



Ezt a programot már hónapokkal ezelőtt beharangoztuk. Azután minden volt valami fontosabb. A VC 20-as rajongók viszont azóta többször reklámálták megjelentetését. Közben egyik külső munkatársunk elkezdte a program átirását C 64-re is. Ahogy az lenni szokott – a fejlesztés közben nem várt problémák adódtak. Amint a gondok megoldódnak, közöljük majd a szükséges átalakítások listáját!

TG BASIC

Tóth Géza egyik hasznos programjával már találkozhattak a VC 20 tulajdonosok 1986/5. számunkban. A szerző 16 éves, és ilyen korban keveseknek adatik meg, hogy bármit is – utcát, teret – elnevezzenek róluk. Most azonban egy olyan kiváló segédprogrammal rukkolt elő, amely méltán képviseli alkotóját, így monogramja alapján a program neve TGBASIC. Nem kevesebbet tud, mint a 16 kbyte-tal bővített VC 20-ast 32 új utasítással vértezi föl – így gépünket akár VC 21-esnek tekinthetjük.

Hogy mik ezek az új utasítások? Soroljuk fel őket, szintaktikájuk megjelölésével – az olvasó talán már ebből is rájön, hogy mennyi mindenre is jó ez a bővítés.

HIRES
PLOT x,y,t
CUR x,y,a\$
COLOR b1,b2,b3,b4
SOUND hg,m,t
OLD
& a
KEY a\$
PROC
FRA (x)

HCLS
TEXT x,y,a\$
CSET a
LOC m,n
IN hg
DEL a,b
@ a
CGOTO x
FIR

LORES
TYPE a,b
CHAR a,h0,...h7
VOL n
OUT hg
REPEAT a
! a
ELSE...
JOY

A négy új függvény pedig a következő:
PNT (x,y) **CPT** (x,y)
SCL (x,y)

SCR (x,y)

A sorrend nem volt véletlenszerű: a részletes ismertésnél ugyanígy kerülnek majd sorra az új utasítások.

A TGBASIC PROGRAM HASZNÁLATA

Végül lássuk programunk elhelyezkedését a memoriában egyrészt amiatt, hogy áttekinthessük felépítését, másrészt pedig azért, hogy biztosak legyünk a program memoriában elfoglalt helyében – nehogy egy POKE utasítással akaratlanul elrontsuk korábbi munkánkat.

Memória térkép

4096	0,5k kisfelbontású képernyő 3,75k HIRES képernyő
8192	0,5k gépi kódú program
8704	0,5k kisfelbontású képernyőnek HIRES grafika alatt (illetve kisfelbontású grafika esetén a HIRES kép 256 byte-jának)
9216	2k program
11264	BASIC program

kezdőcím: \$2126



UTASÍTÁSOK, FÜGGVÉNYEK

NAGYFELBONTÁSÚ ÉS TOBBSZÍNU GRAFIKA

HIRES – be

HCLS – képernyőtörles

LORES – ki

PLOT x,y,t – pont kigyújtása

Itt, és a következő grafikai utasításoknál x,y a koordinátákat jelenti. Értéke a kétféle üzemmódban a következő lehet:

nagyf. mód **többszín mód**

x 0–159 0–79

y 0–192 0–192

"t" a rajzolás típusa, a következő táblázat szerint:

t nagyf. gr. **többszín mód**

0 törlés háttérszínnel ír

1 kigyújtás keretszínnel

2 invertálás tintaszínnel

3 invertálás kiegészítő színnel

TEXT x,y,a\$ – szöveg kiírása

TYPE a,b – a TEXT-tel kiírt betűk méretének megadása, a és b értéke egyaránt 0 és 15 között változhat – ekkorára tudjuk nagyítani az eredeti betűméretet. E két utasítás használatára mutat példát leírásunk végén az 1. program.

Figyelem!

BASIC-ünk bővített változatában csak 4 kontrollkarakter működik:

CRSRUP (kisbetűk), **CRSRDOWN** (nagybetűk)

RVSON (inverz mód be), **RVSOFF** (inverz mód ki)

Van két hasznos új függvényünk:

PNT(x,y) – megadja, hogy az adott koordinátahelyen lévő pont kigyújtott (tintaszínű) vagy sem (háttérszínű). A függvény alkalmazását illusztrálja a 2. program.

CPT(x,y) – azt adja meg, hogy mit tartalmaz az adott ponthoz tartozó színtároló cella. (Ha értéke nagyobb 7-nél, akkor többszínmódban dolgozunk.)

KISFELBONTÁSÚ GRAFIKA

Ebben a módban x értéke 0 és 21 közötti lehet, y-é pedig 0 és 22 közötti.

CURx,y,... – képernyőre kiíratás adott pozíójú karakterhelytől kezdve. Használata hasonló a PRINT-éhez, arra persze ügyelni kell, hogy x és y értéke a megadott tartományban legyen.

Az utasítást a 3. példaprogram mutatja be.

Két, a grafikához kapcsolódó függvény:



LISTÁK

SCR(x,y) – az adott pozícióban levő karakter képernyőkódját, illetve
SCL(x,y) – a hozzá tartozó színtároló cella tartalmát adja. (Ha nagyobb 7-nél – többszín mód.)

KARAKTERES GRAFIKA

(Amit mindenekelőtt meg kell jegyeznünk: a nagyfelbontású grafika tárolója fedi a karakteres grafika tárolóját.)

CSET a – a karakterekkel való „manipulálást” segíti, egyszerűsíti.
„a” értéke 0 és 4 közötti lehet, jelentése pedig a következő:

0 nagybetűk

1 kisbetűk

2 programozható karakterek

3 nagybetükészlet a RAM-ba + mint 2.

4 kisbetükészlet a RAM-ba + mint 2.

CHAR k, h0, h1,...,h7 – Adott képernyőkódú karakter alakjának meghatározása.

Az utasítás természetesen a **CSETa** előzetes megadása után használható. k és h értéke 0 és 255 közötti lehet. Egy betű 8*8 képpontból áll, azaz egy 8*8-as mátrixot alkot. Ennek minden sora 1 byte, minden sorának egy pontja pedig egy bit. Ha a betűben az adott helyen tintaszínű a pont, akkor a megfelelő bit értéke 1, ha háttérszínű, akkor 0. Többszín módban egy betű 4*8-as mátrixot alkot, minden ponthoz 2 bit tarozik.

A CSET és a CHAR utasítás használatát a 4. program mutatja be. (A most még esetleg érthetetlen jelölésekkel annyit, hogy a felkiáltójelek a mögöttük álló számokat konvertálják kettes számrendszerből tízesbe.)

Általános grafikai utasítások:

A többszínű üzemmódban négy színt használhatunk egy karakterhelyen (egy betűméretnek megfelelő nagyságú négyzeten) belül: egy háttérszínt, egy keretszínt, egy tintaszínt és egy kiegészítő színt.

COLOR b1, b2, b3, b4 – ez az utasítás szolgál a négyféle szín megadására.

b1 papírszín

b2 az alsó 3 bit a keretszín, a 3. bitet általában 1-re szokás állítani. Ha értéke mégis 0, a képernyő „inverzen látszik”.

b3 tintaszín (ha nagyobb 7-nél: többszín mód)

b4 többszín mód kiegészítő szín

A b-vel jelölt változók 0 és 15 közöttiek lehetnek.

A többszín karakter tervezését az 5. program mutatja be.

LOC m,n – a képernyő pozíciójának megadása

m 0 és 127, n pedig 0 és 255 között változhat.

Hang

A továbbiakban hg jelöli a hanggenerátort (értéke 0 és 3 közötti), m a hangmagasságot (90-tól 127-ig), t pedig az időt, 1/60 másodpercben. (Legnagyobb értéke 65535 lehet.)

VOL n – a hangerő beállítása (n:0–15)

SOUND hg,m,[t] – hang kiadása

IN hg és **OUT hg** – adott hanggenerátor be-, illetve kikapcsolása.

A SOUND utasítás bemutatásánál a t körüli szögletes zárójel azt jelzi, hogy e változó értékének megadása elhagyható! Ebben az esetben a hang addig szól, amíg a bekapcsoláshoz hasonló módon ki nem kapcsoljuk. Érdemes megnézni a 6. példát a hanggal kapcsolatos új utasítások használatáról.

