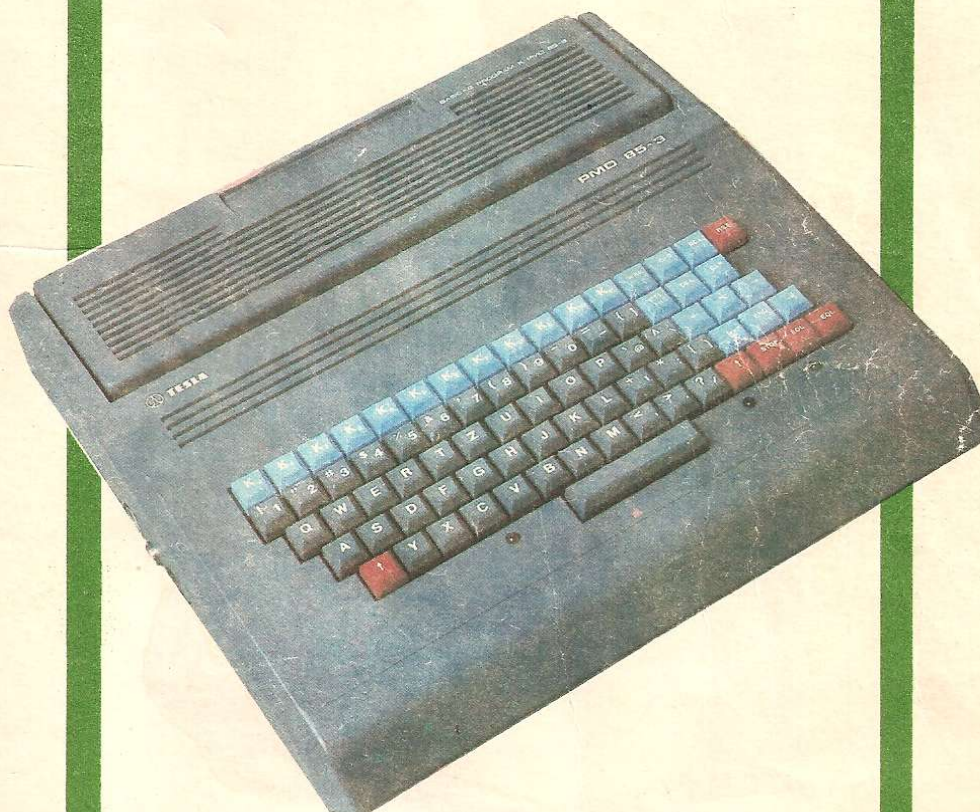


# PMD 85-3



NÁVOD NA POUŽITIE A OBSLUHU



**TESLA** BRATISLAVA k.p.

## O B S A H

1.	ÚVOD.....	1
1.1	Technické údaje.....	2
2.	ZÁKLADNÉ INFORMÁCIE O PMD 85-3 .....	5
2.1	Opis technického vybavenia .....	5
2.1.1	Modul klávesnice.....	5
2.1.2	Modul procesora.....	7
2.1.3	Modul interfejsov.....	7
2.1.4	ROM-modul.....	8
2.2	Konfigurácia PMD 85-3 a jeho obsluha .....	8
3.	KOMUNIKÁCIA POUŽÍVATEĽA S PMD 85-3.....	11
3.1	Zobrazovacia jednotka a spôsob zobrazovania.....	11
3.1.1	Okno binárnej grafiky.....	12
3.1.2	Okno symbolovo orientovanej grafiky.....	14
3.1.3	Okno súradnicovo orientovanej grafiky.....	15
3.1.4	Okno dialógového riadka.....	16
3.2	Klávesnica a vstup údajov.....	17
3.2.1	Znakové klávesy.....	17
3.2.2	Riadiace klávesy.....	22
3.3	Programové klávesy.....	25
3.3.1	Priradenie hodnoty programovému klávesu.....	25
3.3.2	Sprístupnenie obsahu programového klávesu.....	26
4.	BASIC-G.....	27
4.1	Úvodné pojmy.....	27
4.2	Opis príkazov BASIC-G.....	33
5.	MONITOR.....	133
5.1	Príkazy MONITOR-a.....	134
5.2	Režim kompatibility.....	143
5.3	Režim terminálu.....	144
5.4	Podprogramy MONITOR-a.....	146
5.5	Významné premenné MONITOR-a.....	161
5.6	Mapa operačnej pamäte.....	162



## PRÍLOHY

A	OPIS MODULU INTERFEJSOV	163
B	PROGRAMOVANIE VSTUPNO-VÝSTUPNÝCH KANÁLOV	175
C	TABULKA VNÚTORNÉHO KÓDU PMD 85-3	181
D	TABULKA INŠTRUKCIÍ ASEMLERA 8080	183
E	TABULKA ZÁPISNÍKA JAZYKA BASIC-G/V3.0	187
F	TABULKA ZÁPISNÍKA MONITOR-A	189
G	CHYBOVÉ HLÁSENIA INTERPRETERA BASIC-G/V3.0	191
H	PRÍKLADY PROGRAMOV	193
I	SCHÉMY ZAPOJENIA	



# 1. Úvod

Mikropočítač PMD 85-3 je ďalším typom 8-bitových mikropočítačov radu PMD. V jeho konštrukcii sú oproti predošlým typom realizované zmeny v hardvérovej a softvérovej časti, ktorých výsledkom je zvýšenie funkčných a úžitkových vlastností.

Operačná pamäť je konštruovaná na báze modernejších obvodov, ako v časti RWM, tak aj v časti ROM. Mikropočítač umožňuje zobrazovať informácie v štyroch základných farbách. Hardvérové zmeny umožnili rozšíriť základné programové vybavenie, prácu v grafickom režime, zobrazovanie znakov s diakritikou, použitie nových príkazov v jazyku BASIC-G, vykonávať autodiagnostiku mikropočítača atď.

Príručka obsahuje opis základnej zostavy, hardvérových triedkov, príkazov MONITOR-a a programovacieho jazyka BASIC-G/V3.0. Používateľ má možnosť overiť si vykonávanie príkazov na krátkych programoch. V prílohách sú uvedené doplňujúce informácie o práci s periférnymi zariadeniami, tabuľky kódov a inštrukcií mikroprocesora, ako aj príklad programu v jazyku BASIC-G a v jazyku symbolických adries mikroprocesora.

## 1.1. Technické údaje

Mikropočítač PMD 85-3 svojimi parametrami spĺňa podmienky ČSN 34 2860, ČSN 33 4200, ČSN 36 9001, ČSN 36 9302, ČSN 36 7303 a ČSN 34 5751.

Rozmery : 313 x 285 x 65 mm

Hmotnosť: 2.0 kg

Napájacie napätie:        + 5 V  $\pm$  5% / 2.5 A  
                                 + 12 V  $\pm$  5% / 0.1 A  
                                 - 5 V  $\pm$  5% / 0.01 A

(Odbery platia pri použití štandardného ROM-modulu BASIC-G/V3.0).

