

Az egér, a botkormány (joystick) és a nyomtató programozása DOS-ban

Az egér, a joystick és a nyomtató egyre elterjedtebb segédeszközzé vált a számítógép felhasználásában. Napjaink Windows alapú programjai könnyen is használják őket, de még számos olyan DOS alatti programot kell írni (játékok, gazdasági alkalmazások stb.), amelyek ezeket a perifériákat kezelni tudják. Tekintsünk hát be a perifériák DOS alatti világába.

Az egér

Az *egér* megszakítás szinten kommunikál a számítógéppel. Ez a megszakítás a **\$33**-as, ezt kell tehát nekünk programoznunk. A számítógép alapállapotban nem képes ennek a megszakításkérésnek a kielégítésére, ezért minden egérhez mellékelik az eszközmeghajtóját is (**driver**). Először is ezt a *driver*-t kell installáljuk az **autoexec.bat** vagy a **config.sys** állományban. Az installálás és indítás után jelenik meg az egérkurzor, ami az egér mozgását követi a képernyőn. Két típusú kurzorról beszélhetünk: *hardware*- és *software*-kurzorról. Ez a két típusú kurzort mind szöveges, mind grafikus üzemmódban értelmezett. A hardware-kurzort a monitort vezénylő áramkörök állítják elő közvetlenül, a software-kurzort pedig mi definiálhatjuk.

Lássuk tehát az egérmeghajtó rutinjainak a használatát. Az **AX** regiszterben kell megadni a kívánt megszakítás sorszámát, majd a többi paramétereket, ha léteznek, a BX, CX, DX, ES, SI, DI regiszterekbe tölteni, ezután pedig meghívni a **\$33**-as megszakítást. Az esetleges visszatérő paramétereket ugyancsak ezekben a regiszterekben kapjuk vissza. A koordináta-megadás esetén azonban transzformációt kell végezzünk, a következőképpen: Bemenetnél az egérkoordináta $(\text{KépernyőX}-1) * \text{KépernyőXFaktor}$, illetve $(\text{KépernyőY}-1) * \text{KépernyőYFaktor}$ lesz, kimenetnél pedig a képernyőkoordináta $(\text{EgérX} \text{ div } \text{KépernyőXFaktor}) + 1$, illetve $(\text{EgérY} \text{ div } \text{KépernyőYFaktor}) + 1$ lesz, ahol a **KépernyőXFaktor** illetve a **KépernyőYFaktor** a megfelelő képernyő üzemmódok faktorát jelenti a következőképpen:

Képernyőmód	X faktor	Y faktor
40x25 szöveg	16	8
80x25 szöveg	8	8
Grafikus módokban	1	1

Ez a koordinátatranszformáció érvényes az összes megszakítás-funkciónál.

A következő táblázat az egér legfontosabb megszakításrutinjait foglalja össze:

Funk.	Be	Ki	Jelentés
0	AX = 0000h	AX BX	A driver inicializálása. Visszatéréskor AX = 0 ha ez sikerült, AZ = \$FFFF ha hibás. A BX-ben a gombok számát kapjuk meg.
1	AX = 0001h	-	A kurzor láthatóvá tétele.
2	AX = 0002h	-	A kurzor láthatatlanná tétele.
3	AX = 0003h	BX CX DX	Az egér helyzetének lekérdezése. Visszatéréskor: BX : a gombok helyzete: 0: nincs gomb lenyomva, 1 a bal, 2 a jobb, 4 a középső, CX az egér X koordinátája, DX az egér Y koordinátája.
4	AX = 0004h BX az új X koordináta CX az új Y koordináta	-	Az egér mozgatása egy adott pontra.