PROGRAMOZÁST SEGÍTŐ UTASÍTÁSOK

OLD – NEW-val törölt program visszahozása

DEL s1,s2 – sorok törlése a listából. s1 az első, s2 pedig az utolsó törölni kívánt sor száma.

```

* C= UJSAG      SORSZAM: 094 *
*          TGBASIC
* PROGRAM: TOTH GEZA
*****
2000 84 49    sty $49      y elemetése
2002 24 0f    bit $0f      TOKENKIIRAS
2004 30 37    bmi $203d   ha idézjelen belül,ugrás
2006 c9 60    cmp #$60
2008 90 33    bcc $203d   ha a kód kisebb,mint $60,ugrás
200a c9 ff    cmp #$ff
200c 00 2f    beq $203d   pi?
200e a2 20    ldx #$20
2010 a0 24    ldy #$24
2012 c9 80    cap #$80
2014 90 06    bcc $201c   saját token
2016 e9 11    std #$21
2018 92 9d    ldx #$9d
201a 00 c0    ldy #$c0
201c 86 61    stx #$61
201e 84 62    sty #$62
2020 a0 00    ldy #$00
2022 e9 5f    sbc #$5f
2024 aa      tax
2025 f0 08    beq $202f
2027 c8      iny
2028 b1 61    lda ($61),y
202a 10  fb    bpl $2027
202c 00      dex
202d 00  f8    hex $2027
202f c8      iny
2030 b1 61    lda ($61),y
2032 30 05    bmi $2039
2034 20 47    cb    jsr $cb47
2037 d0 f6    bne $202f
2039 a4 49    ldy $49
203b 29 7f    and #$7f
203d 20 47    cb    jsr $cb47
2040 48      pha
2041 c8      iny
2042 b1 5f    lda ($5f),y
2044 00 06    ldy $204c
2046 ae 0d    ldx $208d
2049 ca      dex
204a f0 fa    beq $2046
204c 88      dey
204d 68      pla
204e 4c f6    jmp $c6f6
2051 20 73    jsr $0073
2054 f0 2a    beq $2080
2056 c9 8c    cmp #$8c
2058 d0 1b    ldy $2075
205a 20 73    jsr $0073
205d d0 00    ldy $2065
205f 20 1d    cb    jsr $81d
2062 4c ae    jmp $204d
2065 20 21    jsr $21a2
2068 5f      ldy $5f
206a 38      sec
206b e9 01    sbc #$01
206d a4 60    ldy $0
206f 20 24    cb    jsr $c824
2072 4c ae    jmp $c7ae
2075 c9 a7    cmp #$a7
2077 f0 31    beq $20aa
2079 c9 8b    cmp #$8b
207b f0 20    beq $209d
207d ad 00    tay
207e 00 06    ldy $2086
2080 20 79    jsr $0079
2083 4c e7    jmp $c7e7
2086 20 80    jsr $208c
2089 4c ae    jmp $c7ae
208c c9 6a    cmp #$6a
208e 90 5d    bcc $20ed
2090 0a      asl
2091 a8      tay
2092 b9 d0    ldy $23d0,y
2094 00 00    ldy $0
2096 b9 cf    ldy $23cf,y
2099 48      pha
209a 4c 73    jmp $0073
209d 20 73    jsr $0073
209f 20 9e    cd    jsr $cd9e
20a3 a5 61    ldy $61
20a5 5f      sta $fb
20a7 4c ad    jmp $20ad
20aa 20 73    jsr $0073
20ad 38      sec
20ae a2 76    ldy $#76
20b0 b0 05    bcs $20b7
20b2 10      tlc
20b3 a2 a7    ldy #$a7
20b5 68      pla
20b7 a5 fb    lda $fb
20b9 00 03    bcc $20be
20b9 20 22    bne $20df
20bd 20 0f    ldy $1ff0
20c0 20 00    ldy $0
20c0 20 c9    jsr $c90b
20c3 aa      tax
20c4 f0 10    beq $20d6
20c6 80 0a    sty $0a
20c8 a0 0b    ldy $#8b
20ca 20 00    ldy $0
20cc 00 00    ldy $0
20cc c4 0a    cpy $0a
20cc 00 00    ldy $0a
20d1 b0 03    bcs $20d6
20d3 20 09    c9    jsr $c909
20d6 20 fb    jsr $c8fb
20d9 20 79    jsr $0079
20dc 4c 54    jmp $2054
20df 20 79    jsr $0079
20e2 b0 06    bcs $20ea
20e4 20 a0    c8    jsr $c8a0
20e7 4c ae    jmp $c7ae
20e9 20 00    ldy $#00
20e9 00 00    ldy $0
20e9 00 00    ldy $0
20f1 84 93    sty $93
20f3 05 0a    ldy $0a
20f5 4a      lsr
20f6 b1 7a    ldy ($7a),y
20f8 f0 29    beq $2123
20fa b0 18    bcs $2114
20fc c9 3a    cmp #$3a
20fe 00 23    beq $2123
2100 ad 73    ldy #$73
2102 00 00    ldy $0
2104 c9 b2    cmp #$b2
2106 f0 15    beq $211d
2108 c9 28    cmp #$28
210a d0 02    bne $210e
210c e6 93    inc #$93
210e c9 29    cmp #$29
2110 d0 02    bne $2114
2112 c6 93    dec #$93
2114 c8      iny
2115 c9 22    cmp #$22
2117 d0 da    beq $2043
2119 c9 00    ldy $0
211b d0 06    bne $2043
211b 20 79    jsr $0079
2120 4c a5 c9    jmp $c9a5
2123 4c 37    jmp $2937
2126 a0 0c    ldy #$0c
2128 b9 ce 24    ldy $24ce,y
212b 99 ff    sta $02ff,y
212e 88      dey
212f d0 f7    bne $2128
2131 04 fb    sty #$fb
2133 00 fc    sty $fc
2135 04 fd    sty $fd
2136 04 fe    sty $fe
2139 0c ff    ldy $2bff
213c a9 2c    ldy #$2c
213e b5 2c    sta $2c
2140 84 2b    sty $2b
2142 20 fa e3    jsr $e3fa
2145 20 8f    jsr $2a8f
2148 68      pla
2149 68      pla
214a a9 eb    ldy #$eb
*****
```

RESTORE

THEN

ELSE

a visszaugrási cím kivétele

az utolsó IF kif. előjelle

nem 0

#=2c+beq \$20df

keresés az X-ben tárolt kar. után,

nincs meg,sorvég

Y mentése

keresés IF után

előbb volt IF,mint ELSE v THEN

az utolsó IF kif. előjelle

ELSE volt

az előjel elementése

a köv. utasítás végrehajtása

vissza a ROM-ba

műtök

valós kif. beolv.

az előjel

saját utasítás végrehajtása

vissza a ROM-ba

vissza a ROM-ba

végrehajtása

az előjel mutato beáll.

zárójel számító beáll.

során köv. karakter

sorvég,tehát PROC hivas

idézjelen belül

műtök

végrehajtás

mint GOTO

köt. utasítás végrehajtása

LET v. PROC hivas?

idézjel mutato beáll.

zárójel számító beáll.

vissza,ha nem idézjel

számító csökkentése

vissza

CHRGET

PROC hivas

BASIC kezdet 44*256

44*256 cím-be 0+43/44 növelése

IRQ,BRK,NMI,OUTPUT vektor áll.

