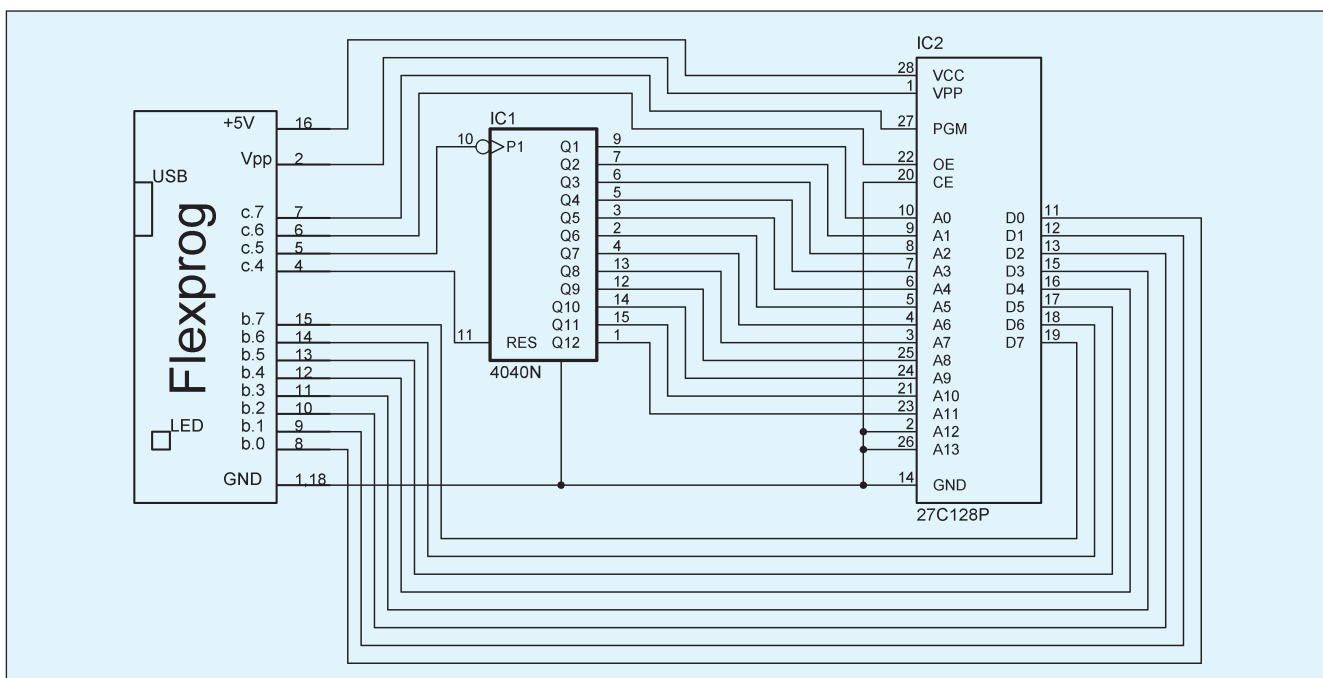
7. ábra

pán egy apró változtatást hajtunk végre. A „b” portra kiküldött adatot immár nem a programból vesszük, hanem kívülről, méghozzá a programozás fülön található baloldali adattáblából. Az „An” első karaktere jelöli a baloldali adattáblát, a második pedig a ciklusváltozóval egyezik meg. Ez esetben az adattáblát fel kell tölteni adatokkal, az ábra alján megadott értékekkel a LED-ek szépen sorban fognak felvillanni kb. 100 ms különbséggel. Az adattábla tartalmát is elmenthetjük bin vagy hex formátumba. Így legközelebb, ha kedvünk tartja megnézni a fényjátékot, mindkét fájlt csak meg kell nyitni a megfelelő helyen és a START gombra kattintva már in-

dul is a program. A LED-ek villogtatása természetesen csak játék, de ezen keresztül érdemes a szoftver és az utasítássorozatok alapműködését megismerni, mert egyszerűen szemmel látható a programfutás végeredménye.

A második példa egy párhuzamos elérésű EPROM kiolvasása. Már az alkatrész adattábláját tanulmányozva felismerhetjük, hogy bizony túl sok lába van az EPROM-nak. Egészen pontosan, ha az E (chip enable) lábat nem vezéreljük, akkor is 25 kivezetésre kell feszültséget kapcsolni. A problémát egy 4040 alkalmazásával küszöböljük ki, ennek vezérlése csupán két lábat „emészt fel” cserébe viszont ellátja a teljes EPROM-címzést. A

kapcsolás a 8. ábrán látható. A FlexProg modul +5 V-os USB tápfeszültsége működteti a 4040-es számlálót is és a 27C128-at is. Az áramkör a számlálón kívül semmilyen egyéb alkatrészt nem igényel. Az EPROM-nak a Vpp lábára kapcsoljuk a modul háromállapotú kimenetét, az adatbuszt pedig a „b” portra. A 4040-es léptetését és az EPROM vezérlését a modul „c” portja látja el. A 9. ábra bal oldalon látható az adattábla olvasásra vonatkozó jelalakja. Látszik, hogy először a címzést kell beállítani, majd a E és az G lábakat kell GND-re húzni. Ezután rövid idővel megjelenik a kimeneten a beállított címen található adatbájtt. Nagyon fontos, hogy a Vpp lábán olvasás közben +5 V, írás közben pedig +12 V legyen. Lássuk a programot! Az első sor bemenetnek konfigurálja a „b” portot. Ez nagyon fontos, mert kimeneti állapotban hartzard keletkezne az EPROM kimenetével. A második sor a vezérlőportot kimenetnek állítja be. A harmadik sorban a háromállapotú kimenetre +5 V-ot kapcsolunk. A 4-5-6-7 sorok logikai magas szintre állítják a vezérlőportot. A 8-9 sor reseteli a 4040-est. A művelethez a tok egy GND-re húzást igényel, így minden kimenete GND-n lesz. Az ezután következő utasítás engedélyezi az EPROM kimenetét, így az adattartalom láthatóvá válik. Egy



8. ábra