5	AX = 0005h BX a gomb száma	AX BX CX DX	A legutolsó olvasás óta a kért gomb lenyomásának a száma. Ez a BX-ben lesz, AX-ben a gombok helyzete, CX, DX-ben pedig a koordináták.
6	AX = 0006h BX a gomb száma	AX BX CX DX	A legutolsó olvasás óta a kért gomb felengedésének a száma. Ez a BX-ben lesz, AX-ben a gombok helyzete, CX, DX-ben pedig a koordináták.
7	AX = 0007h CX X min. értéke DX X max. értéke	-	Mozgási intervallumot határoz meg vízszintes (X) irányban.
8	AX = 0008h CX Y min. értéke DX Y max. értéke	-	Mozgási intervallumot határoz meg függőleges (Y) irányban.
9	AX = 0009h BX az aktív pont X-je CX az aktív pont Y-ja ES:DX mutató a maszkokra.	-	A grafikus kurzor definiálása. A grafikus kurzor két maszkból: XOR és AND , és egy aktív pontból áll. A maszkokat két array[1..16] of word tömb tartalmazza, ezek a word számok a 16*16 -os dimenziós képpontokat jelentik.
10	AX = 000Ah BX kurzortípus CX, DX a megfelelő specifikációk.	-	A szöveges kurzor definiálása. Kurzortípus: software (0000h) CX: képernyőmaszk, DX: kurzormaszk; hardware (0001h) CX: a kurzor felső sora, DX: a kurzor alsó sora.
11	AX = 000Bh	CX DX	Az utolsó hívás óta az elmozdulások számát adja vissza. CX-ben a vízszintes, DX-ben a függőleges mozgásokat. Az érték pozitív jobbra, felfelé mozgásnál, illetve negatív balra, lefelé.
12	AX = 000Ch CX hívási maszk ES:DX a megszakítás címe (FAR)	-	A hívási maszk által meghatározott pillanatokban fellépő megszakítás megírása. Hívási maszk: Bit Esemény 0 A kurzor pozíciója változott 1 Bal gomb lenyomva 2 Bal gomb felengedve 3 Jobb gomb lenyomva 4 Jobb gomb felengedve 5 Középső gomb lenyomva 6 Középső gomb felengedve 7-15 Fenntartott
13	AX = 000Dh	-	Fényceruza emuláció bekapcsolva.
14	AX = 000Eh	-	Fényceruza emuláció kikapcsolva.
15	AX = 000Fh CX X érzékenység DX Y érzékenység	-	A képernyőn való elmozdulás érzékenységét állítja be. Alapbeállítás: X: 8, Y: 16.
16	AX = 0010h CX, DX bal felső sarok, SI, DI jobb alsó sarok koordinátái	-	A kurzor letiltása egy téglalap alakú területről.
18	AX = 0012h BH a kurzor szélessége CH a kurzor magassága BL az aktív pont X-e CL az aktív pont Y-a ES:DX a két maszk kezdőcíme	AX	A grafikus kurzor méretének megadása. AX-ben FFFFh siker esetén
29	AX = 001Dh BX a lap száma	-	A képernyőlap kiválasztása.

Egérkurzorok

Mint már említettük, az egér kurzora kétféle lehet: szöveges és grafikus kurzor. A kurzor megjelenítése a következőképpen történik: Definiálnunk kell egy képernyő-(**AND**) és egy kurzor-(**XOR**) maszkot. A számítógép először az adatblokkal (az a hely, ahol a kurzor megjelenik) és a megadott képernyőmaszkkal végez **AND** műveletet, majd az eredmény és a kurzormaszk között bonyolódik le **XOR** művelet és az eredmény jelenik meg a kurzorblokk helyén a képernyőn. A kurzort mozgatva tehát, az előző pozícióban automatikusan visszaállítódik a képernyő.

Szöveges kurzorok

A szöveges kurzor lehet **hardware**-kurzor, ezt közvetlenül a monitort vezénylő áramkörök állítják elő. A hardware kurzor szélessége egy karakterpozíció, a magasságát pedig állítani lehet. Ezenkívül értelmezett a **software**-kurzor, ez egy ASCII karaktert jelent és attribútum **word**-ja a következő felépítésű:

v	h	h	h	s	s	s	s	k	k	k	k	k	k	k	k
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

ahol **v** a villogást, **h** a háttérszint, **s** a szint és **k** a karaktert jelenti. Mindkét maszk felépítése megegyezik az előbb leírt attribútumword felépítésével.