214c a0 24 ldy #\$24
 214e 4c 0f e4 jmp \$e0f szöveg ki+utána direkt mód
 2150 2d ce jsr \$cefd vesszó ell.
 2154 a9 10 lda #10
 2156 2c a9 16 bit \$16a9 =#2c+lda #\$16
 2157 2c a9 04 bit \$00a9 =#2c+lda #\$04
 2158 2c a9 17 bit \$17a9 =#2c+lda #\$17
 2159 2c a9 80 bit \$00a9 =#2c+lda #\$80
 2162 85 c1 sta \$c1 számbeolv. megadott felső határral
 2164 20 d7 jsr \$d79e 1 byte-os szám beolv.
 2167 e4 c1 cpw \$c1 a beolvásott érték összehasonlítása
 2169 90 30 bcc \$219b a maximálissal,ha kisebb,RTS-hez
 216b 4c 48 d2 jmp \$d248 illegal q. error
 216e 20 54 21 jsr \$2154 1.paraméter be(0-15) COLOR
 2170 85 0a asl rts
 2172 0a asl 4 bittel balra
 2173 0a asl
 2174 0a asl
 2175 0a asl
 2176 85 0a sta \$0a tárolás
 2178 20 51 21 jsr \$2151 "," beolv + 2.paraméter (0-15)
 217b 0a txa
 217c 05 0a ora \$0a tárolás a VIC chip 15. regiszterében
 217e 8d 0f 90 sta \$900f 3.paraméter be(0-15)
 2181 20 51 21 jsr \$2151 a tintaszín beirásá
 2184 8e 02 stx \$0286 4.paraméter be(0-15)
 2186 20 51 21 jsr \$2151 4.paraméter be(0-15)
 2188 0a asl
 2189 0a asl 4 bittel balra
 218c 0a asl
 218d 0a asl
 218e 0a asl
 218f 05 62 sta \$62 beirása a VIC chip
 2191 ad 0e 90 lda \$900e 14. regiszterének felső
 2194 29 0f and \$0f 4 bit-jén
 2196 05 62 ora \$62
 2198 8d 0f 90 sta \$900e
 219b 60 rts
 219c 22 a2 21 jsr \$21a2 sorszám be CGOTO
 219f 4c a3 c9 jmp \$c8a3 ugrás a ROM-beli GOTO rutin elő részére
 21a2 8a cd jmp \$cd8a 2 byte-os kif. SORSZ.REOLV.
 21a5 20 f7 d7 jsr \$d7f7 beolv. \$14/\$15-be
 21aa 8a 15 idx \$15
 21ac b0 bd bcs \$216b ugrás,ha HI nagyobb,mint \$fa
 21ae 20 13 c6 jsr \$c613 sorkeresés
 21b1 b0 e8 bcs \$219b ha megvan,RTS
 21b3 4c e3 c8 jmp \$c8e3 UNDEF'D STATEMENT ERROR
 21b6 20 60 21 jsr \$2160 1.paraméter be(0-127)
 21b9 86 0a sta \$0a
 21b9 8d 0a 90 da \$0000 tárolása a VIC chip
 21b9 29 00 and \$00 0. regiszterének alsó
 21c0 05 0a ora \$0a 7 bitjén
 21c2 8d 00 90 sta \$9000
 21c5 20 f1 d7 jsr \$d7f1 ","+1 byte-os szám beolv.
 21c8 8e 01 stx \$0001 a VIC 1. regisztere
 21cb 60 rts
 21cc a9 03 lda #\$03
 21ce 20 62 21 jsr \$2162 paraméter be(0-2) REPEAT
 21d1 8a txa
 21d2 4a lsr
 21d3 6a ror
 21d4 6a ror
 21d5 8d 0a 02 sta \$028a
 21d8 60 rts
 21d9 a6 c6 ldx \$c6 X-ben a bill. pufferben levő KEY
 21db 0f fc beq \$21d9 karakterek
 21dd 4c 7b cb jmp \$cb7b ROM-beli GET rutin
 21e0 d0 f6 bne \$21db RTS-re,ha az utasításszó után OLD
 21e2 a0 01 ldy #\$01 van még valami
 21e4 98 tya
 21e5 91 2b sta (\$2b),Y sorok újraláncolása
 21e7 20 33 c5 jsr \$c533 a változókézdet
 21ea a6 02 ldy #\$22 kiszámítása
 21ec 69 02 adc #\$02
 21ed 05 2d sta \$22
 21f0 a5 23 lda \$23
 21f2 69 00 adc #\$00
 21f4 85 2e sta \$2e
 21f6 20 60 c6 jsr \$c660 CLR
 21f9 4c 74 c4 jmp \$c474 READY.+parancs mód
 21fc 00 brk
 21fd 00 brk
 21fe 00 brk
 21ff 00 brk
 2200 ff ????

2589 d0 5d bne \$2568 HIRES
 2590 20 65 26 jsr \$2665 Biztos kisfelbontású mód?Ha nem,hiba.
 259e a0 10 ldy #\$10 A 2.képernyőterület feltöltése
 259f 98 tya a 16-tól a 255. helyig 16,17,...,255-tel
 25a1 99 f0 21 sta \$21ff0,y Miután az 1. és a 2. képernyőterületet
 kicséréljük,ez az elsőbe kerül.
 25a5 d0 f9 bne \$2510
 25a6 22 cc ldx \$c6
 25a7 a9 17 ldy #\$17
 25a8 22 cc ldx \$c6
 25a9 a9 14 ldy #\$14
 25a9 00 0c bne \$252b
 25da d0 47 bne \$2568
 25db 00 0c bne \$2568
 25e1 20 68 26 jsr \$2668 Hibajelzés,ha nem nagyf. mód.
 25e2 a9 16 lda #\$16
 25e6 a2 c0 ldx #\$c0
 25e8 a0 2e ldy #\$2e
 25a2 18 clc A kis-/nagybetű váltás engedélyezése v.
 25b2 6b 91 02 ror \$0291 letiltása
 25c2 8e 90 stx \$9005 VIC chip 5. regiszter
 2531 8e 90 stx \$9003 VIC chip 3. regiszter
 2532 8e 90 04 stx \$9004 VIC chip 2. regiszter
 2537 a2 02 ldx #\$02 A RUNSTOP+RESTORE letiltása
 2539 8e 91 stx \$911e
 253c a5 d2 lda \$d2 A cursor alatt levő karakter címe(HI)
 253e 49 32 eor #\$32 a képernyőterületen
 2540 85 d2 sta \$d2
 2542 29 fe and #\$fe A képernyőterület kezdetének HI byte-ja
 2544 8d 88 02 sta \$0288 (ahova a PRINT ir.)
 2547 20 b2 ea jsr \$eab2 A cursor alatt levő karakter szín-RAM
 254a a2 17 ldx #\$17 beli címének beállítása
 254c b5 d9 lda #\$09,x (217-240=a képernyősor kezdetének
 254d 4c 32 eor #\$32 MSB-je.A 7. bit azt mutatja meg,
 2550 75 09 lda #\$07,x Kapcsolódik-e az adott képernyősor
 2552 ca dec a elölözött.)
 2553 10 f7 bpl \$254c Az 1. és a 2. képernyőterület csereje
 2555 a0 10 ldy #\$10
 2557 a9 22 lda \$22
 2559 20 69 25 jsr \$2569 cseré rutin
 255c a0 94 ldy #\$94 Az 1. és a 2. színterület cseréje
 255e a9 96 ldy #\$96
 2560 20 69 25 jsr \$2569 cseré rutin
 2563 a9 82 lda #\$82 A RUNSTOP+RESTORE engedélyezése
 2565 88 1e 91 sta \$911e
 2566 60 rts
 2569 a2 02 ldx #\$02 X-ben a lapok száma CSERE RUTIN
 256b 18 clc Y-ban az egyik terület kezdőcímének HI
 256e 84 62 sty \$62 byte-ja,A-ban a másiké
 2570 a0 00 ldy #\$00
 2572 84 61 sty \$61
 2574 84 63 sty \$63
 2576 a9 00 lda #\$00
 2578 b0 08 bcs \$2582 Ha c=1,akkor csak az egyik terület
 257a b1 61 lda (\$61),y törlése.
 257c 48 pha
 2581 b1 63 ldy (\$63),y
 2584 c8 iny
 2585 d0 ef bne \$2576
 2587 e6 62 inc \$62
 2589 e6 64 inc \$64
 258b ca dex
 258c d0 e8 bne \$2576 Ha van még lap,visszaugrás.
 258d 60 rts
 258f d0 d7 bne \$2568 Biztos nagyf. mód?
 2594 a2 0f ldx #\$0f hossz=15 lap=15*256 byte
 2596 a9 11 lda #11 kezdőcím 11=256
 2598 20 6e 25 jsr \$256e CSERE rutin(c=1,tehát törles)
 259b a9 01 lda #01 A szín-RAM feltöltése
 259d 9d 00 94 sta \$9400,x 1-gyel.(fehér)
 25a0 ca dex
 25a1 d0 fa bne \$259d
 25a3 60 rts

