

# DIDAKTIK GAMA



**OSOBNÝ MIKROPOČÍTAČ**  
**HC**  
**NÁVOD K OBSLUHE**

# Úvod

Didaktik Gama je osobný osembitový mikropočítač, konštrukčne odvodený od mikropočítača ZX SPECTRUM. Je konštruovaný na báze mikroprocesora Z 80.

Didaktik Gama je určený pre najširší okruh použitia - od oblasti hier a zábavy, cez výuku a programovanie, až po nasadenie pri jednoduchších konštruktérskych činnostiach. Je kompatibilný s mikropočítačom SINCLAIR ZX SPECTRUM programove a z veľkej časti i hardwareove. Zobrazuje v rastrí 256 x 192 bodov, ktoré sú usporiadané do 32 x 24 polí o veľkosti 8 x 8 bodov. Zobrazenie môže byť v ôsmich farbách v sústave PAL. V prípade použitia čierneho-bieleho televízneho prijímača je zobrazovaný motív v ôsmich jasových úrovniach. Mikropočítač je vybavený kontaktnou klávesnicou so 40 klávesmi.

Ako vonkajšiu pamäť pre programy a dáta je možné pripojiť bežný kazetový magnetofón, ktorý sa na rozdiel od ZX SPECTRUM nepripája na reproduktorový výstup, ale na bežný výstup magnetofónu. To umožňuje použiť aj kazetový magnetofón bez koncových zosilňovačov, napr. M 710A, SM 260 a pod. Priamy konektor mikropočítača Didaktik Gama je identický /až na napätia +9V, +12V, -12V a signál VIDEO/ s priamym konektorom mikropočítača ZX SPECTRUM. Okrem toho mikropočítač Didaktik Gama disponuje tridsaťpólovým konektorom FRB, ktorý dovoľuje priame pripojenie tlačiarne, súradnicového zapisovača, joysticku. Z hľadiska užívateľa predstavuje 23 programovateľných vstupno-výstupných liniek. Tento paralelný interface je osadený stykovým obvodom MHB 8255, ktorého jeden bit je systémovo využitý na programové prepínanie pa-

mátovej banky RAM s veľkosťou 32 kB. Celková kapacita pamäti RAM je 80 kB, pamäť ROM má rozsah 16 kB.

Súčasťou dodávky je sieťový napájací zdroj. Umožňuje pripojiť mikropočítač na bežnú sieť 220 V/50 Hz.

# Zostava mikropočítača

Pri zostave pracoviska s mikropočítačom Didaktik Gama postupujte nasledovne :

- prepojte VF výstup mikropočítača s anténnym vstupom TV prijímača, resp. VIDEO výstup so vstupom TV prijímača určeným pre magnetoskop /pri použití VIDEO výstupu docielite lepšiu kvalitu obrazu/
- prepojte zdroj mikropočítača s mikropočítačom a kazetovým magnetofónom nasledovne : konektor zo zdroja s jedným vodičom zapojíme do kazetového magnetofónu, konektor s dvoma vodičmi zapojíme do mikropočítača Didaktik Gama
- po vzájomnom prepojení zapnite TV prijímač a nalaďte približne požadovaný kanál /UHF pásmo kanál č. 48/
- zapojte zdroj Didaktik Gama do siete. Správna funkcia zdroja je indikovaná rozsvietením zelenej LED diódy na klávesnici mikropočítača /označená symbolom ~/
- po zahriatí mikropočítača na pracovnú teplotu /asi po 1 min./ stlačte tlačidlo RESET /označené na klávesnici res/
- vyladte obraz na TV prijímači tak, aby bol čitateľný text

© 1989 Didaktik Skalica

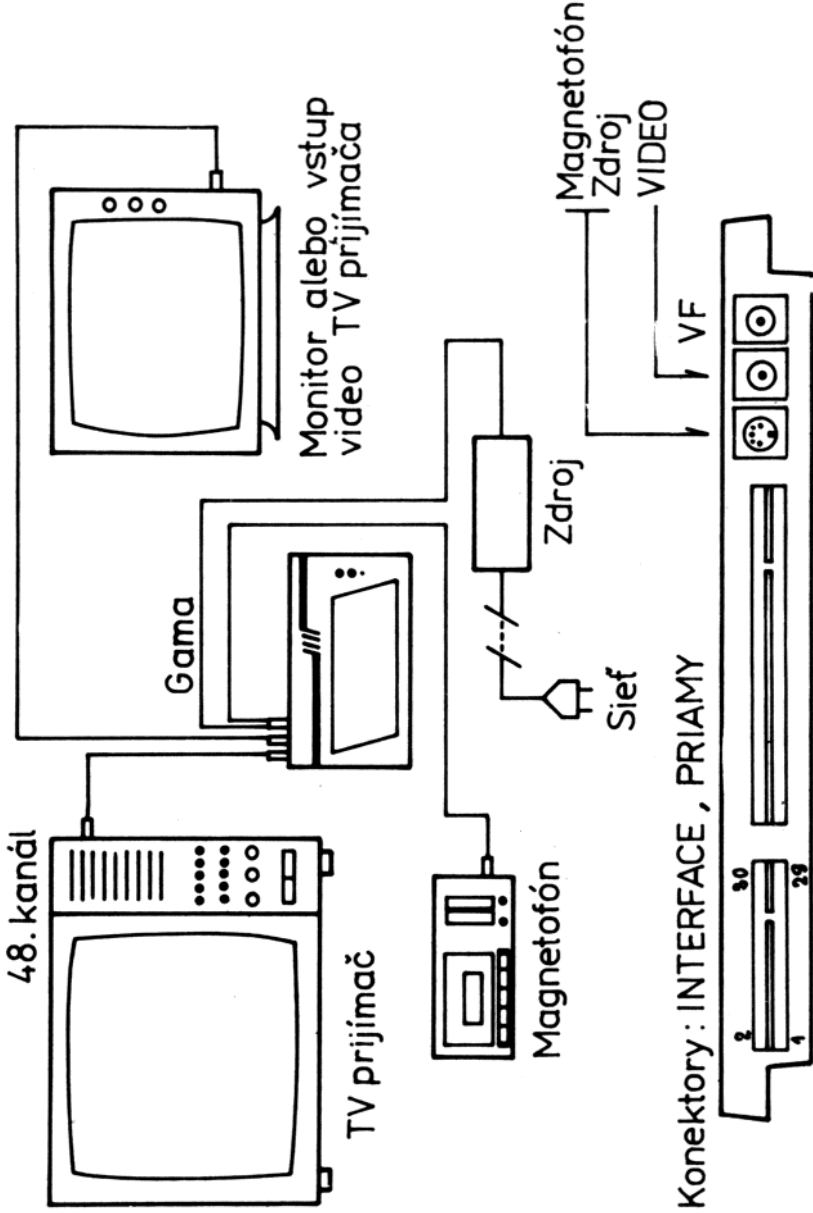
Ak je na obrazovke uvedený text, je mikropočítač pripravený k činnosti. V prípade, že tomu tak nie je a svieti zelená LED dióda, znova stlačte /prípadne i niekoľkokrát/ tlačidlo res.

Ak nesvieti po pripojení zdroja na sieť zelená dióda LED, je pravdepodobne prepálená sieťová poistka v zdroji.

## POZOR :

Pri výmene sieťovej poistky musí byť zdroj odpojený z napájacej siete !

Pre väčšiu názornosť je celá zostava s prepojením schematicky znázornená na obr. č. 1.



Obr. č. 1

# Obsluha klávesnice

Klávesnica mikropočítača Didaktik Gama je kontaktná a obsahuje 40 klávesov. Väčšina klávesov má niekoľko významov. Klávesnica je usporiadaná do štyroch riadkov. Okrem základných alfanumerických symbolov /číslo 0 až 9, písmená A až Z/ obsahuje príkazy a symboly programovacieho jazyka BASIC.

Z tohto hľadiska majú osobitný význam klávesy s označením CAPS SHIFT, SYMBOL SHIFT, BREAK SPACE, ENTER.

Mikropočítač Didaktik Gama obsahuje okrem týchto špeciálnych klávesov dva typy klávesov /podľa ich využitia/. Prvý typ sú klávesy obsahujúce v hornom rade číslice ako hlavný znak a druhý typ sú zvyšné klávesy obsahujúce ako hlavný znak písmená. Každý kláves /s výnimkou spomenutých špeciálnych/ umožňuje pri svojom stlačení použiť viac významov v závislosti od súčasného stlačenia klávesov CAPS SHIFT alebo SYMBOL SHIFT a od módu počítača, o ktorom informuje blikajúce písmeno na obrazovke /kurzor/. Počítač využíva klávesnicu v piatich módoch a sám ich volí podľa režimu činnosti, pričom niektoré módy môže voliť programátor z klávesnice. Jednotlivé módy a ich použitie ukážeme na príklade klávesu prvého typu obsahujúceho číslo 1 a klávesu druhého typu /písmeno Q/.

Mód K - očakáva počítač číslo riadku alebo kľúčové slovo jazyka BASIC, preto pri stlačení klávesu 1 sa objaví číslica 1 a pri stlačení klávesu Q príkaz PLOT /spodný biely/. Pri stlačení týchto klávesov so súčasným stlačením SYMBOL SHIFT sa objaví ! resp. <= , t. j. platí zásada, že spolu so SYMBOL SHIFT sa vypíšu znaky červenej farby vpra-

vo na klávese.

Mód L - mód pre číslice a písmená - objaví sa l resp. q. Pre písmená platí, že spolu so stlačením CAPS SHIFT možno písať veľké písmená. Pre červené symboly platí to isté, čo v režime K.

Režim C - je obdoba režimu L, teraz však píše veľké písmená rovnako pri súčasnom stlačení klávesu CAPS SHIFT ako aj bez neho. Prechod z režimu L do C a naspäť ovláda používateľ súčasným stlačením CAPS SHIFT a CAPS LOCK.

Režim E - dovoľuje využívať ďalšie symboly a príkazy /biele príkazy v hornej časti klávesu a červené v ľavej dolnej/. Do režimu E je možné prejsť súčasným stlačením klávesov CAPS SHIFT a SYMBOL SHIFT. V tomto režime pri stlačení klávesu druhého typu /písmená/ sa objaví príkaz vytlačený bielou farbou v pravom hornom rohu a pri súčasnom stlačení so SYMBOL SHIFT červený príkaz v ľavom dolnom rohu. Pre klávesy prvej skupiny /čísllice/ platí, že pri súčasnom stlačení so SYMBOL SHIFT sa vypíše červený symbol z ľavého dolného rohu, ale pri samostatnom stlačení klávesu sa objaví čierny štvorec - význam klávesu v tomto móde nie je definovaný.

Mód G - grafický - volí sa súčasným stlačením CAPS SHIFT a GRAPHICS a je určený pre zobrazenie grafických znakov /bielych/ na prvých ôsmich klávesoch horného radu. Pre všetky módy platí pre horný rad, že pri súčasnom stlačení týchto klávesov spolu s CAPS SHIFT sa vykoná príkaz umiestnený nad číslicou:

- 1 EDIT - režim editovania programu resp. vymazanie celého riadku na obrazovke /dialógového riadku/
- 2 CAPS LOCK - prechod z režimu L do C a naopak
- 3 TRUE VIDEO - neinvertovaný text

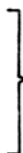
4 INV VIDEO - invertovaný text

5 ←

6 ↓

7 ↑

8 →



ovládanie pohybu kurzora a ukazovateľa

čísla riadku /programového kurzora/

9 GRAPHICS - prechod do režimu G

10 DELETE - mazanie napísaného symbolu nachádzajúceho sa vľavo od kurzora

Pre všetky klávesy platí zároveň funkcia autorepeat t.j. opakovaný výpis symbolu pri súvisle držanom stlačennom klávese. Význam klávesov CAPS SHIFT a SYMBOL SHIFT je zrejmý z predchádzajúceho a teraz vysvetlíme význam klávesov ENTER a BREAK SPACE.

ENTER - slúži na odoslanie napísaného riadku programu. V prípade, že riadok obsahuje syntaktickú chybu, nie je zobrazený v hornej časti obrazovky, ale zostáva v dialógovom riadku a otáznik signalizuje miesto syntaktickej chyby.

BREAK SPACE - slúži ako medzera pri vkladaní textov a spolu so stlačením CAPS SHIFT aj na prerušenie činnosti počítača pri vykonávaní programu. .

Zvláštny význam má tlačítko umiestnené v pravom hornom rohu klávesnice s označením res. Po jeho stlačení dochádza k hardwareovému resetu mikropočítača, t.j. k vymazaniu pamäti a nastaveniu východzieho stavu. Stlačenie tlačítka res má teda rovnaký účinok ako odpojenie a opätovné pripojenie napájacieho zdroja k mikropočítaču. Tlačítko je preto realizované tak, aby k jeho stlačeniu nemohlo prísť náhodne, ale je potrebné použiť pero, ceruzu a pod.