Grafikus kurzorok

A grafikus kurzorok is kétfélék lehetnek: **hardware**- és **software**-kurzorok. A hardware-kurzort közvetlenül a gép állítja elő, a software-kurzort mi definiálhatjuk. Grafikus kurzorok esetén kell tudnunk, hogy melyik pontra mutat a kurzor, itt az alap-egység nem a karakter, hanem a **pixel**. Ezért meg kell adnunk az ún. aktív vagy referenciapont koordinátáit is. A maszkok megadása egy kissé körülményesebben történik. Először egy **16*16**-os táblázatban megadjuk a pontok helyeit, majd a vízszintes sorokat hexadecimális számokká alakítjuk. Így egy **1*16**-os táblázatot kapunk. Ezt a táblázatot kell megadni maszknak. Lássuk például egy nyíl alakú grafikus kurzornak a definiálását. Először megadjuk a táblázatokat:

A képernyőmaszk:

0111111111111111	\$7FFF
0001111111111111	\$1FFF
0000011111111111	\$07FF
0000000111111111	\$01FF
0000000001111111	\$007F
0000000000011111	\$001F
0000000000000111	\$0007
1111110001111111	\$FC7F
1111110001111111	\$FE3F
1111111000111111	\$FF1F
1111111100011111	\$FF8F
1111111110001111	\$FFC7
1111111111000111	\$FFE3
1111111111100011	\$FFF1
1111111111110001	\$FFF8
1111111111111100	\$FFF8
1111111111111111	\$FFFF

A kurzormaszk:

1000000000000000	\$8000
1010000000000000	\$A000
1000100000000000	\$8800
1000001000000000	\$8200
1000000010000000	\$8080
1000000000100000	\$8020
1111101111110000	\$FCF8
0000001010000000	\$0280
0000000101000000	\$0140
0000000010100000	\$00A0
0000000001010000	\$0050
0000000000101000	\$0028
0000000000010100	\$0014
0000000000001010	\$000A
0000000000000101	\$0007
0000000000000000	\$0000

Az átalakított hexadecimális számokat két vektorba írjuk, ezek lesznek a maszkok, majd megadjuk az aktív pontot, a nyíl hegyet: **(0,0)**.

A joystick

A **joystick** a másik kedvelt segédeszközünk, főleg játékoknál szoktuk használni. A joystick megszakításon, vagy porton keresztül kommunikálhat a számítógéppel. A joy-

stick-nak nem kell külön drivert installálnunk, a számítógép alapértelmezésben le tudja olvasni az adatait.

A joystick a **\$15**-ös megszakítás **\$84** funkcióját használja. Ez a megszakítás olvassa le a gombok helyzetét, illetve a koordinátákat.

A megszakítás használata:

Az **AX** regiszterbe **\$84**-t töltünk, a **DX**-be pedig a kívánt funkciót: **\$00** a gombok helyzetének beolvasása, **\$01** a koordináták beolvasása. Ezután meghívjuk a **\$15**-ös megszakítást. A megszakítás visszatérő értékei: Hiba esetén a **Carry Flag** be van állítva, az **AL** 4-7 bitjei a gombok helyzetét adják meg, a koordinátákat pedig a következőképpen kapjuk meg:

AX A joystick X koordináta
 BX A joystick Y koordináta
 CX B joystick X koordináta
 DX B joystick Y koordináta.

A joystick porton keresztül is elérhető. Minden joystick a **\$201**-es portot használja. Innen egy **word**-ot olvas be, amelynek a bitjei a következőket jelentik:

7	6	5	4	3	2	1	0
0	A joystick X koordináta			4	A joystick 1 gomb		
1	A joystick Y koordináta			5	A joystick 2 gomb		
2	B joystick X koordináta			6	B joystick 1 gomb		
3	B joystick Y koordináta			7	B joystick 2 gomb		

A gombok helyzetének a beolvasása nagyon egyszerű, csak meg kell nézni, hogy a megfelelő bit be van-e állítva, vagy sem. A koordináták leolvasása egy kicsit bonyolultabb. Először egy egyesekből álló bitsorozatot kell kiküldeni a porton, majd számolni, hogy a figyelt koordinátának megfelelő bit mikor lesz zéró. A számolt érték felel meg a figyelt koordinátának.

A nyomtató

A nyomtató megjelenésével még szorosabbra zárható a számítógép és a külvilág közötti kapcsolat, hisz ami a gépben van, az papíron is megjeleníthető. A nyomtató porton keresztül kommunikál a számítógéppel (LPT1, LPT2, LPT3, PRN). Láttuk, hogy a **Printer** unit egy **Lst**-nek nevezett szövegállományt rendel hozzá a porthoz, és ezáltal tudunk adatokat küldeni ki. A nyomtató szolgáltatásait megszakításon keresztül is elérhetjük. Ez a megszakítás a **\$17**-es. Ennek a megszakításnak három funkciója van:

\$00: Egy karaktert küld ki a nyomtatóra. Adatai:

be: ah,00h

al, a karakter

dx, a port száma: 0- LPT1, 1- LPT2, stb.

ki: ah, a nyomtató állapota.

