

Dobos Péter főiskolai adjunktus:

A BILLENTYŰMÁTRIX KEZELÉSE

A számítástechnikai foglalkozásokon az új anyagrészek oktatását egy vagy több működő program bemutatásával teszem hatékonyabbá. A következőkben olyan programot mutatok be, amelyik alkalmas:

- annak szemléltetésére, hogy egy billentyű lenyomásakor milyen feladatokat old meg a "gép";
- a GETKEY utasítás működésének bemutatására és
- annak illusztrálására, hogy hogyan építhető fel egy BASIC program és egy gépi kódú rutinból álló oktató program.

A számítógép billentyűzetének használatát a forgalomban lévő kézikönyvek alaposan ismertetik. A billentyűk programból történő lekérdezéséről azonban csak annyit ismertetnek, amennyi a GET és a GETKEY utasítások használatához feltétlenül szükséges. A programozási technika megalapozásához azonban szükséges a gépi háttér ismerete is.

A billentyűzet folyamatos lekérdezését a hardvermegszakító rutin végzi. Az éppen lenyomott billentyű ASCII kódja a billentyűzetpufferbe kerül. A BASIC interpreter (a GET utasítás is) innen veszi ki - a beírás sorrendjében a karaktereket. Ez az átmeneti tárterület a S0527...S0530 címeken helyezkedik el. Hossza alaphelyzetben 10 byte-nyi. Ha terjedelmét csökkenteni akarjuk, például amikor csak egy-egy billentyű lenyomására vár programunk, akkor 1-et írunk a S053F címre.

A S0527-es címen található a következő, még fel nem dolgozott billentyű kódja. Ha az átmeneti tároló megtelt, a lenyomott

billentyűnek nincs hatása. A \$00EF címen elhelyezett szám adja meg a pufferbe került, de még fel nem dolgozott karakterek számát. GET utasítás előtt a puffer ürítése zérusnak a \$00EF címre történő írással szokott megtörténni.

Mivel a billentyűkódokat a gépi rutin a pufferbe viszi át, a felhasználót az érdeklí, hogy hogyan olvassa ki onnan saját programjával. Ebben az esetben a következőket kell elvégezni:

- az első karaktert ki kell venni a \$0527 címről;
- a \$0520 - \$0530 címmezőt 1-gyel lejjebb kell tolni és
- a \$00EF tartalmát 1-gyel kell csökkenteni.

Ezeket a feladatokat a GETIN (\$FFE4) rutin végzi el. A billentyűzetpufferből az AC regiszterbe olvassa a soron következő karaktert.

A billentyűzet logikailag egy 8x8-as mátrixba van rendezve, ennek 8 vonala (sorok) egy 6529B típusú portáramkörhöz csatlakozik, 8 vonala (oszlopok) pedig a TED K0 - K7-es vonalaira kapcsolódik. Az egyes billentyűk a sor- és oszlopvonalak keresztezési pontjain találhatók.

Nyomdatechnikai okból a billentyűzetmátrix ábráját a tanulmány végére helyeztem.

A ROM-ban az \$E026 - \$E129 címtartományban a billentyűmátrixnak megfelelő dekódoló táblázat van, amely az egyes billentyűkhöz rendelendő ASCII kódokat tartalmazza.

A billentyűzetolvasás folyamata a következő:

- első lépésként egy sort kell megcímezni: az \$F030 címre kell kiírni egy olyan byte-ot, amelyben csak a vizsgálni kívánt sor bitje zérus. A beírt byte-ot a 65298 (U27-es) tárolja;
- a TED \$FF08 című regiszterébe bekerül a K0...K7-es vonalak pillanatnyi állapota. Amennyiben a vizsgált sorban le van nyomva egy billentyű, ennek hatására zérusba viszi a saját pozíciójának megfelelő oszlop-vonalakat is. A többi bit természetesen egy értékű marad;
- a teljes billentyűzetvizsgálathoz ezt a műveletsorozatot minden sorra el kell végezni.

A „lekérdezés”-hez először ezt kell megvizsgálni, hogy az illető billentyű a mátrix melyik sorában van. A „K” például az 5-ös sorban van. Ebben az esetben a \$FF30 címre 1110 1111 (= \$EF)-et kell írni, ezután olvasnunk kell az \$FF08 címről - egy byte-ot. A kiolvasott értékben az a bit lesz zérus értékű, amelyiknek megfelelő oszlopban levő billentyű lenyomott állapotban van. A jelen esetben a 3-as oszlopban van zérus, azaz 1101 1111 (= \$DF)-et fogunk olvasni.