# Priamy režim činnosti

Priamy režim činnosti je charakterizovaný tým, že príkaz je zadávaný bez predchádzajúceho zadania čísla riadku. Po zadaní príkazu tlačítkom ENTER počítač tento hneď vykoná, pokiaľ to povaha príkazu dovoľuje. Nie všetky príkazy je totiž možné použiť v tomto režime práce. Napríklad po zadaní príkazu:

PRINT "DIDAKTIK"

a stlačením klávesu ENTER sa v hornej časti obrazovky objaví nápis

DIDAKTIK

Do spodného riadku obrazovky tzv. dialógového riadku vypíše počítač hlásenie

Ø OK, Ø:1

Uvedené hlásenie značí:

- Ø - hlásenie o chybe - nenastala žiadna chyba /zoznam hlásení o chybách je uvedený v nasledujúcom/
- OK - hlásenie počítača o ukončení činnosti
- Ø - číslo riadku, na ktorom bola činnosť ukončená
- 1 - poradie príkazu na danom riadku, vykonaním ktorého počítač ukončil svoju činnosť

V priamom režime činnosti je možné použiť mikropočítač vo funkcii kalkulačky nasledovným spôsobom:

po zadaní príkazu PRINT /kláves P/ je treba zadať tak ako na kalkulačke príslušné matematické operácie s použitím operátorov +, -, \*, / pomocou tlačítka SYMBOL SHIFT/ a ukončiť operáciu stlačením tlačítka ENTER. Napríklad:

PRINT 222 + 13

Po stlačení tlačítka ENTER sa na obrazovke objaví výsledok  
235 a hlásenie

Ø OK, Ø:1

# Programový režim činnosti

Ak začína riadok zadaný počítaču číslom, nevykoná sa hneď po jeho odoslaní klávesom ENTER, ale uloží sa do pamäti počítača ako programový riadok. Súčasne mikropočítač zadaný riadok vypíše na obrazovku. V prípade, že sa pri zápise programového riadku dopustíme syntaktickej chyby, počítač takýto riadok nezaradí do svojej pamäti a ohlásí chybu výpisom blikajúceho otáznika. Pri kontrole správnosti kontroluje mikropočítač zápis zľava, čo značí, že oznámi prvú chybu od začiatku riadku.

Výpis programových riadkov začína z horného ľavého rohu obrazovky s tým, že sa programové riadky vypisujú v postupnosti podľa rastúcich čísiel riadku. V prípade, že pri zápise programu zadáme riadok s číslom menším ako je posledný zadaný riadok, zaradí mikropočítač takýto riadok do postupnosti riadkov podľa jeho čísla. Programový riadok sa mikropočítaču zadáva stlačením klávesu ENTER. Posledný zadaný programový riadok je vo výpise na obrazovke označený symbolom > za číslom riadku. Symbol > budeme nazývať programový kurzor. V prípade, že chceme na jeden riadok zadať viacej príkazov, oddelíme jednotlivé príkazy dvojbodkou. Takýto riadok sa označuje ako viacnásobný príkazový riadok.

Príklad:

```
1Ø PRINT "SKALICA"
```

```
2Ø PRINT "1987"
```

```
3Ø PRINT "DIDAKTIK":PRINT "GAMA"
```

# Riadiace a pomocné príkazy

Riadiace a pomocné príkazy riadia činnosť mikropočítača a uľahčujú komunikáciu s ním. Dovoľujú odštartovanie programu, jeho výpis a úpravu, zastavenie a znovuspustenie ako aj jeho zmazanie. Z pohľadu ostatných príkazov teda zaujímajú osobitné postavenie. Sú to tieto príkazy: RUN, LIST, EDIT, NEW, STOP, CONT, REM, CLEAR, CLS, PAUSE.

Pri popise uvedených príkazov ako aj ostatných príkazov uvedených v ďalších častiach tejto publikácie budeme používať nasledovnú symboliku:

- < > - obsah týchto zátvoriek musí určiť užívateľ
- [ ] - z položiek uvedených v týchto zátvorkách si užívateľ môže vybrať jednu, resp. túto časť príkazu nemusí použiť
- { } - z položiek uvedených v týchto zátvorkách si užívateľ musí vybrať jednu

RUN

Syntax: RUN

prípadne syntax: RUN<N>

Príkaz RUN sa používa na odštartovanie programu, ktorý sa nachádza v pamäti mikropočítača. V prvom prípade sa program odštartuje od najnižšieho programového riadku, v druhom prípade sa odštartuje od riadku s číslom N. V prípade, že takýto riadok v programe neexistuje, odštartuje sa program od najbližšieho existujúceho vyššieho riadku.

Príklad:

```
1Ø PRINT "DIDAKTIK"
```

2Ø PRINT "GAMA"

3Ø PRINT "SKALICA"

Zadajte príkaz RUN, RUN 2Ø, RUN 25.

LIST

Syntax: LIST

prípadne:syntax: LIST<N>

Tento príkaz sa používa na výpis programu na obrazovku - vylistovanie programu. V prvom prípade počítač vypíše program od najnižšieho programového riadku, v druhom prípade od riadku s číslom N. V prípade, že je výpis dlhší ako je možné vypísať na jednu obrazovku, počítač výpis po jej zaplnení preruší a vypíše hlásenie

scroll?

Stlačením klávesu N sa výpis ďalšieho programu preruší, po stlačení ľubovoľného iného klávesu výpis pokračuje. Po prerušení výpisu programu vypíše počítač do dialógového riadku hlásenie

D BREAK - CONT repeats, Ø:1

a pri ukončení výpisu celého programu vypíše

Ø OK, Ø:1

EDIT

Slúži na editovanie programu vloženého do pamäti mikropočítača. Pri súčasnom stlačení CAPS SHIFT a EDIT sa objaví v dialógovom riadku ten riadok programu, na ktorý bol predtým nastavený programový kurzor / > /. Je možné ho ovládať stlačením CAPS SHIFT a ↑ resp. ↓ . Potom je možné pomocou tlačítiek CAPS HIPT a → resp. ← ovládať pohyb kurzora, robiť príslušné zmeny v programovom riadku a po ukončení editovania ho opäť odoslať stlačením klávesu ENTER.

## NEW

Syntax: NEW

Príkaz NEW sa používa na zmazanie programu, ktorý sa nachádza v pamäti mikropočítača. Po zadaní príkazu NEW zmaže počítač program z pamäti, o čom sa môžeme presvedčiť následným zadaním príkazu LIST.

## STOP

Syntax: STOP

Príkaz STOP sa používa na zastavenie vykonávania programu na príslušnom programovom riadku. Jeho použitie je vhodné, ak užívateľ potrebuje napríklad výsledky výpočtov odpísať z obrazovky. Po vykonaní príkazu STOP vypíše počítač do dialógového riadku hlásenie /ak bol príkaz STOP použitý na 40-tom riadku/

9 STOP statement, 40:1

## CONT

Syntax: CONT

Tento príkaz sa používa na znovuspustenie programu po predchádzajúcom zastavení napr. príkazom STOP, prípadne po zastavení vykonávania programu súčasným stlačením klávesov CAPS SHIFT a BREAK SPACE. Príkaz CONT je skrátenejší zápis príkazu CONTINUE, ktorý sa po stlačení klávesu CONT vypíše na obrazovku. Pokiaľ počítač pri zastavení vypíše správu začínajúcu číslom 9 resp. L, pokračuje program po zadaní príkazu CONT na ďalšom príkaze programu, inak zopakuje predchádzajúci príkaz.

## REM

Syntax: REM [ komentár ]

Príkaz REM sa používa na komentovanie časti programu. Pri

vykonávaní programu počítač komentár nachádzajúci sa za príkazom REM ignoruje. Vypíše ho len pri výpise programu /napr. príkazom LIST/.

CLS

Syntax: CLS

Tento príkaz sa používa na zotieranie obrazovky. Môže byť použitý buď v priamom alebo programovom režime práce mikropočítača.

CLEAR

Syntax: CLEAR

prípadne syntax: CLEAR<m>

Príkaz CLEAR zruší všetky premenné a zotrie obrazovku. Okrem toho vykoná príkaz RESTORE a nastaví pozíciu pre príkaz PLOT do ľavého dolného rohu obrazovky. V prípade, že je za príkazom adresa m, a pokiaľ je to možné, posúva systémovú premennú RAMTOP na adresu m. Nastaví nový zásobník pre GOSUB na novú hodnotu RAMTOP.

PAUSE

Syntax: PAUSE<n>

Príkaz PAUSE zastaví vykonávanie programu na dobu určenú hodnotou n. Pre n platí:

$$0 \leq n \leq 65535$$

Pre iné n ohlásí počítač chybu. Pre  $n = 0$  zastaví počítač vykonávanie programu až do stlačenia ľubovoľného klávesu. Ak  $n = 1$ , počítač zastaví vykonávanie programu na 0,02 sek. Vykonávanie príkazu PAUSE je možné zrušiť v priebehu programu stlačením ľubovoľného klávesu.

# Konštanty a premenné

Mikroočítač Didaktik Gama pracuje s reálnymi číslami z intervalu  $10^{-38}$  až  $10^{38}$  /kladnými aj zápornými/, ktoré sa môžu zadávať v priamom alebo semilogaritmickom tvare. Priamy tvar udáva znamienko a zápis čísla, pričom celá a desatinná časť sú od seba oddelené desatinnou bodkou, napr.:

227.14

-18.254

.3812

Semilogaritmický tvar je daný mantisou s jej znamienkom a exponentom s jeho znamienkom a označením E /exponent musí byť celé číslo z intervalu od -38 do +38/ napr.:

1ØE25

-22E4

2.3E-11

Pod pojmom premenná je treba rozumieť skupinu písmen a číslic začínajúcu písmenom. Je možné použiť aj medzeru, ale pri vnútornom spracovaní v mikroočítači nie je použitá. Premenná nesmie obsahovať znaky, ktoré nie sú písmenom ani číslicou, napr.: -, \*, a iné.

Pri zadávaní premennej nie je v jej názve rozdiel medzi malými a veľkými písmenami. Na priradenie konštanty pre zvolenú premennú sa používa príkaz LET, ktorý má syntax:

LET < premenná > = < výraz >

pričom výraz je buď konštantou, alebo sa vytvorí z konštant, premenných, funkcií a operátorov. Príklad:

1Ø LET A = 1Ø \* 1ØØØ

2Ø PRINT A

Po vykonaní programu sa objaví na obrazovke číslo 1ØØØØ.



# Funkcie numerických premenných

Veľmi výhodné pri práci s numerickými premennými je využívať zadefinované funkcie týchto premenných. Sú to jednak funkcie aritmetické, trigonometrické, generovanie náhodných čísiel a iné. Syntax goniometrických funkcií je:

SIN<X>

COS<X>

TAN<X>

určujú hodnotu goniometrických funkcií sínus, kosínus a tangens pre argument X zadaný v radiánoch.

ASN<X>

ACS<X>

ATN<X>

určujú hodnoty funkcií arkussínus, arkuskosínus, arkustangens parametra X, pričom ich výsledok je v radiánoch.

Pre matematické funkcie platí:

ABS<X> - udáva absolútnu hodnotu reálneho čísla X

SGN<X> - udáva znamienko reálneho čísla X. Hodnota tejto funkcie je 1, ak je argument kladný, -1 ak je argument záporný, 0 ak je argument X rovný nule.

INT<X> - udáva celočíselnú časť argumentu X

SQR<X> - udáva druhú odmocninu argumentu X, pričom X musí byť nezáporné číslo

LN<X> - udáva hodnotu prirodzeného logaritmu so základom  $e = 2.71828$ , pričom argument X musí byť väčší ako 0

EXP<X> - udáva hodnotu exponenciálnej funkcie  $e^x$  pre zadaný argument maximálne  $x = 88$

Funkcie pre generovanie náhodných čísiel

RND - generuje náhodné číslo z intervalu od 0 do 1 /táto funkcia nemá argument/. V skutočnosti ide o stálu postupnosť 65536 čísiel.

RAND [X] - umožňuje odštartovať funkciu RND v určitom mieste postupnosti danom argumentom X, ktorý môže byť od 1 do 65535.

V ďalšom definujeme špeciálne funkcie STR\$ a CHR\$.

STR\$<X> - táto funkcia mení číslo na reťazec a umožňuje potom preň používať funkcie reťazcových premenných, ktoré budú uvedené v ďalšom.

CHR\$<X> - udáva znak zodpovedajúci v ASCII kóde argumentu X /X je zadané dekadicky/.

Príklad:

```
PRINT CHR$ 77
```

na obrazovku vypíše M.

V prípade častejšieho používania niektorých ďalších funkcií, ktoré nepatria medzi uvedené štandardné funkcie, je možné definovať zvláštnu funkciu so svojim označením pomocou príkazu DEF FN.

```
DEF FN<meno funkcie ([formálny parameter, formálny parameter...])  
= výraz>- takto definovanú funkciu je možné potom v programe  
používať pomocou príkazu FN, ktorý má nasledovnú  
syntax:
```

```
FN<meno funkcie ([skutočný parameter, skutočný parameter ...])>
```

Príklad:

```
1) DEF FN s (x) = x * x
```

```
2) PRINT FN s (4)
```

Definovaná funkcia na výpočet plochy štvorca v danom prípade pre hodnotu parametra x = 4 vypíše na obrazovku výsledok 16.

## Práca s reťazcami

Okrem operácií s numerickými hodnotami je možné pracovať aj s reťazcami znakov. Reťazec je postupnosť znakov, ktorá sa spracováva ako jeden celok. Reťazcová konštanta je postupnosť písmen, číslíc a špeciálnych znakov ohraničená úvodzovkami. Reťazcová premenná sa označuje jedným písmenom /nie je rozdiel v označení veľkým alebo malým písmenom/ a symbolom  $\$$ .

Príklad:

```
1$ LET a$ = "DIDAKTIK"  
2$ PRINT a$
```

Pri práci s reťazcami je možné vykonávať spájanie /zlučovanie/ reťazcov pomocou operátora + a porovnávanie pomocou relačných operátorov.

Príklad:

```
1$ LET m$ = "DID"  
2$ LET n$ = "AKTIK"  
3$ PRINT m$ + n$  
4$ IF m$ > n$ THEN PRINT "Gama"
```

Pre prácu s reťazcami sú určené tieto funkcie:

LEN<X\$> - táto funkcia určuje dĺžku reťazca, t. j. počet znakov, z ktorých je reťazec zložený /vrátane medzier/.

Príklad:

```
1$ LET z$ = "DIDAKTIK GAMA"  
2$ PRINT LEN z$
```

Počítač vypíše na obrazovku číslo 13.

VAL<X\$> - premieňa reťazec na číslo a dáva možnosť pracovať s ním pomocou aritmetických operátorov a funkcií.

Príklad:

```
1$ LET N$ = "100"
```

```
2$ PRINT VAL N$
```

Počítač vytlačí číslo 100.

VAL \$ <"X\$" > - funguje podobne ako VAL, ale výsledok funkcie  
je považovaný za reťazec.