\$01: A nyomtató inicializálása. Adatai:

be: ah,01h

dx, a port száma: 0- LPT1, 1- LPT2, stb.

ki: ah, a nyomtató állapota.

\$02: A nyomtató állapota. Adatai:

be: ah,02h

dx, a port száma: 0- LPT1, 1- LPT2, stb.

ki: ah, a nyomtató állapota.

A nyomtató állapotát egy **byte** hosszúságú bitsor jellemzi, a következőképpen:

7	6	5	4	3	2	1	0	a bitsor
1.....	a nyomtató szabad (0 foglalt)
.....1.....	a nyomtató működik
.....1.....	a papír ki van fogyva

.....1.....	nyomtató inicializálva
.....1.....	I/O hiba
.....0..0.....	nem használt port
.....1	Time-out hiba.

Nyomtató nagyon sokféle van. Itt az **EPSON FX-1000** és a vele kompatibilis típusú nyomtatók programozási kódjait közöljük. **w** a **write** rövidítése.

Funkció	be	Pascal	ki	Pascal
Inicializálás	ESC @	w(lst, #27'@')	-	-
Centered (középre)	ESC a #1	w(lst, #27'a'#1)	ESC a #0	w(lst, #27'a'#0)
Italic (dőlt)	ESC 4	w(lst, #27'4')	ESC 5	w(lst, #27'5')
Bold (vastag)	ESC E	w(lst, #27'E')	ESC F	w(lst, #27'F')
Underlined (aláhúzott)	ESC 1	w(lst, #27'1')	ESC 0	w(lst, #27'0')
DoubleStrike (duplázott)	ESC G	w(lst, #27'G')	ESC H	w(lst, #27'H')
NLQ nyomtatás	ESC x 1	w(lst, #27'x1')	ESC x 0	w(lst, #27'x'0')
PS nyomtatás	ESC p 1	w(lst, #27'p1')	ESC p 0	w(lst, #27'p1')
Condensed (sűrített)	^Q	w(lst, #17)	^R	w(lst, #18)
Dupla szélesség	ESC W 1	w(lst, #27'W1')	ESC W 0	w(lst, #27'W0')
Dupla magasság	ESC w 1	w(lst, #27'w1')	ESC w 0	w(lst, #27'w0')
Aláírányítás	ESC S 1	w(lst, #27'S1')	ESC T	w(lst, #27'T')
Felírányítás	ESC S 0	w(lst, #27'S0')	ESC T	w(lst, #27'T')
Lapkidobás	^L	w(lst, #12)	-	-
Új sor (LF)	^J	w(lst, #10)	-	-
Sor elejére (CR)	^M	w(lst, #13)	-	-

Kovács Lehel

tudománytörténet

Kémia történelmi évfordulók

2001. szeptember - október

270 éve, 1731. október 10-én született a franciaországi Nizzában *Henry CAVENDISH*. Jelentős vagyont örökölve hatalmas könyvtárat és laboratóriumot szerelt fel magának és életét a tudománynak szentelte. Főleg a gázok fizikai és kémiai tulajdonságaival foglalkozott. Felfedezte a nitrogént, de eredményeit nem közölvén, az elsőség D.Rutherfordnak jutott. Előállította és izolálta a hidrogént, meghatározta a sűrűségét és ezért őt tartják a hidrogén felfedezőjének, habár a létét előtte már jelezte Paracelsus, Van Helmont és Boyle is. J.Priestleyvel egyidőben hidrogén-klorid gázt állított elő. J.Wattal közösen megvalósította a víz szintézisét eudiométer-csőben, bizonyítva ezzel, hogy a víz nem elem, mint ahogy azt korábban vélték. Elektromos szikra segítségével levegőben nitrogén-oxidokat állított elő. Közel járt az argon felfedezéséhez, kimutatva azt, hogy a levegő nagyon kis mennyiségben egy közömbös gázt is tartalmaz. Meghatározta a szén-dioxid sűrűségét és kimutatta a mészkő oldódásában játszott szerepét, valamint azt, hogy alkoholos erjedéskor is keletkezik. A flogisztonelelmélet híve volt és az általa felfedezett hidrogént tiszta flogisztonnak vélte. Megállapította, hogy az