Celková kapacita operačnej pamäte:	64 KB
Kapacita operačnej pamäte pre používateľa:	48 KB
Kapacita pevnej pamäte MONITOR-a:	8 KB
Kapacita pevnej pamäte programu BASIC-G/V3.0:	10 KB
Celková možná kapacita ROM-modulu:	32 KB
Videostránka:	12 KB

Vstupy - výstupy:

- aplikačný konektor (úroveň TTL)
- konektor na paralelný styk A (úroveň TTL)
- konektor na paralelný styk B (úroveň TTL)
- konektor na styk IMS-2 (úroveň TTL)
- konektor na sériový styk IRPS-20 mA
- konektor na pripojenie kazetového magnetofónu
- konektor pre napájanie
- konektor výstupu UHF
- konektor výstupu VIDEO a RGB



#### Výstupy obrazového signálu:

- vf výstup
  - nosná obrazu 36. kanál (norma PAL)
  - výstupné napätie  $3\text{mV} \pm 1\text{mV}$
- video výstup
  - 1 V / 75 ohmov medzivrcholová hodnota
- RGB výstup
  - 1 V / 75 ohmov medzivrcholová hodnota

Prevádzkové podmienky: teplota +10 °C až +35 °C

relatívna vlhkosť 80%

Prostredie bez agresívnych plynov a dopadajúceho tepelného žiarenia.

#### Dodávané príslušenstvo:

- účastnícka pripojovacia šnúra PKT 12
- kryt konektora 1 PF 258 47
- kryt konektora 1 PF 258 48
- návod na použitie
- záručný list
- baliaci list
- spotrebiteľský obal



Zoznam opravovní poverených opravami mikropočítačov PMD 85.

ELEKTROSERVIS, opravovna  
Dlouhá 34, sídlisko Vltava, 370 37 České Budějovice

KOVOPODNIK města Brna, opravovna  
Husova 8, 657 08 Brno

Středočeský elektroservis, opravovna OPP  
Dům služeb  
Čsl. armády 1676, 256 01 Benešov

Průmyslový podnik města Plzně, opravovna  
Dukelská 17, 301 53 Plzeň

Montážné závody Liberec, opravovna Trizon  
28. října 10, 415 00 Teplice

Opravovna OPMP  
Purkyňovo nám. 1400, 413 03 Roudnice n/L.

Kovoslužba, opravovna OPP  
Hliny 7, 010 00 Žilina

TESLA Eltos, o.p.  
Soukenická 3, 110 00 Praha 1

TESLA Eltos, o.p.  
Mackarova 9, 700 00 Ostrava

TESLA Eltos, o.p.  
Sibírska 1, 800 00 Bratislava

TESLA Eltos, o.p.  
Lesnícka 35, 040 00 Košice

## **2. Základné informácie o PMD 85-3**

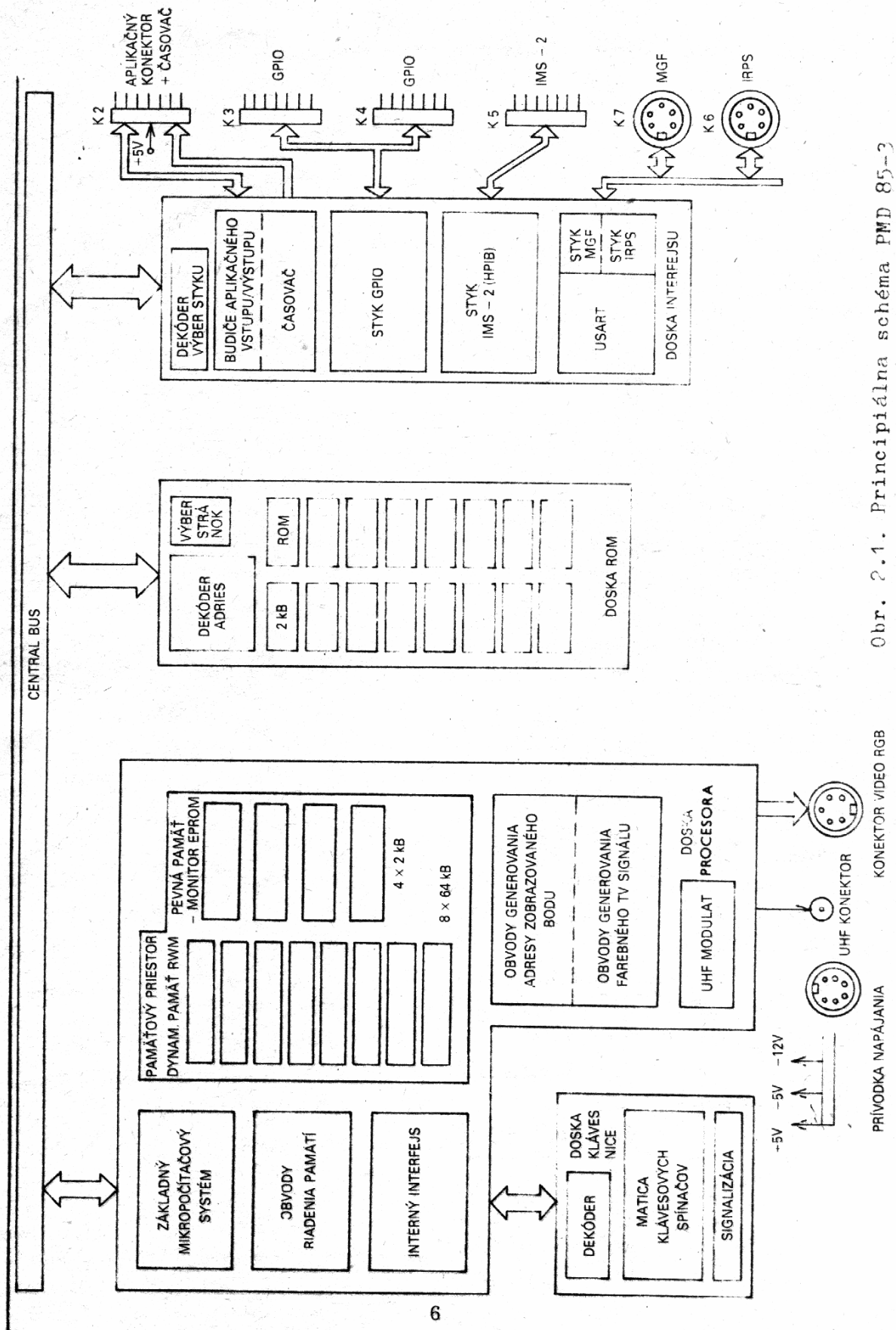
Mikropočítač PMD 85-3 sa svojím technickým a programovým vybavením zaraďuje do kategórie osobných počítačov a jeho použitie je vhodné najmä v oblasti výuky, pri rôznych experimentoch, na riešenie menej náročných úloh z oblasti hromadného spracovania údajov, ale aj ako predmet zábavy. Vývojovo nadväzuje na predchádzajúce typy PMD 85 a PMD 85-2.