25a4 20 68 26 jsr \$2668 Biztos nagyf. mód? KÖRD BEOLV.
 25a5 20 9e d7 jsr \$d79e 1 byte beolvásása
 25a6 86 0a lda \$0a Eltárolása X koordinátaként
 25a7 20 bb 25 jsr \$25bb Az X koordinátához ellenőrzése
 25a8 86 93 lda \$93 Eltárolása Y koordinátaként
 25a9 20 9e 01 lda \$0a Hiba,ha nem kisebb 192-nél
 25ba 60 rts
 25bb ad 86 02 lda \$0286 Az X koord. ellenőrzése
 25be c9 09 cmp #\$08 Többszin mód?(Tintaszín > 7?)
 25c0 a5 05 lda \$0a
 25c2 90 01 bcc \$25c5 Hiba,ha nem kisebb 160-nál
 25c4 0a asl cmp #\$a0 Ha igen,X:=2*X
 25c5 70 01 bcc \$25b4
 25c9 4c 48 d2 jcd \$d248
 25d0 86 02 lda \$0286 Hiba,ha nem kisebb 160-nál
 25d1 08 and #\$08 PONT HELY KISZAMITASA
 25d2 f0 02 beq \$25d6 Ha többszin mód,X:=X*2
 25d4 06 0a asl cmp #\$a0
 25d6 a5 93 lda \$93
 25d8 29 f0 and #\$f0
 25da 05 62 sta \$62
 25dc 4a lsr
 25dd 4a lsr
 25e0 18 clc
 25e1 05 62 sta \$62
 25e3 55 0a lda \$0a
 25e5 4a lsr
 25e7 4a lsr
 25e8 18 clc
 25e9 65 bb adc \$bb
 25eb 85 bb sta \$bb
 25ed 69 10 adc #\$10
 25ef 85 62 sta \$62
 25f1 84 04 ldy #\$04
 25f3 05 65 ldy \$65
 25f4 04 00 ldy #\$00
 25f7 06 62 asl cmp #\$62
 25f9 26 63 rol \$63
 25fb 88 dey
 25fc d9 f9 bne \$25f7
 25fe a5 63 lde \$63
 2600 09 10 ora #\$10
 2602 29 1f and #\$1f
 2604 85 63 sta \$63
 2606 09 73 lde \$73
 2608 09 01 and #\$0f
 260a 05 62 ora \$62
 260c 85 62 sta \$62
 260e a9 80 lde \$80
 2610 85 b9 sta \$b9
 2612 a5 0a lde \$0a
 2614 29 07 and #\$07
 2616 f0 06 beq \$261e
 2618 a8 tay
 2619 46 b9 lsr \$b9
 261b 88 dey
 261c d9 fb bne \$2619
 261d 29 00 ldy #\$00
 261f 40 09 ldy \$09
 2621 a5 b9 lde \$b9
 2623 4a lsr
 2624 05 b9 ora \$b9
 2626 85 b9 sta \$b9
 2628 46 0a lsr \$0a
 262a a5 b9 lde \$b9
 262c 49 ff eor #\$ff
 262e 85 66 sta \$66
 2630 60 rts
 2631 20 84 25 jsr \$25e4 Koord. beolv.
 2632 20 84 21 jsr \$215a A típus beolvásása(0-3)
 2633 20 84 25 jsr \$25cc A pont helyének kiszámítása
 2634 a0 00 ldy #\$00
 2636 ad 86 02 lde \$0286
 2642 91 bb sta (\$bb),y A szín beállítása
 2644 29 08 and #\$08
 2646 f0 0c beq \$2654 Nem többsz. mód.
 2648 8a txa
 2649 46 b9 lsr \$b9
 264b b0 03 bcc \$2650
 264d 0a asl
 264f 90 f9 bcc \$2649
 2650 85 b9 sta \$b9
 2652 a1 01 ldy \$01
 2654 b1 62 lde (\$62),y A pontot tartamazó byte a kar.gen.-ból
 2656 ca dex
 2657 30 07 bni \$2660 Tipus=0
 2659 d0 02 bne \$265d Tipus=2 v. 3
 265b 25 66 and #\$66 Pont kigyűjtése
 265d 45 b9 eor #\$b9 Invertálása
 265f 2c 25 66 bit #\$6625 (=#2c+and \$66)törlése
 2662 91 62 sta (\$62),y És a byte eltárolása
 2664 60 rts
 2665 a9 10 lde #\$10 BIZTOS KISF. MOD?
 2666 29 22 ldy \$22
 2668 dd 02 cmp \$0288
 266d f0 55 ldy \$2664
 266f a2 1f ldx #\$1f
 2671 a5 d2 lde \$d2
 2675 90 07 bcc \$267e
 2677 86 b9 stx \$b9
 2679 20 21 25 jsr \$2521 X mentése
 267c a6 b9 ldx #\$b9
 267e e0 1f cpx #\$1f X vagyis a hibákod)<\$1f-->a ROM rutinra
 2680 90 0a bne \$268c A HI byte
 2685 85 22 sta \$22
 2687 a1 21 ldy \$21
 2689 20 84 22 ldy \$22
 2690 20 84 02 cmp \$0288
 2691 f0 55 ldy \$2664
 2692 a1 21 ldy \$21
 2693 4c 34 54 ldy \$445
 2694 4c 34 c4 ldy \$445
 2695 20 57 21 jsr \$2157 A ROM hibázáset kiíró rutinja
 2696 86 0a lde \$0a X koord. (0-21) be
 2697 20 5d 21 jsr \$215d A cursor címe(HI)<\$14-->kisf.mód.
 2698 86 0a lde \$0a X eld
 2699 a4 01 ldy #\$01 Y koord. (0-22) be
 269a a4 0a ldy #\$0a Y reg.-be az X koordináta
 269b 20 0a e5 jsr \$50a A cursor az adott pozicióból
 269c 49 0d ca jmp \$ca9d C=0,ha INc,c=1,ha OUT.
 269d 20 5a 21 jsr \$215a Hanggenerátor száma(0-3) be
 269e 20 5d 21 ldy \$215d Alapértelmezett eld
 269f bd 0a 90 lde \$900a,x A hanggenerátor tartalma
 26a0 29 7f and #\$7f 7-bit=(0:kikapcsolás)
 26a1 20 5d 02 bcc \$26b2 Ugrásba OUT volt.
 26a2 00 00 ora #\$00 7-bit=1(bekapcsolás)
 26b2 9d 0a 90 sta \$900a,x A megváltoztatott érték
 26b3 60 rts
 26b4 20 54 21 jsr \$2154 Hangerő(0-15) be
 26b5 86 62 stx \$62 Tárolása
 26b6 ad 0e 90 lde \$900e És betevésse a VIC chip 14.
 26b7 20 50 00 and #\$f0 regiszterének alsó 4 bitjére
 26c0 4c 96 21 jmp \$2196 Hanggenerátor száma(0-3) be
 26c3 20 5a 21 jsr \$215a Tárolása
 26c6 86 0a stx \$0a "," beolv.
 26c8 20 fd ce jsr \$cefd Hang (0-127) beolv.
 26d0 20 60 21 jsr \$2160 Hang generátor száma(0-3) be
 26e8 8a 0a txa
 26f0 09 00 ora #\$00 Az adott hanggenerátor bekapcsolása
 26f1 44 0a ldy \$0a És a hang beállítása
 26f3 99 0a 90 sta \$900a,y Van még valami
 26f4 20 79 00 jsr \$0079 Ha nincs vége
 26f5 d0 01 bne \$26dc "," beolv.
 26f6 60 rts
 26f7 20 fd ce jsr \$cefd 2 byte-os szám beolv
 26f8 20 8a 00 lde \$0079 \$14/\$15-be
 26f9 70 f7 d7 jsr \$d7f7 IRQ megszakítás letiltása
 26fa a9 02 sei
 26fb a9 02 lde \$a2 Anaklik kiszámítása,hogy áll
 26fc 18 02 clc idő mulva
 26fd 65 14 adc \$14 Anaklik kiszámítása,hogy áll
 26eb 85 14 sta \$14 idő mulva
 26ed a5 01 lde \$01
 26ef 65 15 adc \$15
 26f1 85 15 sta \$15
 26f3 4c f7 26 jmp \$26f7 IRQ engedélyezése
 26f5 58 cli
 26f7 a5 a1 lde \$a1
 26f9 c5 15 cmp \$15

TG BASIC

REPEAT a – billentyűismétlés szabályozása

„a” értékének jelentése:

O: alapállapot

1: egyik billentyű sem ismétel

2: minden egyik ismétel

KONVERTÁLÓ FÜGGVÉNYEK

& tizenhatos számrendszerből, @ négyes számrendszerből, ! pedig kettes számrendszerből konvertál tízes számrendszerre. A konvertálható szám legfeljebb 2 byte nagyságú lehet. Pl. !101 = 5

PROCEDÚRÁK

Procedúrákat (azaz eljárásokat) is használhatunk programkészítés során. Ezek olyan szubrutinok, amelyeket nevükkel tudunk meghívni, nem pedig sorszám szerint, mint a GOSUB-ot. A procedúráknak paramétereit, változókat tudunk átadni az utasításban definiált változólistán keresztül.