Príklad:

```
1$ LET K$ = "SYMBOL"
```

```
2$ PRINT LEN VAL$ "K$"
```

Vytlačí počet znakov reťazca K\$, t.j. číslo 6.

# Aritmetické výrazy

Aritmetické výrazy sú zložené z konštánt, premenných, funkcií a operátorov. Počítač tieto výrazy vyhodnotí a ich hodnotou je číslo. Konštanty, premenné a funkcie boli už definované a prípustné operátory používané pri práci s mikro-počítačom delíme na aritmetické a relačné.

Aritmetické operátory:

+ sčítanie čísiel

- odčítanie, záporné znamienko

\* násobenie

/ delenie

† umocňovanie /argument musí byť kladné číslo/

Relačné operátory:

= rovnosť

> väčší ako

< menší ako

>= väčší alebo rovný

<= menší alebo rovný

<> nerovnosť

Logické výrazy sú zložené z konštánt, premenných, funkcií a logických operátorov. Najčastejšie vznikajú spojením aritmetických výrazov, pričom sa vyhodnocuje pravdivosť celku a výsledkom je logická 0 - logický výraz je nepravdivý, alebo logická 1 - výraz je pravdivý. Na spojenie aritmetických výrazov je možné použiť tri základné logické operátory AND, OR, NOT, ktorých funkcia je nasledovná:

AND /logický súčin/ - výsledný logický výraz je pravdivý, ak sú všetky výrazy /spojené navzájom ope-

rátorom AND/ pravdivé.

OR /logický súčet/ - výsledný logický výraz je pravdivý, ak aspoň jeden z výrazov spojených operátorom OR je pravdivý.

NOT /negácia/ - výraz je pravdivý vtedy, ak pôvodný výraz bol nepravdivý.

Uvedieme usporiadanie jednotlivých operácií podľa ich priority:

- 1./ umocňovanie
- 2./ prosté mínus /mínus ako negácia, resp. záporné znamienko/
- 3./ násobenie a delenie
- 4./ sčítanie a odčítanie
- 5./ operácie porovnávania /rovnosť, nerovnosť, menší ako, väčší ako/
- 6./ negácia - NOT
- 7./ logický súčin - AND
- 8./ logický súčet - OR

Je zrejmé, že použitie zátvoriek má prednosť pred vyhodnocovaním priority jednotlivých operácií.

# Polia

Pole je rad premenných nesúcich rovnaké meno a líšiacich sa len indexom /číslo v zátvorke za menom/. Je to teda skupina indexovaných premenných. Pred použitím poľa je nutné rezervovať preň miesto v pamäti počítača použitím príkazu DIM. DIM<X (rozmer poľa)> - kde X je meno poľa a v zátvorke je určenie jeho rozmeru /pole môže byť jedno i viac rozmerné /.

Príklad:

```
DIM A (10,20)
```

takto je definované pole A, ktoré má rozmery 10 \* 20, t. j. od premennej A/1,1/až po premennú A/10,20/.

Je možné definovať rovnakým menom premennú aj pole - líšia sa indexom použitým v prípade poľa.

Existujú tiež reťazcové polia, pričom reťazec a reťazcové pole nemôžu mať rovnaké meno. Na príklade si ukážeme definíciu reťazcového poľa. Pole A\$ obsahuje 10 reťazcov, z ktorých každý má maximálnu dĺžku 20 znakov a je definované nasledovným príkazom:

```
DIM A$ (10,20)
```

potom napr. A\$/5,10/ je desiaty znak piateho reťazca A\$/5/.

Príklad:

```
10 DIM A$ (10,20)
```

```
20 LET A$ (4) = "ABCDEFGH"
```

```
30 PRINT A$ (4,2)
```

Tento program vypíše na obrazovku písmeno B.

# Výpis na obrazovku

Príkaz PRINT, ako už bolo uvedené pri popisovaní priameho režimu mikropočítača, umožňuje vypísať na obrazovku hodnoty premenných, konštanty, text alebo výsledky matematických operácií /teda hodnoty výrazov/. Pri podrobnejšej analýze možností využitia tohto príkazu popíšeme každú variantu zvlášť vrátane jej syntaktického tvaru.

PRINT [ "text" ] - počítač vypíše na obrazovku zadaný text

PRINT [ zoznam výrazov ] - počítač vypíše hodnotu výrazu zhora do pracovnej časti obrazovky. V prípade použitia samotného príkazu PRINT bez zoznamu výrazov vynechá jeden riadok.

Príklad:

1Ø LET A = 1ØØ

2Ø PRINTA

3Ø PRINT

4Ø PRINT "nad týmto textom je vynechaný jeden riadok"

PRINT [ výraz, výraz, ... ] - jednotlivé položky zoznamu výrazov sú od seba oddelené čiarkami. Počítač vypíše ďalší prvok zoznamu do druhej polovice obrazovky - vytvára teda pri viacerých položkách oddelených čiarkami vlastne dva stĺpce.

PRINT [ výraz; výraz; ... ] - položky sú od seba oddelené bodkočiarkou - počítač ich vypisuje tesne vedľa seba. Tento spôsob vý-



pisu umožňuje zlúčiť napr. výpis viacerých refazcových premenných.

**PRINT [ výraz'výraz' ... ]** - oddelenie položiek apostroфом zabezpečí vytlačenie každej položky na nový riadok.

**PRINT TABX; [zoznam výrazov]** - umožňuje zvoliť stípec, od ktorého začne výpis, pričom platí, že X musí byť celé číslo a počítateľ ho redukuje modulo 32 /delí 32 a berie zvyšok/, čiže TAB 34 znamená to isté ako TAB 2. Zároveň pre výpis platí, že ak neskončil na danom riadku, pokračuje na ďalšom už od začiatku riadku.

**PRINT AT X,Y; [zoznam výrazov]** - kde X je od 0 po 21 a udáva riadok /zhora nadol/ a Y od 0 po 31 udáva stípec /zľava doprava/, na ktorom sa začne výpis daného výrazu.

Príklad:

**PRINT AT 10, 20; "L"**

vypíše písmeno L do 10 riadku a 20 stĺpca pracovnej časti obrazovky.

**PRINT # 0; [zoznam výrazov]** - umožňuje umiestnenie výpisu do 22 riadku, ktorý pri bežnom používaní príkazu PRINT je zakázaný.

**PRINT # 1; [zoznam výrazov]** - umožňuje výpis do 23 riadku. Pre obidva predchádzajúce príkazy platí, že pri praktickom využití treba za ne zaradiť príkaz PAUSE

/najlepšie PAUSE #/, lebo inak bude výpis premazaný hlásením počítača v dialógovom riadku.

Existuje opačná funkcia k funkcii definovanej príkazom

PRINT AT, ktorá má nasledovnú syntax:

SCREEN# (X,Y) - pre parametre X a Y platia rovnaké podmienky ako pri definícii PRINT AT. Táto funkcia určí, aký znak je vytlačený na pozícii danej parametrom X, Y.

# Zadávanie, vstup a výstup dát

Pre zadávanie údajov do počítača a ich výstup existuje niekoľko štandardných príkazov, ktoré teraz uvedieme:

**INPUT<zoznam premenných>** - tento príkaz slúži na vkladanie vstupných údajov z klávesnice mikropočítača počas vykonávania programu. Na riadku, na ktorom je umiestnený príkaz **INPUT**, beh programu zastane, objaví sa vľavo dole blikajúci kurzor a počítač očakáva dáta odoslané tlačítkom **ENTER**. Ak je zoznam premenných zložený z viacerých položiek, musia byť v príkaze **INPUT** oddelené čiarkou. Po odoslaní prvého čísla sa posunie kurzor do polovice dialógového riadku a počítač očakáva ďalšie číslo. Po ukončení zadávania vstupných dát pokračuje vykonávanie programu. V zozname premenných sa môžu vyskytovať aj reťazcové premenné, pričom im zodpovedajúci zadávaný vstupný údaj musí byť reťazec. Ak počítač pri vyhodnocovaní príkazu **INPUT** narazí na reťazcovú premennú, prejaví sa to tým, že vloží blikajúci kurzor na obrazovke pri očakávaní dát do úvodzoviek.

Príklad:

1) INPUT A\$, B

2) PRINT A\$; B

Je možné používať príkaz INPUT zároveň s použitím výpisu textu do dialógového riadku ako to ukazuje nasledovný príklad:

1) INPUT "zadaj svoje meno"; A\$

2) PRINT "moje meno je"; A\$

INPUT LINE <X\$> - pracuje rovnako ako INPUT v prípade reťazcovej premennej, ale nevloží kurzor pri očakávaní zadávaného reťazca do úvodzoviek. Nie je možné používať príkaz INPUT LINE pre numerické premenné.

Pre vkládanie údajov sa najčastejšie používa dvojica príkazov DATA, READ, pre ktoré platí:

DATA <zoznam konštánt> - slúži na zadanie konštánt do počítača, pričom jednotlivé konštanty sú od seba oddelené čiarkami.

READ <zoznam premenných> - načítava údaje z príkazu DATA a v poradí v akom sú uvedené ich priraďuje k jednotlivým premenným oddeleným od seba čiarkami.

Príkaz DATA môže byť umiestnený v programe na ľubovoľnom mieste, príkaz READ priraďuje konštanty v poradí podľa jednotlivých príkazov DATA /od najnižšieho čísla riadku/ a v každom príkaze DATA zľava doprava. Jeden príkaz READ môže teda používať viac príkazov DATA, ale počet konštánt v príkazoch DATA nesmie byť menší ako počet premenných obsiahnutých v príkaze READ. Ak je však počet vložených konštánt väčší ako počet premenných, zostávajú zvyšné konštanty nevyužitú.

Príklad:

```
1Ø DATA 5,1Ø,15,2Ø
2Ø READ A,B,C,D,E,F
3Ø PRINT A,B,C,D,E,F
4Ø DATA 25,3Ø,35,4Ø
```

Tento program vypíše na obrazovku nasledovné čísla:

```
5      1Ø
15     2Ø
25     3Ø
```

RESTORE [ X ] - umožňuje načítavať tie isté konštanty vložené príkazom DATA na riadku s číslom X. Ak nebude v príkaze RESTORE uvedené číslo riadku, vykoná skok na DATA uvedené na najnižšom riadku v programe.

Príklad:

```
1Ø DATA 5,1Ø,15,2Ø
2Ø READ A,B,C,D
3Ø RESTORE 1Ø
4Ø READ E,F
5Ø DATA 1ØØ,2ØØ
6Ø PRINT A,B,C,D,E,F
```

Tento program vypíše na obrazovku čísla

```
5      1Ø
15     2Ø
5      1Ø
```

Ďalšia možnosť vstupu a výstupu je čítanie a zápis na adresovateľné miesta, ktorých je 65536 a nazývajú sa miesta I/O. Na ich ovládanie sa používajú príkazy IN a OUT.

IN<adresa> - výsledkom je byte načítaný z pozície, ktorú udáva zvolená adresa

OUT <adresa,hodnota> - zapíše danú hodnotu na zvolenú adresu.

Keďže klávesnica mikropočítača je rozdelená na 8 polovíc rád po piatich tlačítkach, je možné použiť príkaz IN na ich načítanie, pričom vstupné adresy sú:

65278 od CAPS SHIFT po V  
65022 od A po G  
64510 od Q po T  
63486 od 1 po 5  
61438 od 0 po 6  
57342 od P po Y  
49150 od ENTER po H  
32766 od BREAK SPACE po B

INKEY% - táto funkcia nemá argument a používa sa na čítanie klávesnice. Nasledujúci príklad dáva možnosť pracovať s mikropočítačom ako s písacím strojom /po stlačení tlačítka sa vypíše znak resp. kľúčové slovo v móde L/:

```
10 IF INKEY% <> "" THEN GO TO 10  
20 IF INKEY% = "" THEN GO TO 20  
30 PRINT INKEY% ;  
40 GO TO 10
```

# Práca s magnetofónom

Zadávanie programu z klávesnice je pomerne zdĺhavé, časovo náročné a vyskytujú sa pri ňom časté omyly. Z uvedených dôvodov je vhodné vytvorený program nahráť na bežnú magnetofónovú kazetu, aby v prípade potreby bolo možné ho opätovne nahráť späť do počítača. K spolupráci mikropočítača a magnetofónu sú určené príkazy **SAVE**, **LOAD**, **VERIFY**, **MERGE**.

## SAVE

Existuje niekoľko možností zápisu tohto príkazu, ktoré si pre väčšiu názornosť rozoberieme samostatne. Vo všeobecnosti však platí, že príkaz **SAVE** sa používa na nahratie programu a premenných na magnetofónovú kazetu.

Syntax: **SAVE** "<komentár>"

Komentár je prakticky identifikačným názvom programu, ktorý slúži na jeho neskoršie vyhľadávanie na páske. Môže obsahovať maximálne 10 znakov, písmen, prípadne číslíc. Po zadaní príkazu **SAVE** a jeho odoslaní vypíše mikropočítač správu

Start tape, then press any key

čo značí, že užívateľ má spustiť magnetofón, /predtým zapojený do režimu nahrávania/ a potom stlačiť ľubovoľný kláves. Po jeho stlačení sa na obrazovke objavia vodorovné pásy hrubé asi 1 cm a po asi 5 sek. sa zmenia na pásy zhruba polovičné. Počítač nahráva program po častiach. Koniec nahrávania je signalizovaný hlásením **OK**.

## Upozornenie!

Počítač nekontroluje správnosť obsluhy kazetového magnetofónu. To značí, že po skončení nahrávania vypíše správu

OK aj napr. v prípade, že nebude kazetový magnetofón pripojený. Tak isto je potrebné nastaviť správnu úroveň nahrávania, prípadne zapojiť obvody automatického nahrávania. Pri obsluhu kazetového magnetofónu je potrebné sa riadiť pokynmi výrobcu.

Ďalšou možnosťou zápisu príkazu SAVE je:

Syntax: SAVE " <komentár> " LINE m

Pri takomto zápise programu na kazetu počítač zapíše informáciu o tom, od ktorého čísla riadku /m/ sa má po spätnom nahrať programu do počítača tento automaticky odštartovať. Po načítaní takto zapísaného programu ho počítač automaticky odštartuje od čísla riadku m.