### **2.1. Opis technického vybavenia PMD 85-3**

Mikropočítač je umiestnený v skrinke z termoplastického materiálu. Skladá zo 4 základných modulov, ktoré sú realizované na samostatných doskách plošných spojov: modul klávesnice, modul procesora, modul interfejšov a ROM-modul. Prvé tri z nich sú umiestnené v skrinke mikropočítača a posledný sa pripája ako vymeniteľná jednotka v zadnej hornej časti skrinky. Moduly sú navzájom prepojené konektormi. Principiálna schéma mikropočítača je na obr. 2.1.

#### **2.1.1. Modul klávesnice**

Klávesnica je umiestnená v prednej hornej časti skrinky a má funkciu základného vstupného zariadenia pri komunikácii používateľa s mikropočítačom. Je tvorená spínacími prvkami z vodivého elastomeru, ktoré zabezpečujú jej vysokú životnosť a spoľahlivosť, ako aj mäkký chod klávesov. Klávesnica má 78 klávesov, z ktorých 14 je riadiacich a editačných a 12 programovateľných. Odolnosť voči zakmitávaniu spínacích prvkov a akustická signalizácia zopnutia sa riešia programovo.



Obr. 2.1. Principiálna schéma PWD 85-3

## 2.1.2. Modul procesora

Modul procesora je ústrednou časťou mikropočítača. Je v ňom umiestnený blok mikroprocesora (MHB8080A s podpornými obvodmi MHB8228 a MH8224) a operačná pamäť s kapacitou 64 KB tvorená dynamickými obvodmi K 565 RU5G. Posledných 16 KB operačnej pamäte je vyhradených pre videostránku. Jej obsah je spracovávaný videoprocessorom, ktorý generuje farbový signál tvorený zložkami RGB (červená, zelená, modrá), ktoré sú jednotlivito vyvedené na päťkolíkovom konektore. Na tomto konektore je vyvedený aj signál VIDEO, v ktorom sa zlučujú jednotlivé zložky. Úplný farbový signál je okrem toho upravený a modulovaný v UHF modulátore, ktorý je naladený na 36. kanál IV. TV pásma a vyvedený na špeciálnom konektore.

Videoprocessor riadi aj refrešovanie dynamickej operačnej pamäte.

Modul procesora obsahuje aj základné programové vybavenie mikropočítača - MONITOR s rozsahom 8 KB, ktoré je umiestnené v 4 pamäťových obvodoch typu MHB2716. MONITOR-u je vyhradených posledných 8 KB z adresného priestoru, čím sa prekrýva s videostránkou. Obvodové riešenie mikropočítača umožňuje, aby sa činnosť MONITOR-a striedala s činnosťou videoprocessora. Takýmto riešením je pre používateľa PMD 85-3 vytvorená súvislá pamäťová oblasť s kapacitou 48 KB.

## 2.1.3. Modul interfejsov

Rozšírenie základných možností PMD 85-3 dáva modul interfejsov, cez ktorý je možné k nemu pripájať rôzne vonkajšie zariadenia. Podrobný opis je uvedený v prílohe A.

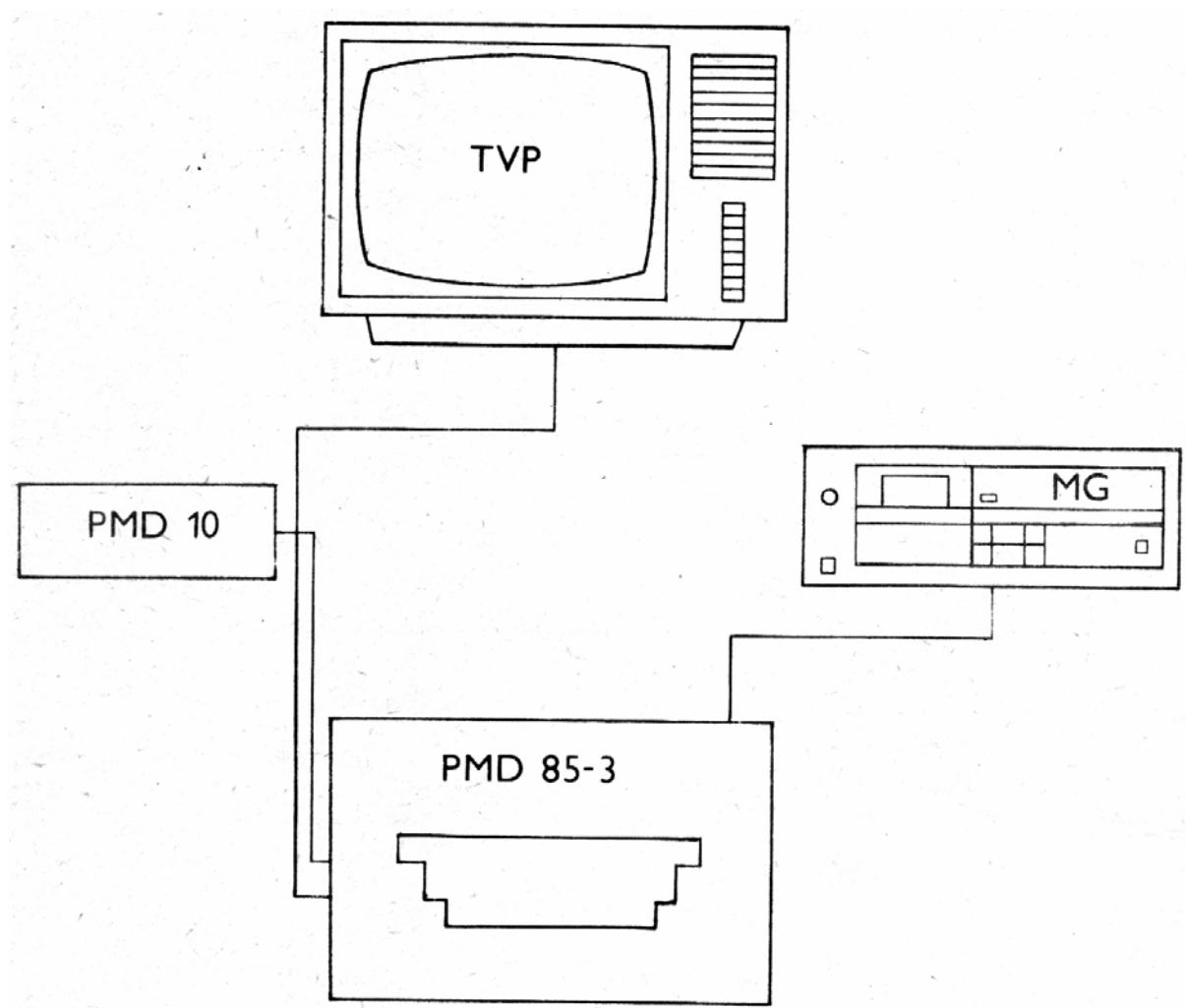
## 2.1.4. ROM modul

V tomto module je možné umiestniť 16 pamätí EPROM typu MHB2716 s celkovou kapacitou 32 KB. Výrobca štandardne osadzuje prvých 5 pozícií (10 KB), v ktorých je uložený interpretér jazyka BASIC-G. Osaďenie zvyšných pozícií si zabezpečuje používateľ sám. Do tohoto pamäťového priestoru je možné uložiť rôzne programy podľa potrieb používateľa, buď vlastné alebo profesionálne produkty, ako napr. vývojový systém MRS-2. Výrobca dodáva aj špeciálne upravený ROM-modul, označený PMD 40, ktorý slúži na programovanie pamätí typu EPROM.