PROC név (változó, változó,...) „procedúra definíálása”

pl.: 10PROC IRJ(A\$):PRINT A\$:RETURN

hívás: IRJ("ABC")

válasz: ABC (mintha A\$ = "ABC":GOSUB10 lett volna).

Megfigyelhetjük, hogy ha egy procedúrát például a következőképpen definíálunk:

PROC CIRCLE (x,y,R), akkor a zárójelen belül megadott változók képesek lesznek arra is, hogy más változók értékét felvegyék. Így:

A=60:B=60:C=10:CIRCLE(A,B,C)

Egy ilyen eljárásra mutat példát a 7. program.

Vigyázat! A procedúra nem zárt szubrutin:

Ha a végrehajtás a PROC-ra kerül, a gép figyelmen kívül hagyja a RETURN-t, és a program a következő sorra lép!

Még egy fontos megjegyzés: a procedúra nevének első karaktere nem lehet token, és nem lehet a névbén egyenlőséggel.

EGYÉB

KEY A\$ – a 10 KEY A\$ sor megfelel a 10 GETA\$:I-FA\$= ""GOTO 10 BASIC sornak.

CGOTOx – kiszámított GOTO, azaz x változó is lehet.

RESTORE x – az utasítással a datamutatót adott sorra is be tudjuk állítani

IF-THEN-ELSE

A feltételeles utasításra is új lehetőségeket ad a program. A legegyszerűbb, ha ezeket példákon nézzük meg:

a) **IF A = B THEN PRINTC** helyett most írhatjuk azt, hogy **IF A = B PRINTC**

b) **IF A = B GOTO 10** helyett: IF (A=B)10

c) **IF A = B PRINTC:ELSE PRINTX** helyett: IF A = B ELSE PRINTX:THEN PRINTC

d) **IF A = B PRINTC:ELSE180:THEN PRINTA**

e) Külön sorban a THEN és az ELSE a legutolsó IF-re vonatkozik

FIR – a joystick tűzgombjának leolvasása. A kapott érték jelentése:

0 – nincs lenyomva

1 – le van nyomva a tűzgomb

JOY – a joystick irányának leolvasása

Értéke 0-tól 15-ig terjed. Vigyázat! most kettes szám-

```

26fb d0 f9 bne $26f6 Vissza,ha a két HI byte nem egyenlő
26fd a5 a2 lda #a2
26ff c5 14 cmp $14
2701 d0 f3 bne $26f6 Vissza,ha a két LO byte nem egyenlő
2703 a6 0a ldx #0a
2705 58 cli
2706 10 a1 bpl $26a9 IN/OUT-ra(c=1-->a hanggenerátor ki)
2708 4c 09 c6 jmp $c609 Vége a tokenizálásnak
2709 a6 7a ldy #004 TOKENIZALAS
2710 a6 04 sty #004 Az "állapot jälző" 4-re állítja
2711 04 e8 inx (Idézjelen kívül,nem DATA v REM után.)
2714 bd 00 02 lda $0200,x A soron köv. karakter
2717 99 fb 01 lda $01fb,y Ø-->sorvége,vége a tokenizálásnak
271a f0 ec beq $2708
271c c9 20 cmp #$20
271e 90 08 bcc $2728 ASCII kód < $20
2720 c9 60 cmp #$60
2722 90 0a bcc $272e ASCII kód < $60
2724 c9 44 cmp #$44
2726 a6 ea bne $2712 PI(3.14...) volt
2728 a5 0f ldx #0f
2729 04 00 lda $0043 Nem DATA után+idézjelen kívül
2730 30 e4 bne $2712
2731 d0 06 bne $2738 Nem idézjelen
2732 a5 0f lda $0f Az állapotjelző
2734 49 00 eor #$00 7. bitjének invertálása
2736 85 0f sta $0f
2738 a9 01 lda $0f
273a c9 03 cmp #$03
273c d0 0b lda $02479
273d bd 00 02 lda $0200,x
2741 c9 3a cmp #$3a
2743 d0 cd bne $2712
2745 a9 04 lda $004
2747 d0 35 bne $2776 Ha ":" volt,az állapotjelző
2749 c9 04 cmp #$04 alapállapotba hozása
274b d0 c5 bne $2712 Az áll. jelző nem 4 ->vissza
274d bd 00 02 lda $0200,x Kérdőjelek?
2750 c9 3f cmp #$3f
2752 d0 09 ldx #09
2754 a9 99 ldy #09
2756 00 00 01 lda $001fb,y Ha igen,helyette
2759 d0 b7 bne $2712 a PRINT kódja
275b 84 b8 sty #b8 Vissza a ciklus elejére
275d 86 b7 stx #b7 X/Y elemzése
275f a9 c0 lda #$c0 A régi tokenek utáni keresés
2761 a0 9e ldy #9e keresés
2763 a2 cb ldy #$cb
2765 20 08 27 jsr $2788 Megvan
2768 90 0b bcc $2775 Az új tokenek utáni keresés
276a a9 24 lda #$24
276c a0 21 ldy #$21
276e a7 07 ldx #$07
2770 20 00 27 jsr $2788
2773 80 9d bcs $2712 Nincs meg
2775 b9 fb 01 lda $01fb,y A beírt token kódja
2778 49 00 eor #$00
277a c9 04 cmp #$04
277c d0 04 bne $2782 REM volt-->állapotjelző=15
277e 85 05 sta $0f
2780 b0 90 bcs $2712 DATA volt-->állapotjelző=3
2782 c9 03 cmp #$03
2784 f0 f8 beq $277e TOKENKERESÉS
2786 d0 8a bne $2712
2788 86 0a stx $0a
278a 80 0b sta $0b
278c 05 bc sta $bc Y és A=a keresés kezdőcímé-1
278e 0a 00 asl
278f b0 02 bcs $2793
2791 a9 60 lda #$00
2793 85 62 sta $62 Az utolsó token kódja-1
2795 a0 ff ldy #$ff
2797 a6 b7 ldy #b7
2799 c8 iny
279a b1 bb lda ($bb),y A tokentáblázatbeli betű
2799 dd 00 02 cmp $0200,x és az aktuális betű összehasonlítása
279f d0 07 ldy #2788 Nem egyezik
27a0 10 80 and #$00
27a3 00 20 bne $27c5 Vége a szónak
27a5 e8 inx
27a6 d0 f1 bne $2799 Nincs vége-->vissza
27a8 90 10 bcc $27ba
27aa 29 7f and #$7f
27ac dd 00 02 cmp $0200,x így sem egyezik
27af d0 14 bne $27c5
27b1 a4 b8 ldy #b8 Megvolt a token
27b3 a5 62 lda $62 Y elő, a token kódjának
27b5 99 fb 01 sta $01fb,y eltárolása
27b7 18 cic
27b9 00 00 rts
27ba 85 61 sta $61
27bc bd 00 02 lda $0200,x
27bf 29 7f and #$7f
27c1 c5 61 cmp #$61
27c3 e0 ec beq $27b1 Rövidítés volt,ezért megvan
27c5 a5 62 cmp #$0a
27c7 c5 0a bcs $27d4 Vége a keresésnek
27cb e6 62 inc $62 Tokenzámláló növelése
27cd b1 bb lda ($bb),y Előre a következő
27cf 30 00 bne $2797 tokenig
27d2 d0 f9 iny
27d4 a6 b7 ldx #b7 X,Z elő
27d6 a4 b8 ldy #b8
27d8 60 rts
27d9 20 65 26 jsr $2665 Biztos kisf. mód?
27dc a9 05 lda #$05 Paraméter (0-4
27de 02 62 21 jsr $2162
27e1 e0 02 cpx #$02 Ha par.<2--> a kis-/nagybetű váltás
27e3 6e 91 02 ror $0291 engedélyezése,egyébként letiltása.
27e6 e0 03 cpx #$03 Par.<3
27e8 70 00 02 bne $2501,x
27ea b0 01 25 lda $012501,x
27eb a1 18 ldy #18
27ef a0 08 ldy #$08
27f1 20 6b 25 jsr $256b RAM --> ROM
27f4 a2 02 ldy #$02
27f6 b0 06 25 lda $2506,x
27f9 60 00 05 sta $9005 A VIC chip 5. regiszterének beáll.
27fc 60 00 rts Alsd 4 bit adjá a karaktergen. helyét.
27fd 20 a4 25 jsr $25a4 Koord. be
2800 20 fd cpx $0cf0 TEXT
2803 20 9e cd jsr $cd7e "
2806 20 a3 d0 ldy #6463 A sztring beolvasása
2809 85 06 sta $06
280b a6 06 ldy #1000
280d 85 0d sta $0d
280f 24 0e ldy #0d
2811 c4 0e cpy #$0e
2813 f0 e7 beq $27fc Vége
2815 b1 22 lda ($22),y A következő karakter
2817 aa tar
2818 29 60 and #$60
281a 00 1e bnd $283a Nem kontroll karakter
281c ba 00 txa
281d 29 7f and #$7f
2821 c9 12 cmp #$12
2823 ba 00 bne $262a
2824 10 02 bpl $2828 CHR$(18) v. CHR$(146) (RVS ON v. OFF)
2826 a9 00 lda #$00 RVS ON
2828 85 c7 sta #c7 Tárolása az inverz írást jelző címen
282a c9 11 cmp #$11
282c d0 09 bne $2837 CHR$(17) v. CHR$(145) (CRSR DOWN v. UP)
282e 8a 00 txa
282f 29 80 and #$80 A kis-/nagybetű mód jelző beállítása
2831 4a 00 lsr
2832 4a 00 lsr
2833 4a 00 lsr
2834 4a 00 lsr
2835 85 fe sta $fe
2837 4c b6 28 jmp $28b6 A rutin végére
2838 ba 00 txa
2839 10 0b bpl $2848 Az ASCII kód átváltása
283b 29 7f and #$7f képernyőkoddá
283f e8 inx
2840 d0 02 bne $2844
2842 a9 5c lda #$5c
2844 09 40 ora #$40
2846 d0 0a bne $2852