Pre záznam poľa na magnetofónovú kazetu sa používa príkaz:

Syntax: SAVE " <komentár> " DATA <meno poľa> ( )

Meno poľa v príkaze špecifikuje pole a môže byť tvorené písmenom alebo písmenom a značkou \$. Po nahrať dátového poľa do pamäti mikropočítača je potrebné pamätať na to, že počítač nezruší len identické znakové pole s daným menom, ale vymaže aj reťazec s daným menom.

Pre nahrať údaje na obrazovke počítača slúži príkaz:

Syntax: SAVE " <komentár> " SCREEN\$

Uvedený príkaz nahrá na magnetofónovú kazetu obsah obrazovky mikropočítača, ktorý je práve aktuálny pred odoslaním tohto príkazu.

Poslednou modifikáciou príkazu SAVE je tvar:

Syntax: SAVE " <komentár> " CODE <a,d>

Uvedený príkaz nahrá na magnetofónovú kazetu obsah pamäti mikropočítača od adresy a v dĺžke d bytov. Adresa a môže byť v rozsahu 0 až 65535. Napríklad obsah obrazovky môžeme na mag-



netofónovú kazetu okrem predchádzajúceho príkazu nahráť aj príkazom

SAVE "obraz" CODE 16384,6912

**Poznámka:**

Po skončení nahrávania na kazetu vypíše mikropočítač správu OK. Toto hlásenie značí, že mikropočítač ukončil záznam. Nie je to však znakom toho, že príslušný program resp. data sú na kazetu zapísané správne. Z tohto dôvodu je vhodné po zapísaní programu na kazetu preveriť správnosť zápisu príkazom VERIFY.

**VERIFY**

Tento príkaz sa používa na kontrolu zapísaného programu na kazete. Počítač pritom kontroluje program a premenné na kazete s tými, ktoré sú uložené v pamäti mikropočítača. Príkaz VERIFY je možné použiť pre všetky doteraz popísané možnosti zápisu príkazu SAVE okrem prípadu, kedy nahrávame obsah obrazovky. V tomto prípade príkaz VERIFY ohlásí chybu, nakoľko pri verifikácii programu vypíše na obrazovku jeho meno a tým vlastne zmení obsah obrazovky voči pôvodnému obsahu na páske. Príkaz VERIFY môže mať teda nasledovné zápisy:

Syntax: VERIFY " [komentár] "

VERIFY " [komentár] " CODE a,d

VERIFY " [komentár] " DATA <meno poľa> ( )

Príkaz: VERIFY " [komentár] " SCREEN\$ je síce možné použiť, ale bude vždy hlásiť chybu. Symbolika v jednotlivých príkazoch je identická so symbolikou v príkaze SAVE. Odlíšnosť je v tom, že ak za príkazom VERIFY neudáme do úvodzoviek názov záznamu ako komentár, počítač skontroluje prvý záznam na magnetofónovej kazete, ktorý prečíta. Príkaz má tvar:

VERIFY " "

Druhá odlišnosť je pri príkaze

VERIFY " [ komentár ] " CODE a,d

Ak je program na kazete dlhší ako udáva dĺžka d, ohlásí počítač chybu R. Počítač kontroluje program vo svojej pamäti od adresy a. Tento základný tvar príkazu má niekoľko modifikácií:

Syntax: VERIFY " [ komentár ] " CODE a

prípadne

Syntax: VERIFY " [ komentár ] " CODE

V týchto prípadoch počítač kontroluje záznam podľa hodnôt a, d, ktoré boli zadané pri príkaze SAVE, teda pri zápise príslušného programu na pásku.

LOAD

Podobne ako u príkazov SAVE a VERIFY existuje niekoľko možností zápisu príkazu LOAD.

LOAD " <komentár> "

Príkaz nahrá z magnetofónu program s názvom určeným komentárom. Pri použití príkazu LOAD dochádza k prepísaniu programu uloženého v pamäti počítača programom z kazety.

Ak sa názov programu na páske nezhoduje s názvom zadaným v príkaze LOAD(komentár), počítač vypíše po prečítaní hlavičky záznamu jeho názov na obrazovku, ale program sa do pamäti nenahrá. Pri použití príkazu

LOAD ""

sa do pamäti mikropočítača nahrá prvý program, ktorý je na páske zaznamenaný. Ak pri nahrávaní prišlo k chybe, počítač vypíše správu

Tape loading error

a nahrávanie sa zastaví. Ak prenos dát prebehol bez chyby,

po nahratí programu vypíše správu

OK

Pri použití príkazu SAVE " <komentár> " LINE m pri nahrávaní programu na kazetu sa program po bezchybnom nahratí automaticky odštartuje od riadku m.

LOAD " <komentár> " CODE

Počítač nahrá do pamäti dáta, ktoré sú uložené od adresy špecifikovanej v hlavičke záznamu.

LOAD " <komentár> " CODE a

Počítač nahrá do pamäti dáta a uloží ich od adresy a.

LOAD "" SCREEN\$

Do oblasti VIDEO RAM sa nahrá obrázok uložený na kazete.

LOAD " <komentár> " DATA meno poľa

Príkaz nahrá do pamäti obsah poľa /numerického alebo reťazcového/. Príkaz ruší predchádzajúcu deklaráciu poľa príkazom DIM, ak bola predtým použitá.

Ak by sme chceli za program v pamäti počítača uložiť program z kazety bez toho, aby sme pôvodný program zmazali použijeme príkaz MERGE. Pomocou tohto príkazu môžeme vložiť do pamäti z kazety niekoľko programov. Podmienkou však je, aby sa v jednotlivých programoch nevyskytovali rovnaké čísla riadkov. Pre jeho použitie platí nasledovná syntax:

MERGE " <komentár> "

# Cyklus

Pre vykonávanie cyklu je určená skupina navzájom súvisiacich príkazov FOR, TO, STEP, NEXT.

Syntax: FOR <premenná> = <výraz 1> TO <výraz 2> STEP[<výraz 3>]  
NEXT <premenná>

Premenná musí byť numerická a jednoznaková /len jedno písmeno/. Výraz 1 udáva začiatočnú a výraz 2 koncovú hodnotu premennej, s ktorou sa budú vykonávať operácie umiestnené na riadkoch medzi príkazmi FOR a NEXT, pričom po dosiahnutí príkazu NEXT sa hodnota premennej zvýši o hodnotu výrazu 3, až kým nepresiahne koncovú hodnotu /výraz 2/. Ak nie je použitý príkaz STEP, mení sa premenná o hodnotu 1.

Príklad:

```
1Ø FOR t=Ø TO 1Ø STEP.5  
2Ø PRINT t  
3Ø NEXT t
```

Po skončení cyklu pokračuje program ďalším riadkom za príkazom NEXT. Pri používaní príkazov cyklu je možné umiestniť jeden cyklus vo vnútri druhého, pričom musí byť celý vnútorný cyklus uložený vo vnútri vonkajšieho.

Príklad:

```
1Ø FOR a=Ø TO 2  
2Ø FOR b=Ø TO 9  
3Ø PRINT a;b  
4Ø NEXT b  
5Ø NEXT a
```

## Vetvenie programu

Pri programovaní je veľmi výhodné použiť možnosť skoku v programe, t. j. nedodržať vykonávanie programu postupne podľa čísiel riadkov, ale umožniť v programe podmienené alebo nepodmienené skoky.

Syntax: GO TO<číslo riadku>

Príkaz nepodmieneného skoku, pri vykonaní ktorého počítač uskutoční skok na riadok s daným číslom a pokračuje ďalej v postupnom vykonávaní programu od tohto riadku. Skok sa môže vykonať na riadok s vyšším alebo nižším číslom ako je číslo riadku, na ktorom je umiestnený príkaz GO TO.

Pri použití podmieneného skoku je definovaný nasledovný syntaktický tvar:

Syntax: IF<podmienka> THEN<príkaz>

Podmienka je daná aritmetickým, relačným alebo logickým výrazom. Pri splnení podmienky počítač vykoná príkaz umiestnený za príkazom THEN. Ak podmienka nie je splnená, pokračuje počítač vo vykonávaní programu od nasledovného riadku.

## Podprogramy

Ak programátor používa niektorú časť programu v programe viackrát, má možnosť túto časť definovať ako podprogram a používať ju pomocou príkazov GOSUB a RETURN.

Syntax: GOSUB <n>

kde n je číslo riadku, na ktorom začína podprogram. Tento príkaz zabezpečí priamy skok na začiatok podprogramu.

Syntax: RETURN

tento príkaz ukončuje podprogram a vracia beh programu na riadok nasledujúci za použitým príkazom GOSUB. Nie je možné použiť príkaz RETURN bez príkazu GOSUB.

Nasledovný príklad umožňuje pomocou podprogramu uskutočniť výpočet štvrtej odmocniny.

Príklad:

```
1Ø FOR X = 1 TO 2Ø
2Ø GO SUB 1ØØ
3Ø PRINT X,Y
4Ø NEXT X
5Ø STOP
1ØØ LET Z = SQR ( X )
11Ø LET y = SQR ( Z )
12Ø RETURN
```

# Použitie grafických príkazov

V tejto kapitole popíšeme možnosti, ktoré poskytuje Didaktik Gama pri kreslení na obrazovku. Pracovná časť obrazovky má 22 riadkov a 32 stĺpcov, t.j.  $22 \times 32 = 704$  znakov. Každý znak je vytvorený z  $8 \times 8$  štvorcových bodov nazývaných pixel /obrazový element/. Každý pixel je definovaný x-ovou a y-ovou súradnicou, ktoré sa píše ako dvojica  $/x,y/$ . Rohové body tej časti obrazovky, do ktorej je možné kresliť, majú súradnice  $/0,0/; /0,175/; /255,0/; /255,175/$ , z čoho je zrejmé, že súradnica x sa môže meniť od 0 po 255 a súradnica y od 0 po 175. Pre kreslenie na obrazovku sa používajú 4 príkazy: PLOT, DRAW, CIRCLE, POINT, ktoré v nasledujúcom vysvetlíme.

Syntax: PLOT  $\langle x,y \rangle$

Tento príkaz zabezpečí na obrazovke vykreslenie bodu so súradnicami  $x,y$ . Pomocou tohto príkazu je možné nakresliť napr. graf funkcie sínus pre hodnoty od 0 po  $2\pi$ /PLOT pracuje v absolútnych súradniciach/.

Príklad:

```
10 FOR m=0 TO 255 STEP.5
20 PLOT m,88 + 50 * SIN (m/50 * PI)
30 NEXT m
```

Syntax: DRAW  $\langle x,y \rangle$

Tento príkaz kreslí vektor so súradnicami  $x,y$  z bodu, kde skončil posledný grafický príkaz. Na rozdiel od príkazu PLOT súradnice  $x,y$  v príkaze DRAW môžu byť aj záporné, čo vyplýva z toho, že príkaz DRAW používa relatívne súradnice, t.j. bod, v ktorom skončila posledná priamka zadaná príkazom DRAW, má pre ďalšiu priamku a teda ďalší príkaz DRAW súradnice

Ø,Ø. Príkaz DRAW je možné použiť aj na kreslenie časti kružnice, potom je nutné použiť tento syntaktický tvar:

Syntax: DRAW<x,y,m>

kde x,y sú súradnice koncového bodu časti kružnice a m udáva otáčanie v radiánoch - pre kladné m sa otáča doľava a pre záporné doprava. Parameter m vlastne určuje dĺžku časti kružnice od bodu, ktorého súradnice boli nastavené predchádzajúcim grafickým príkazom do bodu x,y a tým vlastne určuje jej zakrievanie.

Príklad:

1Ø PLOT 1ØØ,1ØØ

2Ø DRAW 5Ø,5Ø,2

Ak chceme kresliť celé kružnice, môžeme využiť ďalší príkaz CIRCLE, ktorého syntaktický tvar je:

Syntax: CIRCLE<x,y,r>

Kreslí kružnicu so stredom v bode určenom súradnicami x,y a polomerom r.

Príklad:

1Ø PLOT 1ØØ,Ø

2Ø DRAW Ø,5Ø

3Ø CIRCLE 1ØØ,5Ø,5Ø

Syntax: POINT<x,y>

Príkaz určuje, či bod so súradnicami x,y je zobrazený /svieti/ na obrazovke. Ak bod svieti, jeho hodnota je 1, ak nesvieti, tak je Ø.



# Príkazy pre ovládanie farieb

Mikropočítač Didaktik Gama pracuje v 8 farbách. Obrazovka je rozdelená na dve časti: okrajová časť /lemovanie obrazovky/ a stredná pracovná časť. Farby oboch častí je možné meniť programovo alebo priamo z klávesnice. Jednotlivé farby sú uvedené na klávesoch s číslami 0 až 8 v hornom riadku klávesnice v tomto poradí:

0 čierna

1 modrá

2 červená

3 purpurová

4 zelená

5 bledomodrá

6 žltá

7 biela

Po zapnutí pracuje počítač čiernobielo. Na zmenu farieb používame tri príkazy BORDER, PAPER, INK.

Syntax: BORDER<x>

kde x môže byť celé číslo od 0 po 7, pričom každé číslo znamená zvolenú farbu podľa uvedeného zoznamu a určuje farbu okrajovej časti /lemovania/obrazovky.

Syntax: PAPER<f>

Pre parameter f platí, že ním môže byť celé číslo od 0 do 9 a v intervale 0 až 7 určuje farbu podkladu pracovnej časti obrazovky. V prípade použitia príkazu PRINT za príkazom PAPER bude text vytlačený na zvolenom farebnom podklade, ale zvyšná časť obrazovky bude mať nezmenenú predchádzajúcu

farbu. Ak za príkazom PAPER použijeme príkaz CLS, hneď po jeho vykonaní sa vyfarbí podklad celej pracovnej časti obrázovky. Ak má parameter f hodnotu 8, zostane farba nezmenená. Použitie parametra f = 9 umožňuje definovať farbu kontrastnú proti predchádzajúcej.