## 2.2. Konfigurácia PMD 85-3 a jeho obsluha

Základná konfigurácia, ktorá vyhovuje na bežné používanie mikropočítača, je na obr. 2.2. Je zostavená z nasledujúcich častí:

- mikropočítač PMD 85-3,
- štandardný čiernobiely alebo farebný TV prijímač so systémom PAL naladený na 36. kanál IV. TV pásma, prípadne TV monitor so vstupom RGB alebo VIDEO,
- napájací zdroj typu PMD 10, prípadne iný, ktorý vyhovuje podmienkam uvedeným v kapitole 1.1,
- štandardný kazetový magnetofón.



Obr.2.2. Základná konfigurácia PMD 85-3

Jednotlivé časti zostavy sú s mikropočítačom pospájané prepájacími šnúrami, z ktorých iba prepájacia šnúra k TV prijímaču je súčasťou dodávky mikropočítača. Konektory na pripojenie napájacieho zdroja a TV prijímača, resp. TV monitora sú umiestnené v ľavej časti skrinky, konektor magnetofónu je v zadnej časti.

Pre prevádzku mikropočítača vyhovujú parametre bežných kancelárskych a laboratórnych priestorov.

Mikropočítač sa uvedie do činnosti zapnutím napájacieho napätia zdroja. Na TV prijímači je potrebné nastaviť príslušný kanál a doladiť ho tak, aby výpis na obrazovke mal optimálnu čitateľnosť. Úvodný text výpisu je závislý od toho, či je v skrinke mikropočítača zasunutý ROM-modul. V prípade, že mikropočítač bol zapnutý bez tohoto modulu, je v činnosti základné programové vybavenie MONITOR, ktoré sa ohlásí výpisom:

```
++ OS ready ++
```

Ak bol ROM-modul vložený, uvedie sa do činnosti interpretér jazyka BASIC-G, ktorý sa ohlásí výpisom:

```
BASIC G /V3.0
```

Ďalšia činnosť používateľa spočíva v komunikácii s aktívnym programom. Opisy príkazov pre komunikáciu s PMD 85-3 sa nachádzajú v kapitolách 4 a 5.

Manipulovať s ROM-modulom, t. j. zasúvať ho a vyberať, je možné iba pri vypnutom zdroji mikropočítača. V opačnom prípade môže dôjsť k poškodeniu pamäti v ROM-module.



## 3. Komunikácia používateľa s PMD 85-3

Základná komunikácia používateľa s PMD 85-3 spočíva v zadávaní príkazov mikropočítaču cez jeho klávesnicu a v odpovediach mikropočítača na zadané príkazy prostredníctvom zobrazovacej jednotky. Podrobnému opisu používania týchto dvoch vstupno-výstupných zariadení je venovaná táto kapitola.

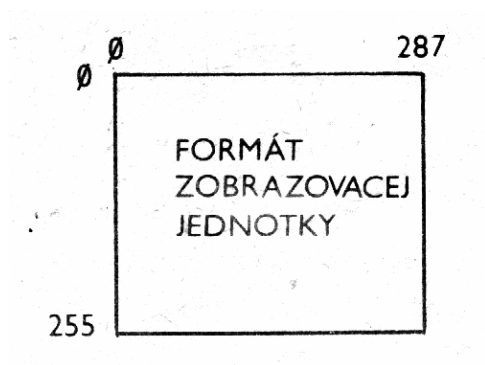
### 3.1. Zobrazovacia jednotka a spôsob zobrazovania

Zobrazovanie textových a grafických informácií sa v osobnom mikropočítači PMD 85-3 uskutočňuje prostredníctvom zobrazovacej jednotky (ZJ), vo funkcii ktorej môže byť pripojený štandardný TV prijímač alebo TV monitor.

Zobrazovacia jednotka dokáže zobrazovať informácie bod po bode v obdĺžnikovom priestore s rastrom 288 x 256 bodov (obr.3.1). Pretože jednotlivé príkazy jazyka BASIC-G pracujú s týmto priestorom odlišným spôsobom, zavedieme logické označenia rôznych častí pracovného priestoru zobrazovacej jednotky. Takáto časť pracovného priestoru s definovanými vlastnosťami a rozmermi sa bude nazývať okno (anglicky window).

Priestor zobrazovacej jednotky 288 x 256 bodov bude predefinovaný týmito oknami:

- okno binárnej grafiky (BG),
- okno symbolovo orientovanej grafiky (SYOG),
- okno súradnicovo orientovanej grafiky (SUOG),
- okno dialógového riadka (DR).



Obr. 3.1

### 3.1.1. Okno binárnej grafiky

Poloha tohoto okna je na zobrazovacej jednotke vymedzená takto (obr. 3.2):

linky : 0 - 242

stĺpce: 0 - 288

Práca s týmto oknom predstavuje elementárnu činnosť pri znázorňovaní informácií na ZJ. Pomocou dvoch príkazov jazyka BASIC-G, ktoré sú na túto činnosť prístupné, je možné (s určitými obmedzeniami) fyzicky ovládať jednotlivé body rastra ZJ.

Okno BG je logicky rozdelené na 243 liniek a 48 stĺpcov, pričom každý stĺpec je reprezentovaný 6 bodmi ( $6 \times 48 = 288$ , čo je šírka rastra). V každom jednom políčku, ktoré sa skladá zo 6 bodov, je možné individuálne ovládať rozsvecovanie a zhášanie ktoréhokoľvek bodu na ZJ s takým obmedzením, že celá šestica bodov má spoločný atribút farby, resp. jas.