```

VC-20

2848 c9 60 cmp #\$60
 284a 90 04 bcc \$2850
 284c 29 df and #\$df
 284e d0 02 bne \$2852
 2850 29 3f and #\$3f
 2852 83 14 sta #14
 2853 05 c7 cmp #c7
 2854 29 10 and #\$10
 2855 0a asl
 2856 0a asl
 2857 0a asl
 2858 0a asl
 2859 0a asl
 285a 0a asl
 285b 05 14 ora #14
 285d 20 11 2a jsr #2111
 2860 65 fe adr \$fe
 A képernyőkód elmentése
 Az RVS kapcsoló beleszámítása
 2862 49 80 eor #\$80
 2864 85 15 sta #15
 2866 a3 93 lda #93
 Y koord. mentése
 2869 48 00 ldy #\$00
 286b 55 fd ldy #\$fd
 Y irányú méret #\$09-be
 286d 85 09 sta #09
 286f a5 0a lda #0a
 X koord. mentése
 2871 48 pha
 2872 a2 80 ldx #\$80
 2874 86 6c stx #\$6c
 2876 a5 fc lda #\$fc
 A bitminta beállítása
 2878 85 08 sta #08
 287a b1 14 lda #(\$14), y Byte a ROM karaktergenerátorából
 287c 25 6c and #\$6c A bit levalasztása a bitminta alapján
 287e f0 09 beq \$2885 A pontot nem gyűjtük ki.
 2880 86 02 lsr #0288 Kinyitjuk
 2883 4a lsr Ha többsz. mód->írási típus=2,
 egyébként 1.
 2884 4a lsr
 2885 4a lsr
 2886 18 clc
 2887 69 01 adc #\$01
 2889 84 6b sty #\$6b
 Y elementése
 X-be az írási típust
 288c aa tax
 288e 20 bb 25 jsr #\$25bb
 Az X koord. ellenőrzése
 288f 20 b4 25 jsr #\$25b4
 Az Y koord. ellenőrzése
 2892 20 23 26 jsr #\$263a Pont tevése
 2895 a4 6b ldy #\$6b
 Az X. koord. növelése
 2897 e0 0a inc #0a
 dec #08
 2899 00 dd bpl \$287a Nincs kész az X irányú nagyítás
 289d 46 6c lsr #\$6c A bitminta jobbra forgatása
 289f d0 d5 bne \$2876 Még mincs kész a sor
 28a1 a6 0a ldx #0a
 X koord. elő
 28a4 85 0a sta #0a
 28a6 e9 93 inc #93
 28a8 c0 09 dec #09
 28a9 10 c3 bpl \$286f Még nincs kész az Y irányú nagyítás
 28ac c8 iny
 28ad c0 08 cpy #\$08
 28af d0 68 bne \$286b Még nincs kész a 8 sor
 28b1 68 pa lpa
 28b3 65 93 sta #93
 Y koord. elő
 28b4 86 0a ldy #0a X koord. a betű után
 28b6 66 0d inc #0d A számítáló növelése
 28b8 4c 0f 28 jmp \$2880 A következő karakterre
 28b9 20 54 21 jsr #2154 1. paraméter(0-15) be
 28be 86 fc stx #\$fc és eltárolása X irányú méretként TYPE
 28c0 20 51 21 jsr #2151 "+2. (0-15) be
 28c3 86 fd stx #\$fd és eltárolása Y irányú méretként
 28c5 60 rts
 28c6 20 a2 21 jsr #21a2 1. sorszám be
 28c9 a5 5f ldy #5f és tárolása #c1/c2-ön
 28cb 85 c1 sta #c1
 28cd 46 60 lde #60
 28cf 95 52 sta #c2
 28d1 20 fd ce jsr \$cefd ", " beolv.
 2. sorszám be
 28d4 20 a2 21 jsr #21a2
 28d7 a0 00 ldy #\$00
 28d9 b1 5f lda #(\$5f), y
 28db aa tax
 28dc c8 iny
 28dd b1 5f lda #(\$5f), y
 28df 85 60 sta #60 \$5f/\$60 most már nem a 2.sor
 28e1 85 5f sta #5f kezdetré, hanem a az utána levő
 28e3 8a txa kezdetré mutat.
 28e4 38 sec #61/#62 a kitörőrend
 rész hossza
 28e7 65 c1 sbc #c1
 28e8 65 61 lda #61
 28e9 65 60 lda #60
 28eb e5 c2 sbc #c2
 28ed 85 62 sta #62
 28ef 90 2d bcc \$291e \$5f/\$60 < #c2/#c3
 28f1 a5 2d lda #2d \$2d/\$2e(a változó ter. kezdete)beáll
 28f3 38 sec (cim=cím - eltöről byte-ok száma)
 28f4 e5 61 sbc #61
 28f6 85 2d sta #2d
 28f8 a5 2e lda #2e
 28f9 e5 62 sbc #62
 28fa 85 2e lda #2e
 28fe 88 dev Y reg.=0
 28ff b1 5f lda #(\$5f), y Másolás
 2901 91 c1 sta #(\$c1), y
 2903 65 5f inc #5f 1.számítáló növ.
 2905 d0 02 bne \$2909 1.számítáló növ.
 2907 e6 60 inc #60
 2909 e6 c1 inc #c1 2.számítáló növ.
 290b d0 02 bne \$290f
 290d e6 c2 inc #c2
 290f a5 c2 lda #c2
 2911 c5 2e cmp #2e
 2913 d8 ea bne \$28ff Nincs vége,tovább!
 2915 a5 2d cmp #2d
 2917 2d bne \$28ff Nincs vége,tovább!
 291b 20 33 c5 jsr #c533 A programsorok üraláncolása
 291e 20 60 c6 jsr #c660 CLR
 2921 18 clc
 2922 4c 4b c8 jmp \$c84b READY.+direkt mód
 2925 a0 01 ldy #\$01 \$7a<->\$b7
 2927 b9 7a 00 lda #007a,y \$7b<->\$b8
 292c b6 b7 lda #b7,y
 292d 99 b7 00 sta #0067,y
 292e 76 7a sta #7a,y
 2931 18 dey
 2932 10 f3 bpl \$2927
 2934 4c 73 00 imp \$0073 CHRGET
 2937 a5 7a lda #7a A CHRGET vektor
 2939 d0 02 bne \$293d csökkentése
 293b c6 7b dec #7b
 293d c6 7a dec #7a
 293e a6 7a lds #7a A CHRGET vektor elmentése
 2941 a4 7b ldy #7b
 2945 86 b9 stx #b9
 2947 a5 2b ldy #2b A számítáló alaphelyzetbe
 2949 38 03 sec #03 állítása
 294a 85 03 sbc #03
 294c 85 b7 lda #b7
 294e a5 2c lda #2c
 2950 e9 00 sbc #00
 2952 85 b8 sta #b8
 2954 20 25 29 jsr \$2925 Számítáló->CHRGET vektor+CHRGET
 2957 20 73 00 jsr #0073 CHRGET
 2958 c9 7c cmp #\$7c
 295c f0 2b bed \$2989 PROC token
 Előre a köv.
 295e 20 f8 c8 jsr #c8ff
 2961 20 79 00 jsr #0077 CHRGOT
 2964 a6 00 tax Nem sorvég
 2965 d0 f0 bne \$2955
 2967 a0 02 ldy #02
 2969 b1 7a lda #(\$7a), y Nincs vége a keresésnek
 296b d0 05 bne \$2972 UNDEF"D" PROCEDURE ERROR
 296d a2 20 ldx #02
 296f 4c 71 26 jmp \$2671
 2972 c8 iny
 2973 b1 7a lda #(\$7a), y Sorszám LD
 2975 85 bb sta #bb
 2977 c8 iny
 2978 b1 7a lda #(\$7a), y Sorszám HI
 297a 85 b5 sta #bc
 297c a5 7a lda #7a A CHRGET vektor
 Növelése 4-gyel
 297f 69 04 adc #04
 2981 85 7a sta #7a
 2983 90 d2 bcc \$2957 Tovább keres
 2985 e6 7b inc #7b