Syntax: INK <f >

Tento príkaz umožňuje voliť farbu písmen, pričom pre parameter f platia tie isté podmienky ako v príkaze PAPER. Aj po skončení programu zostáva v platnosti farba definovaná posledným príkazom INK. Tento príkaz sa dá použiť aj v spojení s grafickými príkazmi napr. príkaz PLOT INK f; x,y vyfarbí bod so súradnicami x,y zvolenou farbou podľa parametra f.

# Ovládanie jasových úrovní

Pri práci s mikropočítačom je veľmi užitočné pri zobrazovaní na obrazovku využívať možnosti zvýšenia jasú, blikania, prepisovania a inverzného výpisu.

Syntax: FLASH <t>

Umožňuje blikanie obrazu. Parameter t musí byť 0 alebo 1.

Ak je 0, tak obraz neblíkajú, a ak sa t zmení na 1, bude prísľušný text blikať až do zmeny parametra t príkazom FLASH.

Príklad:

```
1 0 PRINT FLASH 1; "text"
```

```
2 0 PRINT AT 10,10; FLASH 1; "t"
```

Syntax: BRIGHT <t>

Určuje zvýraznenie jasú, pričom ak t = 0, je jas normálny a pre t = 1 je jas zvýšený. Tento príkaz môžeme kombinovať s príkazmi určujúcimi farbu.

Syntax: OVER <t>

Umožňuje vpísať dva texty do seba, ak je t = 1. Ak je t = 0 platí pre zápis pomocou príkazu PRINT, že najprv zotriepodvodný text a potom na jeho miesto napíše nový.

Príklad:

```
1 0 OVER 1
```

```
2 0 PRINT AT 10,10; "Didaktik.Gama"; AT 10,10; "-----"
```

Syntax: INVERSE <t>

Pre hodnotu parametra t = 0 je výpis normálny a pre t = 1 je výpis inverzný. Napríklad PRINT INVERSE 1; "text".

Existuje špeciálne funkcia, ktorá umožňuje zistiť tzv. atribúty ľubovoľného bodu na obrazovke. Pomocou tejto funkcie

môžeme pre zvolený bod obrazovky určiť, aké boli preň zadané hodnoty parametrov príkazmi PAPER, INK, FLASH a BRIGHT.

Funkcia sa volá ATTR a jej využitie je nasledovné:

Syntax: ATTR <(x,y)>

x,y sú súradnice zvoleného bodu a výsledok funkcie je číslo, ktoré je súčtom 4 čísiel, pre ktoré platí:

128 - ak bod bliká /FLASH 1/ alebo 0 ak je bod stály

64 - ak je zvýšený jas /BRIGHT 1/ alebo 0 pre normálny jas

8 \* hodnota parametra f z príkazu PAPER

1 \* hodnota parametra f z príkazu INK

Príklad:

10 PRINT AT 10,10; FLASH 1; PAPER 2 ; INK 7; "p"

20 PRINT ATTR /10,10/

# Generovanie tónov

Pre ovládanie zvukového výstupu je určený príkaz BEEP,  
pre ktorý platí:

Syntax: BEEP <a,b>

Parameter a určuje dĺžku trvania tónu a parameter b jeho výš-  
ku. Dĺžka trvania sa udáva v sekundách a výška tónu v poltón-  
och od stredného c<sup>1</sup>, ktoré má frekvenciu 261 Hz /čiže b mô-  
že byť kladné i záporné/.

# Príkazy pre tlačiareň

Ako periférne zariadenie k mikropočítaču Didaktik Gama je možné používať špeciálnu tlačiareň ZX Printer, alebo ľubovoľnú mozaikovú tlačiareň. Pre pripojenie tlačiarne je využitý paralelný interface a pre jej používanie platia nasledovné príkazy :

LLIST [n] - vypíše listing programu od riadku s číslom n, resp. celý program, ak nebolo zadané číslo riadku.

LPRINT [zoznam výrazov] - vypíše zoznam výrazov, pričom pre ich oddelenie platí to isté, čo pri príkaze PRINT. Každý príkaz LPRINT bez parametra znamená nový riadok.

COPY - vytlačí kópiu televíznej obrazovky - tento príkaz platí iba pri používaní tlačiarne ZX Printer.

Po zapnutí počítača je obvod MHB 8255 nastavený do módu 0, porty A,B ako vstupné, port C ako výstupný. Tlač je inicializovaná na ZX Printer.

V prípade použitia paralelnej tlačiarne je nutné najprv previesť inicializáciu príkazom RANDOMIZE USR 14630. Tým sa nastaví obvod MHB 8255 do módu 1, port A ako výstupný, na PC6 sa privedie signál ACK, na PC7 signál STB. Súčasne sa nastaví systémová premenná PCZ /23681/ na hodnotu 80 znakov na riadok. Po tejto inicializácii je počítač pripravený komunikovať s tlačiarňou.

Ak tlačiareň potrebuje komplementované dáta, na systémovú premennú PCZ je nutné vložiť hodnotu 128+ počet znakov, teda nastaviť najvyšší bit tejto premennej na hodnotu 1. Z toho tiež vyplýva, že maximálny počet znakov v riadku je 127.

Spätný prechod na ZX Printer a základnú inicializáciu /mód 0/ prevedie príkazom RANDOMIZE USR 14632.

Poznámka : komunikácia s paralelnou tlačiarňou sa dá kedykoľvek prerušiť stlačením klávesov CAPS SHIFT + Q. Použite to však len v prípade, ak sa tlač nedá prerušiť stlačením klávesov CAPS SHIFT + BREAK.

# Príkazy pre mikrodrive

V prípade použitia periférneho zariadenia s veľkou rýchlosťou záznamu - mikrodrive, sú pripravené na použitie špeciálne príkazy pre prácu s ním. Sú to príkazy OPEN # , CLOSE # , MOVE, ERASE, CAT a FORMAT. Tieto príkazy zatiaľ nepracujú.



## Práca s byteami

Možnosť pracovať so znakmi, číslami a ich ukladaním do pamäti / resp.výberom/ je daná existenciou príkazov POKE, PEEK, BIN, CODE.

Syntax: POKE <a,d>

Tento príkaz zabezpečí uloženie hodnoty d na adresu a /obidva parametre sa zadávajú dekadicky/.

Syntax: PEEK <a>

Určuje hodnotu bytu uloženého na adrese a.

Príklad:

```
10 POKE 31000,100
```

```
20 PRINT PEEK 31000
```

Syntax: BIN <b>

Táto funkcia umožňuje zadávať číslo v dvojkovej sústave a prepočíta ho na číslo v desiatkovej sústave. Maximálne číslo ktoré môže prepočítať, je dekadicky 65535.

Príklad:

```
10 PRINT BIN 101
```

vypíše na obrazovku číslo 5.

Syntax: CODE <c#>

Určuje ASCII kód prvého znaku reťazca c#. Ak je reťazec prázdny, hodnota c# je 0. Nasledujúci príklad ukazuje možnosť použitia príkazu CODE v spolupráci s príkazom INKEY# pri výpise kódu stlačeného klávesu.

Príklad:

```
10 IF INKEY# = " " THEN GO TO 10
```

```
20 PRINT CODE INKEY#; " " GO TO 10
```

# Použitie strojového kódu

V tejto časti sa budeme venovať možnostiam využitia mikropočítača Didaktik Gama za predpokladu znalosti strojového kódu mikroprocesora Z 8 $\mu$ . Na spustenie programu v strojovom kóde sa používa funkcia USR.

Syntax: PRINT USR <a>

vykoná program v strojovom kóde začínajúci od adresy a a vytlačí obsah dvojice vnútorných registrov mikroprocesora B,C.

Syntax: RANDOMIZE USR <a>

vykoná program v strojovom kóde od adresy a a výsledok uloží na adresu, na ktorú ukladá číslo z postupnosti náhodných čísiel.

Syntax: RESTORE USR <a>

Len vykoná program v strojovom kóde od adresy a.

Je možné pomocou príkazu USR zistiť rozsah voľnej časti pamäti nasledovným spôsobom:

```
PRINT 65536 - USR 7962
```

mikropočítač vypíše na obrazovku počet bytov voľnej pamäti. Program v strojovom kóde by mal byť ukončený inštrukciou RET, aby bol zaistený návrat do jazyka BASIC.

## Príklad programu v strojovom kóde

```
1 $\mu$  FOR a= $\mu$  TO 23
```

```
2 $\mu$  READ b
```

```
3 $\mu$  POKE 6 $\mu$  $\mu$  $\mu$  $\mu$ +a,b
```

```
4 $\mu$  NEXTa
```

```
5 $\mu$  DATA 33,8 $\mu$ ,195,17, $\mu$ ,64,1, $\mu$ ,27,237,176,2 $\mu$ 1,33, $\mu$ ,64,17,8 $\mu$ ,  
195,1, $\mu$ ,27,237,176,2 $\mu$ 1
```

```
6 $\mu$  BEEP 1,1 $\mu$ :PAUSE  $\mu$ 
```

```
100 FOR t=0 TO 2 * PI STEP PI/100
110 PLOT INK 4;127,87
120 DRAW INK 4;120 = SIN t + SIN t,80 = COS t + COS t
130 NEXT t
140 RANDOMIZE USR 60012
150 BEEP 1,20; PAUSE 0: CLS
200 FOR a=0 TO 6911
210 POKE 16384+a,PEEK (50000+a)
220 NEXT a
230 BEEP 1,10;PAUSE 0
240 CLS:PAUSE 0
300 RANDOMIZE USR 60000
```

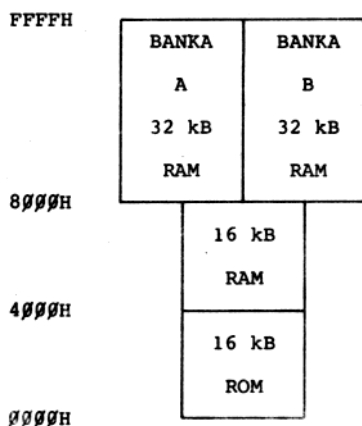
Program v strojovom kóde zadávame pomocou príkazov DATA a POKE. Tento program je založený na uložení adres do dvojíc registrov HL, DE a počtu bytov do BC. Využitím inštrukcie LDIR pre blokový prenos dát /dĺžku bloku určí obsah dvojice BC/ prenášame obsah pamäťovej bunky, ktorej adresa je v HL do pamäťovej bunky adresovanej DE, až kým nie je prenesený celý blok. Jedna časť programu v strojovom kóde umožňuje prenos časti pamäte RAM do VIDEORAM a druhá prenos opačným smerom, t.j. z VIDEORAM do RAM. Vlastný program v programovacom jazyku BASIC na riadkoch 10 až 50 zabezpečí uloženie programu v strojovom kóde do pamäti RAM od adresy 60000 /druhá časť programu začína od adresy 60012/. Na riadku 60 je pípaním ohlásené naplnenie časti RAM programom v strojovom kóde a počítač očakáva stlačenie ľubovoľného klávesu, aby mohol pokračovať v programe. Cyklus na riadkoch 100 až 130 kreslí obrázok na obrazovku. Príkaz RANDOMIZE USR na riadku 140 odštartuje druhú časť programu v strojovom kóde od adresy 60012 a tým uloží obrázok z VIDEORAM do oblasti RAM od adresy 50000.

Potom nasleduje opäť príkaz BEEP a stlačenie klávesu. Na riadkoch 200 až 220 sa uskutoční prenos dát z pamäte RAM od adresy 50000 /určujúcich predtým vykreslený obrázok/ do VIDEORAM pomocou príkazov POKE a PEEK. Na riadku 240 opäť čaká na stlačenie klávesu. Potom príkazom RANDOMIZE USR vykoná prvú časť programu v strojovom kóde - prenesie obrázok z RAM priamo do VIDEORAM použitím strojového kódu. Je možné porovnať rýchlosť prenosu dát určujúcich obrázok z RAM do VIDEORAM pomocou príkazov programovacieho jazyka BASIC a pomocou strojového kódu.

# Popis pamäti

V mikropočítači Didaktik Gama sú použité dva druhy pamäti. Pamäť ROM je pevná pamäť, v ktorej je uložený interpreter Basic. Pamäť RAM slúži na vkladanie programov a dát a na rozdiel od pamäti ROM, z ktorej je možné len čítať, je možné do pamäti RAM dáta i zapisovať.

Kapacita pamäti ROM je 16 kByte, pamäť RAM má rozsah 80 kByte. Na nasledujúcom obrázku je graficky znázornená mapa pamäti.



Pamäť ROM leží adresne od 0000H po 3FFFH. Od adresy 4000H leží pamäť RAM 16 kByte. Do tejto časti pamäti RAM sa ukladajú údaje zobrazované na obrazovke. Takúto časť pamäti RAM označujeme ako VIDEORAM. Okrem obrazovky sa do tejto časti pamäti vkladajú systémové premenné, premenné, program v Basicu i strojovom kóde a rad ďalších údajov.

Od adresy 8000H sú umiestnené dve pamäťové banky RAM,

každá s rozsahom 32 kByte. Užívateľ môže programove meniť banku, v ktorej pracuje príkazom OUT 127,0 resp. OUT 127,1. Potom je na klávesnici indikované diódou LED, v ktorej banke sa momentálne užívateľ nachádza.

Poznámka:

Pri prepnutí z jednej banky do druhej a pri nastavení MHB 8255 sa prenášajú dáta medzi bankami a to od adresy 8000H až po RAMTOP.