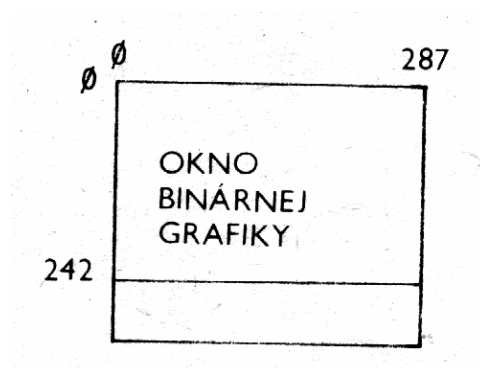
Polohu políčka na ZJ, s ktorým sa bude manipulovať, určuje kurzor okna BG. Jeho hodnotu je možné meniť príkazom BMOVE. Samotnú činnosť nad políčkom vykonáva príkaz BPLOT, ktorého parameter je 1-bajtový údaj, ktorý obsahuje informáciu o ovládaní jednotlivých bodov políčka. Význam jednotlivých bitov tohoto parametra je uvedený na obr. 3.3.

Horné dva bity (bit 6 a 7) obsahujú atribúty pre celú šesticu, bity 0 až 5 ovládajú rozsvetovanie a zhašovanie jednotlivých bodov šestice, 0 znamená zhasnutie, 1 znamená rozsvietenie.

Význam atribútov (6. a 7. bit)

7	6	5	4	3	2	1	0		VIDEO	RGB	ČB
								0 0	- biela	zelená	biela
								0 1	- zelená	červená	\\\\\\\\
								1 0	- červená	modrá	/////
								1 1	- hnedá	fialová	XXXXX
atribúty		ovládanie									
		bodov									

Obr. 3.3



Obr. 3.2

## 3.1.2. Okno symbolovo orientovanej grafiky

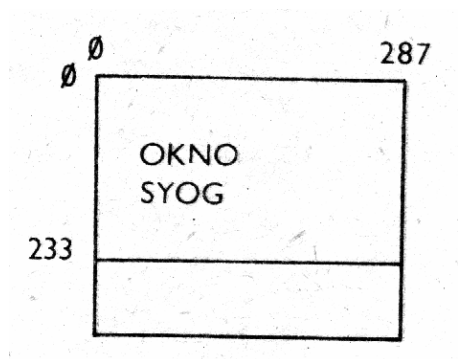
Poloha tohto okna sa v rastrí ZJ vymedzuje takto (obr. 3.4):

linky : 0 - 233

stĺpce: 0 - 287

Význam tohoto okna je predovšetkým v zobrazovaní textových informácií (písmená, číslice, interpunkčné znamienka, rôzne špeciálne symboly). Okno SYOG je logicky rozdelené na 26 riadkov a 48 stĺpcov, pričom každé políčko má rozmer 6x9 bodov, vrátane medziriadkovej medzery nad znakom. V tomto rastrí sa potom zobrazujú textové a špeciálne symboly, ktoré sú definované vo vnútornom kóde PMD 85-3 (pozri prílohu C). Príklad zobrazenia písmena E týmto spôsobom je na obr. 3.5.

Políčko, do ktorého bude zobrazený ďalší údaj, určuje kurzor okna SYOG, ktorý sa dá ovládať rôznymi príkazmi, medzi typické patrí PRINT.



Obr. 3.4

```
.....  
.xxxxx  
.x....  
.x....  
.xxxx.  
.x....  
.x....  
.xxxxx  
.....
```

Obr. 3.5

### 3.1.3. Okno súradnicovo orientovanej grafiky

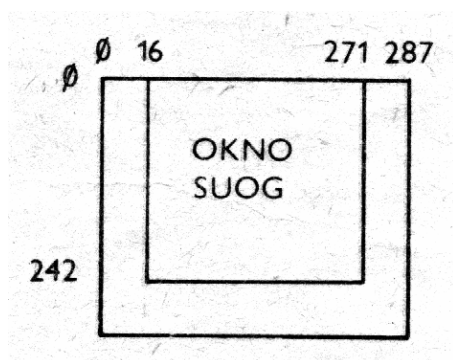
Poloha tohto okna sa na ZJ vymedzuje takto (obr. 3.6):

linky : 0 - 242

stĺpce: 16 - 271

Okno SUOG a množina príkazov, ktoré nad ním pracujú, dáva používateľovi PMD 85-3 skutočné možnosti práce s grafikou. Na rozdiel od binárnej grafiky, kde je nutné vykonávať veľa elementárnych operácií, táto grafika umožňuje zobraziť dva základné geometrické tvary: bod a úsečku. Na ich základe je potom možné "kresliť" rôzne motívy, ako sú kružnice, krivky, priebehy rôznych funkcií, vyfarbovať plochy geometrických útvarov a pod. Táto činnosť sa vykonáva vo zvolenej súradnicovej sústave a vo zvolenej mierke.

Určité obmedzenia vyplývajú z toho, že táto grafika je vlastne iba nadstavbou binárnej grafiky, a teda možnosť ovládania jednotlivých bodov ZJ je čiastočná. Tiež priradenie farby je možné urobiť len šesticou bodov. Ďalším obmedzením sú menšie rozmery tohto okna (256 x 243 bodov), čo vyplýva z toho, že spojenie dvoch ľubovoľných bodov vykonáva tzv. lineárny interpolátor, ktorého rýchlosť závisí na rozmeroch rastra, nad ktorým pracuje.



Obr. 3.6

### 3.1.4. Okno dialógového riadka

Toto okno má štandardnú pozíciu na zobrazovacej jednotke, v jeho dolnej časti (obr. 3.7). Je vymedzené takto:

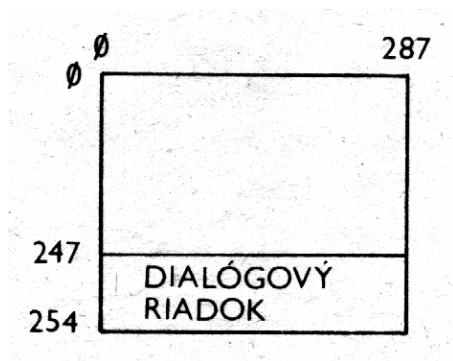
linky : 247 - 254

stĺpce: 0 - 287

Dialógový riadok má osobitný význam, pretože umožňuje:

- zobrazovať informácie vstupujúce do mikropočítača z klávesnice
- zobrazovať niektoré informácie vystupujúce z mikropočítača, ako napr. hlásenia o chybách

Dĺžka dialógového riadka je 80 znakov, počet súčasne viditeľných znakov je 48. Pomocou editačných klávesov (kap. 3.2.2) sa dá zobraziť aj jeho obsah za 48. znakom, resp. ľubovoľný 48-znakový úsek. Pozíciu v dialógovom riadku určuje kurzor dialógového riadka, ktorý je zobrazený značkou. Jeho pohyb je možné ovládať editačnými klávesmi.



Obr. 3.7

## 3.2. Klávesnica a vstup údajov

Klávesnica osobného mikropočítača PMD 85-3 je znázornená na obr. 3.8. Obsahuje tri druhy klávesov:

- znakové klávesy
- riadiace klávesy
- programové klávesy

Stlačenie klávesu sprevádza krátky akustický signál, ktorý je možné vypnúť. Pridržaním klávesu v stlačenej polohe sa docielí opakovanie jeho funkcie.