2987 d0 ce bne \$2957 Tovább keres
 2989 20 25 29 jsr \$2925 Számítáló->CHRGET vektor+CHRGET
 298c 85 0a sta #0a A beolvasott karakter elmentése
 298e 20 25 29 jsr \$2925 Számítáló->CHRGET vektor+CHRGET
 2991 c5 0a cmp #0a Nem azonos a 2 karakter
 2992 d0 59 bne \$2991 CHRGOT
 2995 20 79 00 jsr #0079 Vége az utasításnak
 2998 c9 28 cmp #\$28
 2999 d0 eb bne \$2999 A beolvasott karakter tárolása
 299e 20 73 00 jsr #0073 "()"
 29a1 20 8b d0 jsr #008b Változó beolvasása
 29a4 85 49 sta #49
 29a6 84 4a sty #4a CHRGOT
 29a8 20 79 00 jsr #0079 A beolvasott karakter tárolása
 29ab 85 0a sta #0a "()"
 29af c9 29 cmp #\$29
 29b1 20 25 29 jsr #2925 Számítáló->CHRGET vektor+CHRGET
 29b3 d0 1a ldy #0a Ertékadás
 29b5 20 25 29 jsr #c9b1 CHRGOT
 29b8 20 b1 c9 jsr #c9b1 Vége az utasításnak
 29b9 20 79 00 jsr #0079 Számítáló->CHRGET vektor+CHRGET
 29bf 20 25 29 jsr \$2925 Számítáló->CHRGET vektor+CHRGET
 29c2 68 pla "()"
 29c3 c5 0a cmp #0a Nem azonos a 2 karakter, hiba
 29c5 d0 08 bne \$29c5
 29c9 c9 29 cmp #\$29
 29d1 20 25 29 jsr #2925 Van-e elég hely a veremben?
 29d2 68 pla Az eredeti CHRGET vektor
 29d3 68 pla elmentése
 29d4 a9 c7 lda #c7 A visszatérési cím
 29d6 48 pha elmentése
 29d7 a9 e9 lda #e9 A sorszám tárolása
 29d8 a9 03 lda #03 Van-e elég hely a veremben?
 29d9 20 25 29 jsr #c3fb Az új sorszám beállítása
 29d9 a5 b8 lda #b8 A GOSUB kódja a verembe
 29e1 20 25 29 jsr #c3fb Az új sorszám beállítása
 29e2 48 pha Ugrás a ROM-beli interpreter ciklusba
 29e3 20 25 29 jsr #c3fb ":"
 29e4 48 pha Nem Ø(nem sorvég)
 29e5 20 25 29 jsr #c3fb Ha az egik név végen ":" ,
 29e6 48 pha a másik végén CHR\$(Ø) akkor is jó.
 29e7 20 25 29 jsr #c3fb A CHRGET vektor
 29e8 48 pha előhozása
 29e9 20 25 29 jsr #2a06 Tovább kekresés
 29e9 49 3a eor #3a Rutin a CHAR és a TEXT számára
 29e9 50 3a cmp #3a (\$14-ben eltárolja A regisztert,
 29e9 51 3a inc #3a \$15-ben visszaadja a nyolcszorosát, utána
 29e9 52 3a ldy #b9 A-ba beleteszi #15 tartalmát.
 29e9 53 3a ldx #b9 Biztos kisf.mód?
 29e9 54 3a ldy #15 A képernyőkód(0-255)beírása CHAR
 29e9 55 3a ldx #15 *8
 29e9 56 3a ldy #15 HI:=HI+\$18
 29e9 57 3a ldx #15 B szám lesz, a számítáló beállítása
 29e9 58 3a ldy #15 A byte eltárolása
 29e9 59 3a ldx #15 A számítáló növelése
 29e9 60 3a ldy #15 Nincs még meg a 8 szám,v. ssza
 29e9 61 3a ldx #15 A regiszterek elmentése NMI RUTIN
 29e9 62 3a ldy #15 A szín-RAM mutató HI < #96 (kisf.mód)
 29e9 63 3a ldy #15 LORES
 29e9 64 3a ldy #15 Vektorok a ROM-ból a RAM-ba
 29e9 65 3a ldy #15 Saját vektorok betöltése
 29e9 66 3a ldy #15 Folytatás a ROM NMI rutinjában
 29e9 67 3a ldy #15 Saját vektorok beállítása
 (NMI,BRK,IRQ(ez az egy marad.)
 29e9 68 3a ldy #15 A CHROUT vektor
 29e9 69 3a ldy #15 beállítása
 29e9 70 3a ldy #15 A elmentése
 29e9 71 3a ldy #15 Nem karakteres grafika,nem nagyf. mód
 29e9 72 3a ldy #15 A karakter CHR\$(14) v. CHR\$(142)
 29e9 73 3a ldy #15 Idézőjel mód->vissza a ROM-beliire
 29e9 74 3a ldy #15 A regiszterek előhozatala,CLC,RTS
 29e9 75 3a ldy #15



TG BASIC

rendszerben dolgozunk!

A 0. bit ég ha jobbra, az 1. ha fel, a 2. ha le, a 3. pedig, ha balra van elfordítva a botkormány karja.

FRA(x) – egy szám törtrészét veszi

Ugyanez a megszokott BASIC-ben: $FRA(x) = x - INT(x)$ képlettel számítható. **Még egy hasznos szolgáltatás:** a bővítő segítségével a LIST megállításához használható a SHIFT gomb. Amíg nyomva tartjuk, addig áll, utána pedig folytatódik a listázás.

KÉT ÚJ HIBAJELZÉS

UNDEF'D PROCEDURE – úgy hívtunk egy eljárást, hogy az nem létezik.