Teraz uvedieme popis pamäti RAM:

VIDEO	SCREEN	4000H	
	ATRIBÚTY	5800H	
	BUFFER LP	5B00H	
	SYS PREM	5C00H	
	MAPY MIKRODRIVOV	5CB6H	
	KANÁLOVÉ INFORMÁCIE	CHANS	
	00H	PROG	
	PROGRAM V BASICU	VARs	
	PREMENNÉ BASICU		
	00H	ELINE	
	EDITOVANÝ RIADOK ALEBO PRÍKAZ		
	NL		
	00H	WORKSP	
	VSTUPNÉ DÁTA		
	NL		
	DOČASNÝ PRACOVNÝ PRIESTOR	STKBOT	
	ZÁSOBNÍK KALKULÁTORA	STKEND	
	VOĽNÝ PRIESTOR	SP	
	ZÁSOBNÍK		
	GOSUB ZÁSOBNÍK		
?	RAMTOP	/koniec progra-	
3EH	UDG	movej pamäti/	
UŽIVATEĽSKÁ GRAFIKA	P-RAMT	/koniec fyzickej pamäti/	

# Paralelný interface

Mikropočítač Didaktik Gama má vstavaný vlastný paralelný interface, ktorý je osadený paralelným stykovým obvodom MHB 8255. Užívateľovi je k dispozícii 23 vstupno-výstupných liniek tohto obvodu. Linka PC<sub>0</sub> je interne využitá pre bankovanie pamäti RAM a každá neuvážaná operácia s ňou môže viesť k zrušeniu systému. Pri základnej inicializácii mikropočítača je obvod MHB 8255 nastavený tak, že porty A a B sú ako vstupné a port C ako výstupný. To dovoľuje priame pripojenie joysticku pre ovládanie hier a pod. Pre zmenu režimu činnosti je možné použiť príkaz OUT, pričom platia nasledovné adresy pre daný stykový obvod :

CWR = 127

A = 31

B = 63







C = 95

Pre programovanie režimu činnosti obvodu MHB 8255 platia bežné katalógové údaje pre jednotlivé módy. Signály stykového obvodu nie sú oddelené budičmi a sú priamo vyvedené na priamy interfaceový konektor s 30-timi vývodmi. Jeho popis je v závere tejto kapitoly. Okrem signálov stykového obvodu sú na konektor vyvedené tri invertory pre prípadnú potrebu inverzie signálov, napr. pre pripojenie tlačiarne.

Poznámka :

Orientácia FRB konektora je zrejmä z obr. č. 1. na strane 4.

Popis priameho interfaceového konektora, ktorý je číslovaný rovnako ako FRB konektor. /Zárez na interfaceovom konektore nie je číslovaný/.

- 1. PA<sub>0</sub>
- 2. PA<sub>1</sub>
- 3. PA<sub>2</sub>
- 4. PA<sub>3</sub>
- 5. PA<sub>4</sub>
- 6. PA<sub>5</sub>
- 7. PA<sub>6</sub>
- 8. PA<sub>7</sub>
- 9. PB<sub>0</sub>
- 10. PB<sub>1</sub>
- 11. PB<sub>2</sub>
- 12. PB<sub>3</sub>
- 13. PB<sub>4</sub>
- 14. PB<sub>5</sub>
- 15. PB<sub>6</sub>
- 16. PB<sub>7</sub>
- 17. PC<sub>3</sub>
- 18. PC<sub>2</sub>
- 19. PC<sub>1</sub>
- 20. PC<sub>4</sub>
- 21. PC<sub>5</sub>
- 22. PC<sub>7</sub>
- 23. PC<sub>6</sub>
- 24. GND
- 25. 
- 26. 
- 27. 
- 28. 
- 29. 
- 30. 



# Popis systémového konektora

Systémový konektor je realizovaný ako priamy konektor s 2 x 28 vývodmi, medzi ktorými je vzdialenosť 2,54 mm. Na tento konektor je vyvedená kompletná adresná zbernica /A<sub>0</sub> až A<sub>15</sub>/, úplná dátová zbernica /D<sub>0</sub> až D<sub>7</sub>/, riadiaca zbernica a pomocné signály.

Systémový konektor je identický so systémovým konektorom mikropočítača ZX SPECTRUM až na napájacie napätie a signál VIDEO. Z napájacích napätí má Didaktik Gama vyvedené iba napätie +5 V. Orientácia systémového konektora je zabezpečená výrezom v doske plošného spoja. Rozmiestnenie jednotlivých signálov je nasledovné:

Strana spájkov.		Strana súčiastok
A14	1	A15
A12	2	A13
+ 5 V	3	D7
NC	4	NC
VÝREZ	5	VÝREZ
GND	6	D <sub>0</sub>
GND	7	D <sub>1</sub>
CLK	8	D <sub>2</sub>
A <sub>0</sub>	9	D <sub>6</sub>
A <sub>1</sub>	10	D <sub>5</sub>
A <sub>2</sub>	11	D <sub>3</sub>
A <sub>3</sub>	12	D <sub>4</sub>
IORQ <sub>E</sub>	13	INT <sub>A</sub>
GND	14	NMI
NC	15	HALT
Y	16	MREQ
V	17	IORQ
U	18	RD
BUSRQ	19	WR
RESET	20	NC
A <sub>7</sub>	21	WAIT
A <sub>6</sub>	22	NC
A <sub>5</sub>	23	NC
A <sub>4</sub>	24	M1
ROMCS	25	RFSH
BUSAK	26	A <sub>8</sub>
A <sub>9</sub>	27	A <sub>10</sub>
A <sub>11</sub>	28	NC

NC — nezapojené

## UPOZORNENIE :

Pri pripájaní periférnych zariadení k mikropočítaču Didaktik Gama je nutné odpojiť mikropočítač od napájacieho zdroja !

# Systemové premenné

Časť pamäti od adresy 23552 do 23750 je určená pre špecifické využitie systému a je možné s ňou pracovať pomocou príkazov PEEK a POKE. Uvedieme zoznam systémových premenných spolu s ich menami, ktoré však slúžia len ako pomôcka.

Údaj v prvom stĺpci /poznámka/ má nasledovný význam:

X - premenná sa nesmie spracovávať POKE, môže prísť k havárii systému

N - práca s POKE nebude mať žiadny účinok

Číslo vedľa poznámky určuje počet bytov premennej.

Pozn.	Adresa	Meno	Obsah
N8	23552	KSTATE	použitá pre čítanie klávesnice
N1	23560	LAST K	hodnota posledného stlačeného klávesu
1	23561	REPDEL	čas /v 1/50 sek/
1	23562	REPPER	oneskorenie /v 1/50 sek/
N2	23563	DEFADD	adresa argumentov užívateľom definovanej funkcie
N1	23565	K DATA	ukladá druhý byte - vstup z klávesnice
N2	23566	TV DATA	ukladá byty farby, riadi AT a TAB
X38	23568	STRMS	kanálové adresy, pripojenie k reťazcu
2	23606	CHARS	o 256 menší než adresa znakového poľa
1	23608	RASP	dĺžka varovného signálu
1	23609	PIP	dĺžka signálu pri stlačení klávesu
1	23610	ERR NR	o jednu menšia ako kód správy
X1	23611	FLAGS	rôzne znaky pre kontrolu systému BASIC
X1	23612	TV FLAG	znak pre TV
X2	23613	ERR SP	adresa položky v strojovom reťazci

<u>Pozn.</u>	<u>Adresa</u>	<u>Meno</u>	<u>Obsah</u>
N2	23615	LIST SP	adresa návratu z automatického listingu
N1	23617	MODE	špecifikuje K, L, C, E, G kurzor
2	23618	NEWPPC	riadok, na ktorom sa má skončiť
1	23620	NSPPC	číslo príkazu na riadku, na ktorý sa má skákať
2	23621	PPC	číslo riadku práve vykonávaného príkazu
1	23623	SUBPPC	číslo príkazu v riadku, ktorý sa vykonáva
1	23624	BORDCR	farba lemovania
2	23625	E PPC	číslo riadku /s programovým kurzorom/
X2	23627	VARS	adresa premenných
X2	23629	DEST	adresa premennej s priradením
X2	23631	CHANS	adresa kanálových dát
X2	23633	CURCHL	adresa informácií použitých pre vstup a výstup
X2	23635	PROG	adresa programu v BASIC-u
X2	23637	NXTLIN	adresa ďalšieho riadku v programe
X2	23639	DATA DD	adresa ukončenia minulej položky DATA
X2	23641	E LINE	adresa aktuálnej položky
2	23643	K CUR	adresa kurzoru
X2	23645	CH ADD	adresa ďalšieho interpretovaného znaku
2	23647	X PTR	adresa znaku po otázniku
X2	23649	WORKSP	adresa dočasnej pracovnej oblasti
X2	23651	STKBOT	adresa začiatku kalkulačnej oblasti
X2	23653	STKEND	adresa začiatku voľného priestoru
N1	23655	BREG	kalkulačný b register
N2	23656	MEM	adresa oblasti použitej pre kalkulač-

<u>Pozn.</u>	<u>Adresa</u>	<u>Meno</u>	<u>Obsah</u>
			né pamäti, obvykle MEMBOT, ale nie vždy
1	23658	FLAGS2	viac príznakov
X1	23659	DF SZ	počet riadkov v dolnej časti obrazovky
2	23660	S TOP	číslo vrcholu programového riadku v automatickom listingu
2	23662	OLDPPC	číslo riadku, kde skáče CONTINUE
1	23664	OSPPC	číslo príkazu na riadku, kde skáče CONTINUE
N1	23665	FLAGX	rôzne príznaky
N2	23666	STRLEN	dĺžka reťazca pre použitie priradenia
N2	23668	T ADDR	adresa ďalšej položky v symbolickej tabuľke
2	23670	SEED	použitie pre RND - premenná v príkaze RANDOMIZE
3	23672	FRAMES	čítač rádu prírastkov po 20 ms
2	23675	UDG	adresa prvého znaku užívateľskej grafiky
1	23677	COOROS	x-ová súradnica posledného kresleného bodu
1	23678		y-ová súradnica posledného kresleného bodu
1	23679	P PONS	33 - počet tlačných pozícií
1	23680	PR CC	menej významný byte adresy ďalšej pozície pre LPRINT v bufferi
1	23681	PCZ	počet znakov na riadok tlačiarne
2	23682	ECHO E	33 - počet stĺpcov a 24 - počet riadkov vstupného buffera
2	23684	DF CC	adresa zobrazovacieho poľa - pozícia PRINT
2	23686	DF CCL	ako DF CC pre dolnú časť obrazovky
X1	23688	S POSN	33 - číslo stĺpca pre PRINT
X1	23689		24 - číslo riadku pre PRINT
X2	23690	SPOSNL	ako S POSNL pre dolnú časť
1	23692	SCR CT	počíta scroll - je vždy o jednu viac ako počet, ktorý bude vykonaný pred zastavením scrollu.

Pozn.	Adresa	Meno	Obsah
1	23693	ATTR P	stále bežné farby
1	23694	MASK P	použité pre transparentné farby. Bit, ktorý je 1, ukazuje, že zodpovedajúci atribútový bit neberie z ATTR P, ale priamo z obrazovky
N1	23695	ATTR T	dočasné bežné farby
N1	23696	MASK T	ako MASK P, ale dočasne
1	23697	P FLAG	viac významov
N3Ø	23698	MEMBOT	kalkulačná pamäťová oblasť použitá k uloženiu čísiel, ktoré nemôžu byť uložené v kalkulačnej oblasti
2	23728		nie je využité
2	2373Ø	RAMTOP	adresa posledného bytu v oblasti BASIC-u
2	23732	P-RAMP	adresa posledného bytu fyzickej RAM

# Hlásenie o chybách

Objavujú sa v dialógovom riadku a vysvetľujú, prečo sa zastavil beh programu. Hlásenia obsahujú číslo alebo písmeno a stručný popis vrátane čísla riadku a čísla príkazu na riadku, kde sa zastavil beh programu.

Kód	Význam
Ø	OK Úspešný koniec programu
1	NEXT without FOR Existuje premenná za príkazom NEXT, ale v príkaze FOR bola použitá iná premenná resp. vôbec nebol použitý príkaz FOR
2	Variable not found Premenná bola volená skôr, než jej bola priradená určitá hodnota resp. pre indexované premenné platí, že nebol určený rozmer premennej
3	Subscript wrong Index pre pole leží mimo poľa alebo bol napísaný ako chybné číslo
4	Out of memory V počítači nie je dostatok voľnej pamäti pre zvolenú operáciu
5	Out of screen Príkaz požadoval zápis mimo pracovnú časť obrazovky
6	Number too big Výpočet viedol k číslu, ktoré je väčšie ako $10^{38}$

Kód	Význam
7	RETURN without GO SUB Existuje viac príkazov RETURN ako GO SUB
8	END of FILE Koniec súboru - platí pre prácu s mikrodrive
9	STOP statement Zastaví beh programu na príkaz STOP, je možné pokračovať na ďalšom riadku príkazom CONTINUE
A	Invalid argument Nevhodný argument funkcie
B	Integer out of range Celé číslo, ktoré bolo požadované, je po zaokrúhlení mimo rozsah
C	Nonsense in BASIC Text v reťazci nemá formu platného výrazu
D	BREAK - CONT repeats Zastavenie programu - je možné pokračovať príkazom CONTINUE zopakovaním posledného vykonaného príkazu
E	Out of DATA Načítavanie dát príkazom READ za koncom zoznamu DATA
F	Invalid file name SAVE má prázdne úvodzovky alebo meno dlhšie ako 18 znakov
G	No room for line V pamäti nie je miesto na prijatie ďalšieho riadku
H	STOP in INPUT Niektoré vstupné data začali STOP alebo bol stlačený INPUT LINE
I	FOR without NEXT Chýba príkaz NEXT odpovedajúci FOR a je nesprávny pa-

Kód	Význam
	parameter STEP
J	Invalid I/O device Chybné I/O zariadenie - operácia s mikrodrive.
K	Invalid colour Nevhodné číslo pre určenie farby alebo jasú
L	BREAK into program Zastavenie programu stlačením BREAK, pokračovanie programu pomocou CONTINUE od ďalšieho príkazu
M	RAMTOP no good Číslo špecifikované pre RAMTOP je buď príliš veľké alebo malé
N	Statement lost Skok na neexistujúci príkaz
O	Invalid stream Chybný tok dát - operácia s mikrodrive, tlačiarňou atď.
P	FN without DEF Nedefinovaná užívateľská funkcia
Q	Parameter error Chyba parametra užívateľskej funkcie
R	Tape loading error Chyba pri nahrávaní