### 3.2.1. Znakové klávesy

Táto časť klávesnice sa podobá bežnému písaciemu stroju. Umožňuje zadávať všetky znaky slovenskej a českej abecedy (t. j. vrátane diakritických znamienok). Vnútorňá reprezentácia (kódovanie) znakov v mikropočítači PMD 85-3 je podľa modifikovanej kódovej tabuľky KOI 8 ČS-2 (ČSN 36 9100, pozri Prílohu C). Keďže klávesnica má obmedzený počet klávesov, zadanie všetkých znakov abecedy z kódovej tabuľky je možné týmto spôsobom:

veľké písmená:	stlačenie klávesu písmena
malé písmená:	stlačenie preradovača a klávesu písmena súčasne
malé písmená s diakritikou:	stlačenie klávesu STOP a klávesu písmena súčasne
veľké písmená s diakritikou:	stlačenie preradovača, klávesu STOP a klávesu a klávesu písmena súčasne



K <sub>0</sub>	K <sub>1</sub>	K <sub>2</sub>	K <sub>3</sub>	K <sub>4</sub>	K <sub>5</sub>	K <sub>6</sub>	K <sub>7</sub>	K <sub>8</sub>	K <sub>9</sub>	K <sub>10</sub>	K <sub>11</sub>	WRK	C-D	RCL	RST	
!	"	#	\$	%	&	'	(	)	-	=	}	{	PTL INS	DEL	CLR	
1	2	3	4	5	6	7	8	9	0	-						
Q	W	E	R	T	Z	U	I	O	P	'	@	^	/	←	↖	→
A	S	D	F	G	H	J	K	L	+	*	*	:	]	[	←	END →
	↑	Y	X	C	V	B	N	M	<	,	>	.	?	/	↑	STOP EOL EOL

Obr. 3.8 Pohľad na klávesnicu PMD 85-3

K <sub>0</sub>	K <sub>1</sub>	K <sub>2</sub>	K <sub>3</sub>	K <sub>4</sub>	K <sub>5</sub>	K <sub>6</sub>	K <sub>7</sub>	K <sub>8</sub>	K <sub>9</sub>	K <sub>10</sub>	K <sub>11</sub>	WRK	C-D	RCL	RST
1 ! 2 "	3 #	4 \$	5 %	6 &	7 ' 8 ( 9 )	ø -	= {	PTL	DEL	CLR					
Q q	W w	E e	R r	T t	Z z	U u	I i	O o	P p	@	\ ^				
ä	Ä ä	É é	Ê ê	Ë ë	Ž ž	Ú ú	Í í	Ó ó	Ô ô	Š š	π				
A a	S s	D d	F f	G g	H h	J j	K k	L l	+	;	*	:			
á	Á á	š	ř	ŕ	ü	ű	í	í	í					END	→
Y y	Y y	X x	X x	C c	V v	B b	N n	M m	<	.	?	/		STOP	EOL
↑	↑	á	á	č	β	α	ň	ö						EOL	

Obr. 3.9 Spôsob dosiahnutia jednotlivých znakov

Spôsob zobrazenia všetkých znakov je znázornený v tab. 3.1, kde ↑ označuje preradovač. Súčasné stlačenie klávesov STOP a ↑ slúži ako prepínač veľkých a malých písmen. Zapnutie trvalého zobrazenia malých písmen signalizuje rozsvietenie červenej diódy LED.

znak	spôsob dosiahnutia	kód	znak	spôsob dosiahnutia	kód
Á	STOP ↑ A	E1	á	STOP A	C1
Ä	STOP ↑ Q	F1	ä	STOP Q	D1
À	STOP ↑ X	F8	à	STOP X	D8
Č	STOP ↑ C	E3	č	STOP C	C3
Ď	STOP ↑ D	E4	ď	STOP D	C4
Ě	STOP ↑ E	E5	ě	STOP E	C5
É	STOP ↑ W	F7	é	STOP W	D7
Í	STOP ↑ I	E9	í	STOP I	C9
Ľ	STOP ↑ L	EC	l	STOP L	CC
Í	STOP ↑ K	EB	í	STOP K	CB
Ň	STOP ↑ N	EE	ň	STOP N	CE
Ó	STOP ↑ O	EF	ó	STOP O	CF
Ô	STOP ↑ P	F0	ô	STOP P	D0
Ö	STOP ↑ M	ED	ö	STOP M	CD
Ř	STOP ↑ R	F2	ř	STOP R	D2
Ř	STOP ↑ F	E6	ř	STOP F	C6
Š	STOP ↑ S	F3	š	STOP S	D3
Ť	STOP ↑ T	F4	ť	STOP T	D4
Ú	STOP ↑ U	F5	ú	STOP U	D5
Ů	STOP ↑ J	EA	ů	STOP J	CA
Ů	STOP ↑ H	E8	ü	STOP H	C8
Ý	STOP ↑ Y	F9	ý	STOP Y	D9
Ž	STOP ↑ Z	FA	ž	STOP Z	DA









Tab. 3.1 Spôsob zobrazenia písmen s diakritickými znamienkami

znak	spôsob dosiahnutia	kód
------	-----------------------	-----

alfa	STOP B	C2
beta	STOP V	D6
gama	STOP ↑ B	E2
delta	STOP ↑ V	F6
integrál	STOP @	C0
pi	STOP ↑ @	E0

Tab. 3.2 Spôsob zobrazenia špeciálnych znakov






znak	spôsob dosiahnutia	kód	znak	spôsob dosiahnutia	kód
------	-----------------------	-----	------	-----------------------	-----

	STOP ↑ G	E7		STOP _	DF
	STOP G	C7		STOP ↑ }	FD
	STOP [	DB		STOP ↑ WRK	FB
	STOP \	DC		STOP ↑ WRK	FC
	STOP ↑ [	DD		STOP ↑ WRK	FE
	STOP ↑ \	DE		STOP ↑ WRK	FF

Tab. 3.3 Spôsob zobrazenia špeciálnych grafických znakov

## 3.2.2. Riadiace klávesy

Tieto klávesy sú umiestnené v pravej časti klávesnice. Pomocou nich je možné zabezpečiť editovanie textu v dialógovom riadku a vyvolať rôzne systémové činnosti mikropočítača. Niektoré činnosti sa aktivizujú stlačením viacerých klávesov súčasne. Ich vyčerpávajúci opis je v nasledujúcej tabuľke:

	Posun kurzora DR o jeden znak vľavo
	Posun kurzora DR o jeden znak vpravo
	Posun kurzora na začiatok DR
END	Posun kurzora za posledný nemedzerový znak v DR
	Posun viditeľnej časti DR doprava
	Posun viditeľnej časti DR doľava
INS	Vytvorenie priestoru na vloženie znaku na pozícii kurzora DR
DEL	Vymazanie znaku na pozícii kurzora DR. Zvyšok DR vpravo od kurzora sa posunie o jednu pozíciu doľava
CLR	Vymazanie celého obsahu DR