Ha az O, K és M billentyű egyszerre vannak lenyomva, akkor 1000 1111 (= \$9F)-et olvashatunk ki.

Az előzőekben leírtak alapján bármely billentyű lenyomására várakozó programciklus a következőképpen építhető fel:

```
CIKL STA $FD30 ; a maszk beírás a sor regiszterbe
      STA $FF08 ; a TED betölti a regisztert
      LDA $FF08 ; az oszlop kiolvasása
      CMP #0FF ; ha $FF jött be, akkor
      BEQ CIKL ; folytatjuk a vizsgálatot, egyéb-
      NOP ; ként tovább ...
```

A program működése:

- a 20 - 110-es sorokban van elhelyezve a táblázat, amelyet a 660-as sor ír ki;
- a 180 - 430-as sorokban a billentyűmátrix felhasználási útmutatója található;
- az 510-es sorban történik meg az aktuális sor kiválasztása;
- az 560-as sor a maszk kitárolását végzi;
- az 570-es programsor hívja meg az említett gépi kódú szubrutint;
- az 590-es sor jeleníti meg a kapott értéket, amiből megállapítható, hogy mely billentyű(k) van(nak) lenyomott állapotban;
- a 600-as sor időt biztosít a felhasználónak egy-egy sor vizsgálatához és
- a 680 - 790-es sorokban a billentyűmátrix kezelését végző BASIC- és gépi kódú program listája van elhelyezve.

Az egész feladatot olvéző program listája a táblázat után van elhelyezve.

Ezt a programot az elmúlt kiképzési évben már használtam. Letuttatása eredményesnek bizonyult. Egyrészt jól kiegészítette a tanári magyarázatot, másrészt a hallgatók a működő programot tanulmányozhatták, és ez az elméleti anyag elmélyítését is szolgálta. Ez a program felhasználható a COMMODORE gépcsalád más tagjainól is - természetesen a memóriacímek különbözőségének figyelembevételével.

Tanulmányozásához és esetleges felhasználásához az olvasóknak minden segítséget megadok és sok sikert kívánok!