# Zoznam príkazov

V tejto kapitole uvedieme zoznam funkcií a príkazov jazyka BASIC:

Funkcia	Typ argumentu x	Výsledok
ABS	číslo	absolútna hodnota x
ACS	číslo	arkuskosínus x v radiánoch
AND	binárna operácia: pravý operand je vždy číslo: -numer. ľavý ope- rand -retazc. ľavý ope- rand	$A \text{ AND } B = \begin{cases} A & \text{ak } B < > \emptyset \\ \emptyset & \text{ak } B = \emptyset \end{cases}$ $A\$ \text{ AND } B = \begin{cases} A\$ & \text{ak } B < > \emptyset \\ " " & \text{ak } B = \emptyset \end{cases}$
ASN	číslo	arkussínus x v radiánoch
ATN	číslo	arkustangens x v radiánoch
ATTR	číslo, číslo /x,y/	číslo, ktorého binárna forma dáva kód farby na riadku x a stĺpci y na TV obrazovke. Bit 7 určuje blikanie bodu, bit 6 zvýšený jas, bity 5-3 parameter príkazu PAPER a bity 2 - $\emptyset$ parameter príkazu INK
BIN		prepočítanie čísla z dvojkovej sústavy do desiatkovej
CHR $\emptyset$	číslo	znak, ktorého kód je x zaokrúhlené na najbližšie celé číslo
CODE	retazec	kód prvého znaku v x /pre $\emptyset$ je prázdny retazec/

Funkcia	Typ argumentu x	Výsledok
COS	číslo	kosínus x v radiánoch
EXP	číslo	$e^x$
FN		za FN nasleduje písmeno, ktoré vo- lá užívateľsky definovanú funkciu
IN	číslo	do registra a načína hodnotu z portu x; $\emptyset$ - x - FFFFH
INKEY $\emptyset$	žiadny	číta klávesnicu - výsledok je znak v L alebo C móde reprezentujúci stlačené tlačítko
INT	číslo	celá časť /zaokrúhlená nadol/
LEN	reťazce	dĺžka reťazca
LN	číslo	prirodzený logaritmus
OR	binárna operácia	$A \text{ OR } B = \begin{cases} 1 & \text{ak } B < > \emptyset \\ \emptyset & \text{ak } B = \emptyset \end{cases}$
PEEK	číslo	hodnota jedného bytu v pamäti na adrese x /adresa x je zaokrúhlená na najbližšie celé číslo/
PI	žiadny	3,14159265...
POINT	argumenty x,y	1 ak obrazový bod /x,y/ má farbu INK, $\emptyset$ ak ide o podklad
RND	žiadny	ďalšie pseudonáhodné číslo v pos- tupnosti
SCREEN $\emptyset$	argumenty x,y	objaví sa znak z pozície x,y
SGN	číslo	výsledok = -1; $x < \emptyset$ = $\emptyset$ ; $x = \emptyset$ = 1; $x > \emptyset$
SIN	číslo v radiánoch	sínus x
SQR	číslo	druhá odmocnina

Funkcia	Typ argumentu x	Výsledok
STR\$	číslo	reťazec znakov, ktorý by bol zobrazený pri výpise x
TAN	číslo v radiánoch	tangens x
USR	reťazec	adresa bitového vzoru pre užívateľskú grafiku zodpovedajúcu x
USR	číslo	odštartuje program v strojovom kóde od adresy x a výsledok je pri návrate uložený v bc registri
VAL	reťazec	vyhodnocuje x /bez úvodzoviek/ ako numerický výraz
VAL\$	reťazec	vyhodnocuje x /bez úvodzoviek/ ako reťazcové vyjadrenie

V zozname príkazov použijeme nasledovnú symboliku:

Príkaz	Činnosť
a	reprezentuje prostý znak
v	reprezentuje premennú
x,y,z	reprezentuje numerický výraz
m,n	reprezentuje numerický výraz zaokrúhlený na najbližšie celé číslo
e	reprezentuje výraz
f	reprezentuje reťazcovo vyhodnotenú hodnotu
s	reprezentuje postupnosť príkazov oddelených dvojbodkami
c	reprezentuje postupnosť farebných položiek, každá je ukončená čiarkou alebo bodkočiarkou. Farebné položky môžu mať

Príkaz	Činnosť
	formu: PAPER, INK, FLASH, BRIGHT, INVERSE, OVER
BEEP $x, y$	zaznie tón dĺžky $x$ sekúnd vo výške $y$ poltónov nad $c^1$ /alebo pod $c^1$ pre záporné $y$ /
BORDER $m$	farebné lemovanie obrazovky a farba podkladu v dolnej časti obrazovky /dialógovom riadku/ podľa $m$
BRIGHT $n$	zvážšenie jasu - pre $n=8$ jas normálny, pre $n=1$ jas zvýšený, pre $n=8$ jas priehľadný
CAT	príkaz pre mikrodrive
CIRCLE $x, y, z$	kreslí kružnicu so stredom $(x, y)$ a polomerom $z$
CLEAR	ruší premenné, vykoná RESTORE a CLS, odovzdá pozíciu PLOT do ľavého dolného rohu
CLEAR $m$	podobné CLEAR, ale ak je to možné, nastaví nový zásobník pre GO SUB na novú hodnotu RAMTOP na adrese $m$
CLOSE #	nepracuje bez mikrodrive
CLS	maže obrazovku
CONTINUE	pokračuje v programe tam, kde sa zastavil s iným hlásením ako $\emptyset$ . Ak bolo hlásenie 9 alebo L, pokračuje na ďalšom príkaze, inak opakuje posledný príkaz
COPY	vysielala kópiu horných 22 riadkov TV na tlačiareň ZX Printer
DATA $e_1, e_2, \dots$	číta zo zoznamu dáta - musí byť v programe
DEF FN $a(a_1, \dots, a_k)$	ezužívateľsky definovaná funkcia

Príkaz	Činnosť
DELETE f	nepracuje bez mikrodrive
DIM a(n <sub>1</sub> ,...,n <sub>k</sub> )	vytvára pole a s dimenziami n <sub>1</sub> ,...,n <sub>k</sub>
DIM a\$ ( n <sub>1</sub> ,...,n <sub>k</sub> )	vytvára pole znakov a\$ s dimenziami n <sub>1</sub> ..n <sub>k</sub>
DRAW x,y	DRAW x,y,\$
DRAW x,y,z	kreslí z bežnej grafickej pozície, posúva x vodorovne a y zvisle resp. kreslí oblúk s uhlom z
ERASE	nepracuje bez mikrodrive
FLASH n	definuje blikanie znaku-pre n \$=je stály, pre n=1 znak bliká, pre n=8 sa blikanie nedá meniť
FOR a=x TO y STEP z	ruší premennú a a určí novú premennú s hodnotou x, limitom y a krokom z
FORMAT f	nepracuje bez mikrodrive
GO SUB m	dáva číslo príkazu GO SUB a funguje ako GO TO m
GO TO m	skok na riadok m /ak nie je použitý, tak na najbližší vyšší riadok/
IF x THEN s	ak je x pravdivé, vykoná sa s
INK m	určuje farbu písmen od \$ do 7, m= 8 je priehľadné a m=9 kontrastné
INPUT	vkladanie vstupných údajov z klávesnice
INVERSE n	riadenie kontrolných znakov pre výpis-pre n=\$ sú znaky vypisované normálne, pre n=1 je vymenená farba písmen s farbou podkladu
LET v=e	priradenie hodnoty e do premennej v
LIST m	vypisuje listing programu od riadku m

<u>Príkaz</u>	<u>Činnosť</u>
	/ak nie je udané m, tak od najnižšieho riadku/
LLIST m	obdoba LIST pre tlačiareň
LOAD f	nahráva do pamäti program a premenné z magnetofónu
LOAD f DATA v( )	nahráva numerické pole
LOAD f DATA v\$( )	nahráva do pamäti reťazcové pole
LOAD f CODE m,n	nahráva najviac n bytov začínajúcich od adresy m
LOAD f CODE m	nahráva byty od adresy m
LOAD f CODE	nahráva byty na tú adresu, z ktorej boli zaznamenané na magnetofón
LOAD f SCREENS	nahrá zmienené pole a zruší predchádzajúcu verziu v pamäti
LPRINT	obdoba PRINT s použitím pre tlačiareň
MERGE f	obdoba LOAD f, ale neruší riadky starého programu a premenné, ak nemajú rovnaké čísla a mená
MOVE f <sub>1</sub> ,f <sub>2</sub>	nepracuje bez mikrodrive
NEW	štartuje systém BASIC, ruší program a premenné
NEXT	hľadá kontrolnú premennú, pričíta krok k jej hodnote a pre zvolený krok a limit rozhoduje o návrate na začiatok cyklu
OPEN #	nepracuje bez mikrodrive
OUT m,n	zapíše hodnotu n na adresu m

<u>Príkaz</u>	<u>Činnosť</u>
OVER n	kontrolný znak pre prepísanie už napísaných znakov - pre $n = 0$ nové znaky vynechávajú staré, pre $n = 1$ sú staré a nové znaky spojené
PAPER m	určuje farbu podkladu /pre m platí to isté, čo pre INK/
PAUSE m	zastavuje program na dobu $m * 0.02$ sek - ak $m = 0$ , zastaví program do stlačenia klávesu
PLOT c;m,n	vypíše bod farbou určenou INK daný súradnicami (  m ,  n  ), pričom m je od 0 do 255 a n od 0 do 175
POKE m,n	píše hodnotu n do bytu na adrese m
PRINT	uskutoční výpis na obrazovku - text, numerický výraz alebo refazcové vyjadrenie. Ak sú položky oddelené bodkočiarkou, píše ich tesne vedľa seba, ak čiarkou, píše ich do dvoch stĺpcov a ak apostrofom, píše každú na nový riadok
PRINT TAB x	umožňuje zvoliť stĺpec a, od ktorého začne výpis
PRINT AT x,y	umožňuje zvoliť riadok x a stĺpec y, kde začne výpis
RANDOMIZE m	určuje systémovú premennú /SEED/ na generovanie ďalšej hodnoty RND
READ $v_1, \dots, v_k$	priradenie premenných použitím výrazov v zozname DATA

<u>Príkaz</u>	<u>Činnosť</u>
REM	komentár - pri vykonávaní programu je ignorovaný
RESTORE n	znovu sprístupňuje zoznam DATA na riadku s číslom n pre ďalší príkaz READ
RUN n	odštartuje program od riadku n a vykoná na začiatku CLEAR /ak n nie je udané, odštartuje ho od najnižšieho riadku/
SAVE f	ukladá program a premenné na magnetofónovú pásku
SAVE f LINE m	ukladá program a premenné a po ich nahratí automaticky skáče na riadok s číslom m
SAVE f DATA v( )	ukladá numerické pole
SAVE f DATA vs( )	ukladá reťazcové pole
SAVE f CODE m,n	ukladá n bytov od adresy m
SAVE f SCREEN\$	ukladá informáciu o obrazovke s menom f
STOP	zastavuje program, CONTINUE umožní pokračovanie na ďalšom príkaze
VERIFY	rovnako ako LOAD, ale dáta sa nenahrávajú do RAM, ale porovnávajú sa s jej obsahom



# Tabuľka znakov

Každý znak v mikropočítači má svoje číselné vyjadrenie. Pretože Didaktik Gama nepoužíva ASCII kód, uvádzame kompletnú tabuľku znakov a ich číselné vyjadrenie v ľavej časti tabuľky. V pravej časti uvádzame inštrukcie Z 80 v strojovom kóde a assembleri. Význam stĺpcov "po CB" a "po ED" vysvetlíme na príklade:

inštrukcia \*7A = ld a,d

ak nasleduje po CB

$$\left. \begin{array}{l} *CB \\ *7A \end{array} \right\} = \text{bit } 7,d$$

a po ED

$$\left. \begin{array}{l} *ED \\ *7A \end{array} \right\} = \text{adc hl,sp}$$

Tabuľka umožňuje písať vlastné programy v strojovom kóde.