CLR

Vymazanie obsahu DR od pozície kurzora do konca



Preraďovač, preradenie činnosti, napr. horné znaky klávesnice alebo pozastavenie zobrazovania na zobrazovaciu jednotku

EOL

Ukončenie vkladania znakov do DR a odoslanie správy

STOP {

Zapnutie alebo vypnutie trvalého zobrazenia malých písmen, červená dióda LED indikuje stav zapnutia

RCL

V DR sa zobrazí obsah posledného odoslaného riadka od aktuálnej pozície KDR



RST

Realizuje sa inicializácia a štart systému

C-D

Zobrazí sa posledné systémové hlásenie v DR

STOP

Prerušenie vykonávania programu



WRK

Zobrazenie znakov, ktoré sa vyskytujú v kódovej tabuľke za znakom, na ktorý práve ukazuje KDR. Ak je napr. KDR na pozícii medzera, po stlačení týchto klávesov sa budú postupne zobrazovať znaky ! " \$ % atď. v poradí, ktoré je dané ich kódom. Týmto spôsobom je možné zobraziť aj tie znaky, ktoré sa nevyskytujú na klávesnici PMD 85-3 (napr. vlnovka).



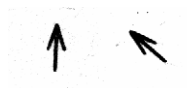
C-D

Prechod do režimu terminál. Má význam vtedy, keď je PMD 85-3 pripojený na počítač radu SMEP (kap. 5)

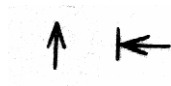


DEL

Štart programu umiestneného na adrese 0000. Je ekvivalentný príkazu JUMP 0000. Umožňuje napr. opätovný štart interpretera jazyka BASIC-G



Zmazanie obsahu všetkých okien obrazovky okrem DR



Zmena podkladu podľa farby DR a súčasné vymazanie údajov zobrazených na obrazovke okrem DR. Pri prvom stlačení je podklad farebný a zobrazované znaky inverzné, pri druhom stlačení naopak



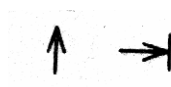
END

Zapnutie alebo vypnutie akustickej signalizácie stlačenia klávesu. Ak je signalizácia vypnutá, svieti ľavá dióda LED žltej farby



RCL

Prechod do programu monitor (kap.5)



Prepínač farby písma v DR



PTL

Prepínač kopírovania obsahu DR do okna SYOG



### 3.3. Programové klávesy

Programové klávesy sú určené na rýchle a pohodlné písanie textov, ktoré sa často opakujú. Každý programový kláves umožňuje uchovanie textu v dĺžke max. 80 znakov. V mikropočítači PMD 85-3 je k dispozícii 12 programových klávesov, ktoré sú označené K0, K1, K2, .... K11. Funkciu každého programového klávesu je možné zdvojiť využitím preraďovača.

#### 3.3.1. Priradenie hodnoty programovému klávesu

Táto činnosť sa realizuje v troch krokoch:

- príprava textu v dialógovom riadku,
- stlačenie klávesu WRK,
- stlačenie zvoleného programového klávesu alebo preraďovača a programového klávesu súčasne.

Po jej ukončení je obsah dialógového riadka uložený v operačnej pamäti od adresy zodpovedajúcej zvolenému programovému klávesu.

Poznámky:

- Poloha kurzora DR určuje koniec ukladaneho textu.
- Po stlačení klávesu WRK nasleduje stlačenie niektorého z programových klávesov. Ak sa stlačí iný kláves, ruší sa tento režim, t. j. uloženie sa nevykoná.
- Je možné maximálne uložiť 1920 znakov textu. Tento počet získame výpočtom takto: 12 klávesov po 80 znakov v režime bez preraďovača a v režime s preraďovačom.

## 3.3.2. Sprístupnenie obsahu programového klávesu

Táto činnosť sa vykonáva týmto spôsobom:

- stlačením príslušného programového klávesu,
- stlačením preraďovača a príslušného programového klávesu súčasne.

Poznámka:

- Obsah programového klávesu sa zobrazí v dialógovom riadku od aktuálnej pozície kurzora DR.

Po zapnutí osobného mikropočítača PMD 85-3 a následnej inicializácii sa automaticky nastaví programové klávesy na tieto hodnoty:

K0	RUN	↑	K0	SCALE_
K1	LIST_	↑	K1	PLOT_
K2	LLIST_	↑	K2	MOVE_
K3	INPUT_	↑	K3	LABEL_
K4	PRINT_	↑	K4	CHR\$ (
K5	GOTO_	↑	K5	MID\$ (
K6	GOSUB_	↑	K6	LEFT\$ (
K7	RETURN	↑	K7	RIGHT\$ (
K8	FOR_	↑	K8	INT (
K9	NEXT_	↑	K9	STR\$ (
K10	THEN_	↑	K10	DISP_
K11	WINDOW	↑	K11	_STEP_

K nastaveniu programových kláves nedôjde, pokiaľ sa po zapnutí počítača počas inicializácie drží kláves STOP.

## 4. BASIC-G

V tejto kapitole, ktorá tvorí centrálnu časť príručky, sú opísané všetky príkazy a funkcie jazyka BASIC-G/V3.0, ktorý je implementovaný v mikropočítači PMD 85-3. Príkazy a funkcie sú uvedené v abecednom poradí ich názvov, teda nie sú členené podľa ich logického významu. Predpokladá sa, že používateľ tejto príručky pozná základy programovania v jazyku BASIC, a preto si táto príručka nedáva za cieľ naučiť programovať.

### 4.1. Úvodné pojmy

Interpreter jazyka BASIC-G rozlišuje dva spôsoby práce:

- priamy mód
- programový mód

Priamy mód sa vyznačuje tým, že interpreter vykoná príkaz, ktorý bol zapísaný do dialógového riadka, ihneď po jeho odoslaní (stlačenie klávesu EOL). Do jedného riadka je možné zapísať aj viac príkazov, je však nutné ich od seba oddeliť znakom dvojbodka (:). Príklady:

```
PRINT "Tesla"
```

```
Tesla
```

```
FOR I=8 TO 10:PRINT 2^I:NEXT
```

```
256
```

```
512
```

```
1024
```

Pri programovom móde nie sú príkazy interpretované ihneď, ale sa ukladajú do operačnej pamäte a vykonajú sa až po zadaní príkazu RUN. Na rozdiel od priameho módu sa príkazy v programovom móde zadávajú tak, že každý riadok musí byť na jeho začiatku označený celým číslom z intervalu 0 až 32769, ktoré jednoznačne určuje poradie riadka v programe. Z tohoto dôvodu nie je nutné zadávať riadky v tom poradí, v akom sa budú neskôr vykonávať, resp. je možné vkladať nové riadky do hotového programu, ak to číslovanie riadkov dovolí. Podobne ako v priamom móde je možné zadať do jedného riadka viac príkazov, ich počet je obmedzený iba dĺžkou DR (80 znakov). Príklad:

```
100 LET A=5
200 PRINT A,B,A+B
150 LET B=8
RUN
5           8           13
```

Všetky príkazy a funkcie opísané v tejto kapitole sú použiteľné v programovom móde, v priamom móde nie je možné použiť príkazy DEF FNC a INPUT.