A BEÍRT ÉRTÉK SFD308 -ON	A KAPOTY ÉRTÉK SFF80 -ON									
	0111 1111	1011 1111	1101 1111	1110 1111	1111 0111	1111 1011	1111 1101	1111 1110		
1111 1110	a	F3	F2	F1	HLP	---	RTN	DEL		
1111 1101	SFT	E	S	Z	4	A	W	3		
1111 1011	X	T	F	C	6	O	R	5		
1111 0111	V	U	H	B	0	G	Y	7		
1110 1111	N	O	K	M	Ø	J	I	9		
1101 1111	,	-	:	.	FEL	L	P	LE		
1011 1111	/	+	=	ESC	JBB	:	K	BAL		
0111 1111	STP	Q	C→	SPC	2	CTL	HOX	1		

```

10 SZ=0: REM BILLENTYU BAS
20 W$(0) = "SOR      K A R A K T E R E K"
30 W$(1) = " 0      @  F3  F2  F1  HLP  =L= RTN DEL"
40 W$(2) = " 1      SFT  E   S   Z   4   A   W   3   "
50 W$(3) = " 2      X   T   F   C   6   D   R   5   "
60 W$(4) = " 3      V   U   H   B   8   G   Y   7   "
70 W$(5) = " 4      N   O   K   M   0   J   I   9   "
80 W$(6) = " 5      ,   -   :   .   FEL  L  P  LE"
90 W$(7) = " 6      /   +   =   ESC  JBB ; *  BAL"
100 W$(8) = " 7      STP  Q  C=> SPC 2  CTL  HOM  1  "
110 W$(9) = "SZAM 128 64 32 16 8 4 2 1  "
120 PRINT "KESZULT A KGYRMF"TAB(50)"TERM. TUD."
121 PRINT "TAB(10)"TANSZEKEN"
130 CHAR1,6,11,"PROGRAMOZAS BASIC NYELVEN ES"
140 CHAR1,15,13,"GEPI KODBAN"
150 CHAR1,5,15,"COMMODORE PLUS/4 SZAMITOGEPEN"
160 CHAR1,7,22,"A BILLENTYUMATRIX KEZELESE"
170 GETKEYW$
180 PRINT "EZ A PROGRAM A BILLENTYU-MATRIX FEL-"
190 PRINT "HASZNALASARA KESZULT, ALKALMAS TOBB "
200 PRINT "BILLENTYU EGYIDEJU LENYOMASANAK ERZE-"
210 PRINT "ELESERE. INDITASKOR A FELHASZNALO "
220 PRINT "ADJA MEG A MATRIX KIVALASZTOTT SORAT. "
230 PRINT "AZ ALSO SORBAN LATHATO SZAM SEGITSE-"
240 PRINT "BEVEL AZONOSITHATJUK A LENYOMOTT "
250 PRINT "BILLENTYUKET. HA A SOR KIVALASZTA-"
260 PRINT "SANAL AZ S BILLENTYUT NYOMJUK LE, "
270 PRINT "AKKOR A PROGRAM FUTASA BEFEJEZODIK. "
280 GETKEYW$
290 PRINT "A BILLENTYUZETOLVASAS FOLYAMATA: "
300 PRINT "EGY SORT KELL MEGCIMEZNI: AZ $FD30 "
310 PRINT "IMRE KELL IRNI EGY DLYAN BYTE-OT, "
320 PRINT "AMELYBEN CSAK A VIZSGALNI KIVANT SOR"
330 PRINT "ITJE ZERUS. "
340 PRINT "TED BEOLVASSA AZ $FF00 REGISZTERBE"
350 PRINT "AZ OSZLOPOK PILLANATNYI ALLAPOTAT. "
360 PRINT "AMENNYIBEN A VIZSGALT SORBAN LE VAN "
370 PRINT "NYOMVA EGY BILLENTYU, AZ 0-BA VISZI "
380 PRINT "SAJAT POZICIOJANAK MEGFELELO "
390 PRINT "OSZLOPVONALAT IS, A TOBBI PEDIG 1. "
400 GETKEYW$
410 PRINT "A TELJES BILLENTYUVIZSGALATHOZ EZT"
420 PRINT "MUVELETSOROZATOT MINDEN SORRA EL "
430 PRINT "KELL VEGEZNI. "
440 PRINT "A PROGRAM LISTAJA: "
450 GETKEYW$
460 GOSUB 600
470 GETKEYW$
480 PRINT "A PROGRAM VEGREHAJTASA KOVETKEZIK:"
490 GETKEYW$
500 PRINT "": GOSUB 660
510 CHAR1,2,22,"": INPUT "SOR=0...7 ";S$
520 IF S$="S" THEN END
530 IF LEN(S$) <> 1 THEN 500

```

```
540 J=ASC(S#): S=VAL(S#)
550 IF J<48 OR J>55 THEN 500
560 POKE 2034,255-21S
570 SYS 829
580 CHAR 1,2,24,"": PRINT"OSSZEG= ";
590 PRINTUSING"#####";255-PEEK(2034);
600 SZ=SZ+1: IF SZ<=300 THEN 560
610 CHAR1,2,22," "
620 CHAR1,2,24," " : SZ=0
630 CHAR1,31,24,"": PRINT"ITTE VAGYAB?";
640 GETKEYW#: IF W# <> "T" THEN 640
650 CHAR1,31,24," " : GOTO 510
660 FOR J=1 TO 11: CHAR1,2,2*J-1,W#(J-1)
670 NEXT J: RETURN
680 PRINT"VADNAI INPUT 'SOR= 'S
690 PRINT"POKE 2034,255-21S MASZK KITAROLAS"
700 PRINT"SYS 829 SZUBRUTINHIVAS"
710 PRINT"PRINT 255-PEEK(2034) FLAG-BYTE"
720 PRINT"GOTO 2 SAREG-BOL"
730 PRINT"*=*033D"
740 PRINT"SEI MEGSZAKITAS LETILTASA"
750 PRINT"STA $FD30 MASZK BEIRAS A SOR REG-BE"
760 PRINT"STA $FF08"
770 PRINT"LDA $FF08 OSZLOP FLAG KIOLVASAS"
780 PRINT"CLI MEGSZAKITAS FELOLDASA"
790 PRINT"RTS VISSZATERES A FOPROGRAMBA";
800 RETURN
```

FELHASZNÁLT IRODALOM

- VADNAI SZABOLCS: COMMOORE PLUS/4 programozási zsebkönyv (1986.),
- GÁSPÁR-GYENES: A PLUS/4 belső felépítése (1988.),
- BABÁN-MASA: Gépi kódú programozás (1988.).