Kód	Znak	Hex	Z80 Assembler	po CB	po ED	
0	}	00	nop	rlc b		
1		01	ld bc,NN	rlc c		
2		02	ld /bc/,a	rlc d		
3		nevyužitý	03	inc bc	rlc e	
4		04	inc b	rlc h		
5		05	dec b	rlc l		
6	PRINT	06	ld b,N	rlc /hl/		
7	EDIT	07	rlca	rlc a		
8	kurzor vľavo	08	ex af,af	rrc b		

Kód	Znak	Hex	Z80 Assembler	po CB	po ED
9	kurzor vpravo	09	add hl, bc	rrc c	
10	kurzor dole	0A	lda, /bc/	rrc d	
11	kurzor hore	0B	dec bc	rrc e	
12	DELETE	0C	inc c	rrc h	
13	ENTER	0D	dec c	rrc i	
14	číslo	0E	ld c, N	rrc /hl/	
15	nevyužité	0F	rrca	rrc a	
16	INK	10	djnz DIS	rl b	
17	PAPER	11	ld de, NN	rl c	
18	FLASH	12	ld /de/, a	rl d	
19	BRIGHT	13	inc de	rl e	
20	INVERSE	14	inc d	rl h	
21	OVER	15	dec d	rl l	
22	AT	16	ld d, N	rl /hl/	
23	TAB	17	rla	rl a	
24	nevyužité	18	jr DIS	rr b	
25		19	add hl, de	rr c	
26		1A	ld a, /de/	rr d	
27		1B	dec de	rr e	
28		1C	inc e	rr h	
29		1D	dec e	rr l	
30		1E	ld e, N	rr /hl/	
31		1F	rra	rr a	
32	medzera	20	jr nz, DIS	sla b	
33	!	21	ld hl, NN	sla c	
34	"	22	ld /NN/, hl	sla d	
35	*	23	inc hl	sla e	
36	§	24	inc h	sla h	
37	%	25	dec h	sla l	

Kód	Znak	Hex	Z80 Assembler	po CB	po ED
38	&	26	ld h,N	sra /hl/	
39	'	27	daa	sra a	
40	(	28	jr z,DIS	sra b	
41	)	29	add hl,hl	sra c	
42	*	2A	ld hl,/NN/	sra d	
43	+	2B	dec hl	sra e	
44	,	2C	inc l	sra h	
45	-	2D	dec l	sra l	
46	.	2E	ld l,N	sra /hl/	
47	/	2F	cpl	sra a	
48	ø	30	jr nc,DIS		
49	1	31	ld cp,NN		
50	2	32	ld /NN/,a		
51	3	33	inc sp		
52	4	34	inc /hl/		
53	5	35	dec /hl/		
54	6	36	ld /hl/,N		
55	7	37	scf		
56	8	38	jr c,DIS	srl b	
57	9	39	add hl,sp	srl c	
58	:	3A	ld a,/NN/	srl d	
59	;	3B	dec sp	srl e	
60	<	3C	inc a	srl h	
61	=	3D	dec a	srl l	
62	>	3E	ld a,N	srl /hl/	
63	?	3F	ccf	srl a	
64	@	40	ld b,b	bit ø,b	in b,/c/
65	A	41	ld b,c	bit ø,c	out /c,b/
66	B	42	ld b,d	bit ø,d	sbc hl,bc

k5d	Znak	Hex	Z80 Assembler	po CB	po ED
67	C	43	ld d,e	bit 0,e	ld /NN/,bc
68	D	44	ld b,h	bit 0,h	neg
69	E	45	ld b,l	bit 0,l	retn
70	F	46	ld b,/hl/	bit 0,/hl/	im 0
71	G	47	ld b,a	bit 0,a	ld i,a
72	H	48	ld c,b	bit 1,b	in c,/c/
73	I	49	ld c,c	bit 1,c	out /c/,c
74	J	4A	ld c,d	bit 1,d	adc hl,bc
75	K	4B	ld c,e	bit 1,e	ld bc,/NN/
76	L	4C	ld c,h	bit 1,h	
77	M	4D	ld c,l	bit 1,l	reti
78	N	4E	ld c,/hl/	bit 1,/hl/	
79	O	4F	ld c,a	bit 1,a	ld r,a
80	P	50	ld d,b	bit 2,b	in d,/c/
81	Q	51	ld d,c	bit 2,c	out /c/,d
82	R	52	ld d,d	bit 2,d	sbc hl,de
83	S	53	ld d,e	bit 2,e	ld /NN/,de
84	T	54	ld d,h	bit 2,h	
85	U	55	ld d,l	bit 2,l	
86	V	56	ld d,/hl/	bit 2,/hl/	im 1
87	W	57	ld d,a	bit 2,a	ld a,i
88	X	58	ld e,b	bit 3,b	in e,/c/
89	Y	59	ld e,c	bit 3,c	out /c/,e
90	Z	5A	ld e,d	bit 3,d	adc hl,de
91	[	5B	ld e,e	bit 3,e	ld de,/NN/
92	/	5C	ld e,h	bit 3,h	
93	]	5D	ld e,l	bit 3,l	
94	↑	5E	ld e,/hl/	bit 3,/hl/	im 2
95	_	5F	ld e,a	bit 3,a	ld a,r

Kód	Znak	Hex	Z80 Assembler	po CB	po ED
96	f	60	ld h,b	bit 4,b	in h,/c/
97	a	61	ld h,c	bit 4,c	out /c/,h
98	b	62	ld h,d	bit 4,d	sbc hl,hl
99	c	63	ld h,e	bit 4,e	ld /NN/,hl
100	d	64	ld h,h	bit 4,h	
101	e	65	ld h,l	bit 4,l	
102	f	66	ld h,/hl/	bit 4,/hl/	
103	g	67	ld h,a	bit 4,a	rrd
104	h	68	ld h,b	bit 5,b	in l,/c/
105	i	69	ld l,c	bit 5,c	out /c/,l
106	j	6A	ld l,d	bit 5,d	adc hl,hl
107	k	6B	ld l,e	bit 5,e	ld hl,/NN/
108	l	6C	ld l,h	bit 5,h	
109	m	6D	ld l,l	bit 5,l	
110	n	6E	ld l,/hl/	bit 5,/hl/	
111	o	6F	ld l,a	bit 5,a	rld
112	p	70	ld /hl/,b	bit 6,b	in f,/c/
113	q	71	ld /hl/,c	bit 6,c	
114	r	72	ld /hl/,d	bit 6,d	sbc hl,sp
115	s	73	ld /hl/,e	bit 6,e	ld /NN/,sp
116	t	74	ld /hl/,h	bit 6,h	
117	u	75	ld /hl/,l	bit 6,l	
118	v	76	halt	bit 6,/hl/	
119	w	77	ld /hl/,a	bit 6,a	
120	x	78	ld a,b	bit 7,b	in a,/c/
121	y	79	ld a,c	bit 7,c	out /c/,a
122	z	7A	ld a,d	bit 7,d	adc hl,sp
123	{	7B	ld a,e	bit 7,e	ld sp,/NN/
124		7C	ld a,h	bit 7,h	

Kód	Znák	Hex	Z80 Assembler	po CB	po ED
125	}	7D	ld a,l	bit 7,l	
126	-	7E	ld a,/hl/	bit 7,/hl/	
127	©	7F	ld a,a	bit 7,a	
128	☐	80	add a,b	res 0,b	
129	☐	81	add a,c	res 0,c	
130	☐	82	add a,d	res 0,d	
131	☐	83	add a,e	res 0,e	
132	☐	84	add a,h	res 0,h	
133	☐	85	add a,l	res 0,l	
134	☐	86	add a,/hl/	res 0,/hl/	
135	☐	87	add a,a	res 0,a	
136	☐	88	adc a,b	res 1,b	
137	☐	89	adc a,c	res 1,c	
138	☐	8A	adc a,d	res 1,d	
139	☐	8B	adc a,e	res 1,e	
140	☐	8C	adc a,h	res 1,h	
141	☐	8D	adc a,l	res 1,l	
142	☐	8E	adc a,/hl/	res 1,/hl/	
143	☐	8F	adc a,a	res 1,a	
144	(a)	90	sub b	res 2,b	
145	(b)	91	sub c	res 2,c	
146	(c)	92	sub d	res 2,d	
147	(d)	93	sub e	res 2,e	
148	(e)	94	sub h	res 2,h	
149	(f)	95	sub l	res 2,l	
150	(g)	96	sub /hl/	res 2,/hl/	
151	(h)	97	sub a	res 2,a	
152	(i)	98	sbc a,b	res 3,b	
153	(j)	99	sbc a,c	res 3,c	

uživatel

ská

grafika

Kód	Znak	Hex	Z80 Assembler	po CB	po ED
154	(k)	9A	sbc a,d	res 3,d	
155	(l)	9B	sbc a,e	res 3,e	
156	(m)	9C	sbc a,h	res 3,h	
157	(n)	9D	sbc a,l	res 3,l	
158	(o)	9E	sbc a,/hl/	res 3,/hl/	
159	(p)	9F	sbc a,a	res 3,a	
160	(q)	A0	and b	res 4,b	ldi
161	(r)	A1	and c	res 4,c	cpi
162	(s)	A2	and d	res 4,d	ini
163	(t)	A3	and e	res 4,e	outi
164	(u)	A4	and h	res 4,h	
165	RND	A5	and l	res 4,l	
166	INKEY\$	A6	and /hl/	res 4,/hl/	
167	PI	A7	and a	res 4,a	
168	FN	A8	xor b	res 5,b	ldd
169	POINT	A9	xor c	res 5,c	cpd
170	SCREEN\$	AA	xor d	res 5,d	ind
171	ATTR	AB	xor e	res 5,e	outd
172	AT	AC	xor h	res 5,h	
173	TAB	AD	xor l	res 5,l	
174	VAL\$	AE	xor /hl/	res 5,/hl/	
175	CODE	AF	xor a	res 5,a	
176	VAL	B0	or b	res 6,b	ldir
177	LEN	B1	or c	res 6,c	cpir
178	SIN	B2	or d	res 6,d	inir
179	COS	B3	or e	res 6,e	otir
180	TAN	B4	or h	res 6,h	
181	ASN	B5	or l	res 6,l	
182	ACS	B6	or /hl/	res 6,/hl/	

užíva-  
teIská  
grafika

Kód	Znak	Hex	Z80 Assembler	po CB	po ED
183	ATN	B7	or a	res 6,a	
184	LN	B8	cp b	res 7,b	laddr
185	EXP	B9	cp c	res 7,c	cpdr
186	INT	BA	cp d	res 7,d	indr
187	SQR	BB	cp e	res 7,e	otdr
188	SGN	BC	cp h	res 7,h	
189	ABS	BD	cp l	res 7,l	
190	PEEK	BE	cp /hl/	res 7,/hl/	
191	IN	BF	cp a	res 7,a	
192	USR	C0	ret nz	set 0,b	
193	STR	C1	pop bc	set 0,c	
194	CHR	C2	jp nz,NN	set 0,d	
195	NOT	C3	jp NN	set 0,e	
196	BIN	C4	call nz,NN	set 0,h	
197	OR	C5	push bc	set 0,l	
198	AND	C6	add a,N	set 0,/hl/	
199	<=	C7	rst 0	set 0,a	
200	>=	C8	ret z	set 1,b	
201	< >	C9	ret	set 1,c	
202	LINE	CA	jp z,NN	set 1,d	
203	THEN	CB		set 1,e	
204	TO	CC	call z,NN	set 1,h	
205	STEP	CD	call NN	set 1,l	
206	DEF FN	CE	adc a,N	set 1,/hl/	
207	CAT	CF	rst 8	set 1,a	
208	FORMAT	D0	ret nc	set 2,b	
209	MOVE	D1	pop de	set 2,c	
210	ERASE	D2	jp nc,NN	set 2,d	
211	OPEN *	D3	out /N/,a	set 2,e	



Kód	Znak	Hex	Z80 Assembler	po CB	po ED
212	CLOSE #	D4	call nc,NN	set 2,h	
213	MERGE	D5	push de	set 2,l	
214	VERIFY	D6	sub N	set 2,/hl/	
215	BEEP	D7	rst 16	set 2,a	
216	CIRCLE	D8	ret c	set 3,b	
217	INK	D9	exx	set 3,ç	
218	PAPER	DA	jp c,NN	set 3,d	
219	FLASH	DB	'n a,/N/	set 3,e	
220	BRIGHT	DC	call c,NN	set 3,h	
221	INVERSE	DD	prefix inštruk- cie, používajú- cej register ix	set 3,l	
222	OVER	DE	sbc a,N	set 3,/hl/	
223	OUT	DF	rst 24	set 3,a	
224	LPRINT	E0	ret po	set 4,b	
225	LLIST	E1	pop hl	set 4,c	
226	STOP	E2	jp po,NN	set 4,d	
227	READ	E3	ex /sp/,hl	set 4,e	
228	DATA	E4	call po,NN	set 4,h	
229	RESTORE	E5	push hl	set 4,l	
230	NEW	E6	and N	set 4,/hl/	
231	BORDER	E7	rst 32	set 4,a	
232	CONTINUE	E8	ret pe	set 5,b	
233	DIM	E9	jp /hl/	set 5,c	
234	REM	EA	jp pe,NN	set 5,d	
235	FOR	EB	ex de,hl	set 5,e	
236	GO TO	EC	call pe,NN	set 5,h	
237	GO SUB	ED		set 5,l	
238	INPUT	EE	xor N	set 5,/hl/	

Kód	Znak	Hex	Z80 Assembler	po CB	po ED
239	LOAD	EF	rst 4 $\emptyset$	set 5,a	
240	LIST	F $\emptyset$	ret p	set 6,b	
241	LET	F1	pop af	set 6,c	
242	PAUSE	F2	jp p,NN	set 6,d	
243	NEXT	F3	di	set 6,e	
244	POKE	F4	call p,NN	set 6,h	
245	PRINT	F5	push af	set 6,l	
246	PLOT	F6	or N	set 6,/hl/	
247	RUN	F7	rst 48	set 6,a	
248	SAVE	F8	ret m	set 7,b	
249	RANDOMIZE	F9	ld sp,hl	set 7,c	
250	IF	FA	jp m,NN	set 7,d	
251	CLS	FB	ei	set 7,e	
252	DRAW	FC	call m,NN	set 7,h	
253	CLEAR	FD	prefix inštruk- cie používajúcej register iy	set 7,l	
254	RETURN	FE	cp N	set 7,/hl/	
255	COPY	FF	rst 56	set 7,a	

# Obsah

Úvod .....	1
Zostava mikropočítača .....	3
Obsluha klávesnice .....	5
Priamy režim činnosti .....	8
Programový režim činnosti .....	10
Riadiace a pomocné príkazy .....	11
Konštanty a premenné .....	15
Funkcie numerických premenných .....	16
Práca s reťazcami .....	18
Aritmetické výrazy .....	20
Polia .....	22
Výpis na obrazovku .....	23
Zadávanie, vstup a výstup dát .....	26
Práca s magnetofónom .....	30
Cyklus .....	35
Vetvenie programu .....	36
Podprogramy .....	37
Použitie grafických príkazov .....	38
Príkazy pre ovládanie farieb .....	40
Ovládanie jasových úrovní .....	42
Generovanie tónov .....	44
Príkazy pre tlačiareň .....	45
Príkazy pre mikrodrive .....	47
Práca s byteami .....	48
Použitie strojového kódu .....	49
Popis pamäti .....	52
Paralelný interface .....	54

Popis systémového konektora .....	56
Systémové premenné .....	57
Hlásenia o chybách .....	61
Zoznam príkazov .....	64
Tabuľka znakov .....	72
Obsah .....	82

**DIDAKTIK**

---

**GAMA**