BAD GRAPHIC MODE – kisfelbontású, vagy karakteres grafikánál LORES, HCLS, PLOT, TEXT, PNT, CPT alkalmazása, nagyfelbontásúnál HIRES, CSET, CHAR használata.

PÉLDÁK

Példa 1. betűk írása a nagyfelbontású képernyőre

10 HIRES:HCLS

20 TYPE,1 – a méret beállítása

30 TEXTO,70,"TGBASIC" – kiírás

40 KEY A\$

50 LORES

RUN

A képernyőn dupla méretben a TGBASIC felirat tűnik fel. Egy billentyű megnyomásával visszatérhetünk a kisfelbontású módba.

Példa 2. pontvizsgálat

10 HIRES:HCLS:PLOT10,10,1

20 PRINTPNT(10,10)

30 LORES

Futtatás után a képernyőn 1 jelenik meg, mutatva, hogy a vizsgált pont tintaszínű.

Példa 3. kurzorvezérlés

10 PRINT CHR\$(147)

20 FOR N=0 TO 20

30 CURN,N**"

40 NEXT

Példa 4. karaktertervezés

0 CSET 3

10 CHAR0,0,!111000,!10000,!1111110,!10000,!10000,!1101100,0

15 ?CHR\$(147)

20 CUR5,5"@"

futtatás után a képernyőn egy emberke jelenik meg. A karaktergenerátort a RAM-ba másoltuk és ott a "@" helyére beírtuk az emberke alakját (a "@" képernyő-kódja 0).

bitek

sorok binárisan

7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	1
0	0	0	1	0	0	0	0	0
1	1	1	1	1	1	1	0	0
0	0	0	1	0	0	0	0	1
0	0	0	1	0	0	0	0	0
0	1	1	0	1	1	0	0	0
0	0	0	0	0	0	0	0	0

ÚJ NUMERIKUS ELEM BEOLV.
A típus jelző beállítása numerikusra
CHRGET

Kód>=\$60
CHRGET

A ROM rutinjára

Kód>=\$6a-->vissza a ROM rutinjára

A kód 0. bitjének leválasztása

A függvény véghajtása

Alapvető, elő, most nem kell. FRA

Zárójelek között valós szám beolvásása

Numerikus ellenőrzés

FAC-->ARG

INT(FAC)-->FAC

ARG-FAC-->FAC

("" beolv.

X koord.(0-21) beolv.

SCR/SCL

Elemente

,"" beolv.

Y koord.(0-22) beolv.

,"" beolv.

LO byte-ja.

\$bb--X koord.+az adott sor kezdetének

LO byte-ja.

\$bc--Az adott sor kezdetének

HI byte-ja.

SCR volt

A felső byte módosítása

Erték a FAC-ba

Érték a KONVERTALÓ FOGGVENYEK

A számjegyek száma

A szám értékét jelző

2 memóriacella

nullázása

CHRGET

0,1,...,9 a karakter

a,b,c,d,e,f a karakter

A beolvásott érték összehasonlítása a maximálissal,vége ha belolv.>max.

Az eddig beolvásott értéket balra forgatjuk.

Nem volt még meg minden számjegy

Az érték FAC-ba tevése

JOY/FIR

VIA#1,B adatirányregiszter

VIA#2,B adatirányregiszter

FIR volt.

\$9120 cím 7. bitje 0,ha balra

\$9111 cím 2. bitje 0,ha fel

\$9111 cím 3. bitje 0,ha le

\$9111 cím 4. bitje 0,ha jobbra

A regiszterek állapotának

visszaállítása

Y reg.-->FAC

\$9111 cím 5. bitje 0,ha a tűzgomb

le van nyomva

A koordináták elmentése

"(" beolv.

Koord. beolv.

Egy pont helyének kiszámítása

")," beolv.

A koordináták elhozatala

CPT volt.

\$d3a2

ldy #\$00

plp

ldi \$2bd8

ldi (\$d2),Y

and #\$b9

lsl \$b9

ldi \$2b71

and #\$b9

ldi \$2b71

TG BASIC

Példa 5.: többszínű karakter tervezése

Többszín módban (mind a finomgrafikus többszín módban, mind a karakteres többszín módban) a felbontás felére csökken, így viszont egy karakteren belül nem csak két (tinta és háttér) szín használható, hanem négy.

Itt jól használható a négyes számrendszerből kódoló függvény is. Írjuk le ábrácskánkat először a színekkel jelző számok formájában:

	3	2	1	0
0	3	3	3	3
1	2	2	2	2
2	1	1	1	1
3	0	0	0	0
4	0	0	0	0
5	1	1	1	1
6	2	2	2	2
7	3	3	3	3

Ezt az alakzatot programmal a következő módon állíthatjuk elő:

0 CSET 3
1 CHAR0,@3333,@2222,@1111,
@0000,@0000,@1111,@2222,
@3333

2 COLOR papírszín, 8+keretszín,
tinta+8 (multicolor be), kiegészítő
szín

4 ?"@"

5 ?" CTRL 7 " – visszakapcsolás
többszín módból a kék színre

Példa 6.: hang

10 VOL 15
20 FOR N=0 TO 127
30 SOUND 0,N
40 NEXT
50 OUT 0

Példa 7.: procedúra

10 PROC CIRCLE (x,y,r)
20 HIRES – grafika bekapcsolása
30 HCLS – képernyő törlése
40 FOR N=0 TO 2* STEP /180
50 PLOT SIN(N)*R+x,COS(N)*R+y,1 – rajzolás
60 NEXT

70 KEY A\$ – Vár, hogy megnyomunk egy gombot.
80 LORES – kikapcsolás
90 RETURN

Próba
CIRCLE(60,60,10)

Ha most megnyomunk egy gombot, akkor ismét az eredeti képernyőre kerülünk.

>2400 42 41 44 20 47 52 41 50 :bad grap Hibajelzés szövegek
>2408 48 49 43 20 4d 4f 44 c5 :hic mode
>2410 55 4e 44 45 46 27 44 20 :undef 'd
>2418 50 52 4f 43 45 44 55 52 :procedur
>2420 c5 :e

>2421 53 43 d2 53 43 cc 50 4e :scrsclpn ÚJ tokenszavak

>2429 d4 43 50 d4 4a 4f d9 46 :tcptjoyf

>2431 49 d2 46 52 c1 a1 a6 c0 :irfra!&@

>2439 50 4c 4f d4 48 43 4c d3 :plotcls

>2441 48 49 52 45 d3 43 55 d2 :hiressur

>2449 43 4f 4c 4f d2 49 ce 4f :colorino

>2451 55 d4 56 4f cc 53 4f 55 :utvolsou

>2459 4e c4 4c 4f 52 45 d3 4c :ndoresl

>2461 4f c3 4b 45 d9 45 4c 53 :ockeyels

>2469 c5 54 45 58 d4 54 59 50 :etexttyp

>2471 c5 44 45 cc 4f 4c c4 43 :edeloldc

>2479 47 4f 54 cf 50 52 4f c3 :gotoproc

>2481 43 53 45 d4 43 48 41 d2 :csetchar

>2489 52 45 50 45 41 d4 :repeat

>248f 00 2b

>2491 00 2b af 2b af 2b 78 2b

>2499 78 2b f0 2a 35 2b 35 2b

>24a1 35 2b 30 26 8e 25 08 25

>24a9 8e 26 6d 21 a1 26 a1 26

>24b1 b5 26 c2 26 1e 25 b5 21

>24b9 d8 21 b1 20 fc 27 ba 28

>24c1 c5 28 df 21 9b 21 f7 c8

>24c9 d8 27 23 2a cb 21

>24cf 71 26

>24d1 83 c4 0b 27 00 20 51 20

>24d9 d5 2a bf ea 75 2a 47 2a

>24e1 00 00

>24e3 00 10

>24e5 02 10 04 10 04 08

>24eb 93 2a 2a 2a 2a 20 54 47 :.**** tg

>24f3 42 41 53 49 43 20 50 4c :basic pl

>24fb 55 53 20 2a 2a 2a 2a 0d :us ****.

>2503 00

>2504 80 88

>2506 c0 c2 ce

ÚJ tokenszavak

Az utasítások és
függvények ugrási
címei



Vektorok

Felesleg

Hibajelzések kezdetének LO
byte-jai
Adatok a konv. függvényekhez

Bej. szöveg

Adatok a CSET-hez