Okrem príkazov a funkcií sú jazykom BASIC-G spracovávané nasledujúce objekty:

- konštanty
- premenné
- výrazy

## Konštanty

Interpreter BASIC-G rozlišuje dva typy konštánt: numerické a reťazcové. Numerická konštantá je reprezentovaná číslom z rozsahu  $10^{-38}$  až  $10^{+38}$  a môže byť v niektorom z nasledujúcich tvarov:

celočíselný	napr.	-123, 987654
desatinný	napr.	56.432, -.6767
exponenciálny	napr.	.62E-3, 3.614E+06
hexadecimálny	napr.	'C1FE, 'FFF0

Reťazcová konštantá je postupnosť 0 až 255 znakov uzavretých v úvodzovkách ("):

```
" "                prázdny reťazec (dĺžka = 0)
"Bratislava"
".33E-8"
```

## Premenné

Podobne ako v predchádzajúcom prípade rozlišujeme iba numerické a reťazcové premenné. Identifikátor premennej je vo všeobecnosti ľubovoľná postupnosť alfanumerických znakov začínajúca písmenom. Na rozlíšenie premenných berie BASIC-G do úvahy iba prvé dva znaky, pričom rozlišuje medzi veľkými a malými písmenami. Napr. identifikátory VYSTUP a VYKON reprezentujú jednu premennú, identifikátory NAVRAT a Navrat dve rôzne premenné. Identifikátory reťazcových premenných sú ukončené znakom \$, napr. RETAZ\$, SPRAVA\$, A\$. BASIC-G rozlišuje medzi identifikátorom numerickej a reťazcovej premennej, t.j. XY a XY\$ sú rôzne premenné.

Špeciálnou skupinou premenných sú polia. Pole je skupina premenných rovnakého typu, ktoré sa označujú rovnakým identifikátorom a rozlišujú indexami v zátvorkách.

Napr.     RIADOK(7), ABC(4,0,3), ZNAK\$(2,1)

Počet indexov udáva rozmer poľa, BASIC-G pracuje s jedno-, dvoj- a trojrozmernými poliami. Index poľa je nezáporné celé číslo, vo všeobecnosti aritmetický výraz, z ktorého hodnoty sa v prípade potreby zoberie iba celočíselná časť. BASIC-G rozlišuje medzi jednoduchou premennou a indexovanou premennou s tým istým identifikátorom. Bez predchádzajúceho dimenzovania je možné použiť premennú ako pole s indexami 0 až 10.

Napr.     ABC(10,8,2)=10\*X+8\*Y+2\*Z  
           R\$(8)=R\$(8)+CHR\$(21)

V prípade použitia rozsiahlejších polí (to znamená, že jeden z indexov má hornú hranicu väčšiu ako 10) je nutné premennú deklarovať príkazom DIM.

Napr.     DIM MATICA(30,25)  
           DIM ZOZNAM\$(65)

## Výrazy

Výrazy slúžia na vyjadrenie výpočtov, funkcií, vzťahov a pod. Argumenty väčšiny príkazov a funkcií môžu byť výrazy. Výrazy konštruujeme z konštánt, premenných a funkcií pomocou operátorov a zátvoriek. Operátory rozdeľujeme na:

aritmetické	^   *   /   +   -
relačné	=   <   >   <=   >=   <>
logické	NOT   AND   OR

Vyhodnocovanie aritmetických výrazov sa uskutočňuje podľa nasledujúcich priorít:

1. výrazy v zátvorkách	( )
2. funkcie	
3. umocnenie	$\wedge$
4. polarita	+ -
5. násobenie	*
delenie	/
6. sčítanie	+
odčítanie	-
7. relácia "je rovný"	=
relácia "je menší"	<
relácia "je väčší"	>
relácia "je rovný alebo menší"	<=
relácia "je rovný alebo väčší"	>=
relácia "nerovná sa"	<>
8. dvojkový komplement	NOT
9. logický súčin	AND
logický súčet	OR

Použitie zátvoriek má teda najvyššiu prioritu, logické operátory AND a OR najnižšiu. Pri rovnakej priorite operátorov sa výraz vyhodnocuje zľava doprava.

Výsledkom vyhodnotenia výrazu je numerická hodnota, reťazec alebo pravdivostná hodnota. Pravdivostná hodnota je reprezentovaná hodnotami -1 pre "pravdu" a 0 pre "nepravdu" (obdoba logických konštánt TRUE a FALSE u iných programovacích jazykov).

Vzhľadom na uvedené tri typy hodnôt rozlišujeme aritmetický, reťazcový a logický výraz.



Výraz:	Typ výsledku po vyhodnotení:
aritmetický	numerická hodnota
reťazcový	reťazec
logický	pravdivostná hodnota

Pri vytváraní výrazov sa riadime jednoduchými pravidlami:

1. Aritmetické výrazy tvoríme z objektov numerického typu (konštanty, premenné, funkcie, výrazy) pomocou aritmetických operátorov.

Napr.  $(1 - \cos(2*U))/2 + (1 + \cos(2*U))/2$   
 $((3.1415*V)*(R(1)^2 + R(1)*R(2) + R(2)^2))/3$

2. Reťazcové výrazy tvoríme z reťazcových konštánt, premenných, funkcií a výrazov spojených aritmetickým operátorom +.

Napr. NAZOV\$ + " fakulta"  
A\$ + MID\$(X\$,1,J) + "?"

3. Logické výrazy dostaneme buď spojením dvoch aritmetických príp. dvoch reťazcových výrazov pomocou relačných operátorov alebo spojením dvoch logických výrazov pomocou logických operátorov.

Napr.  $(B^2 - 4*A*C) < 0$   
RIGHT\$(MENO\$,3) = "OVA"  
 $\sin(2*X) < \sqrt{2}/2$  AND LEN(R\$)=5 OR LEN(R\$)=6

## 4.2. Opis príkazov BASIC-G

Každý príkaz a funkcia má krátky opis najdôležitejších vlastností, formu zápisu a význam jednotlivých parametrov. V prípade potreby je upozornené na významné zvláštnosti spojené s používaním konkrétneho príkazu alebo funkcie. Na konci opisu je vždy uvedený krátky a typický príklad použitia príkazu, resp. funkcie.

Pri definovaní parametrov jednotlivých príkazov a funkcií sú použité určité symboly a pomenovania, ktoré sa bežne používajú pri definíciách programovacích jazykov. V tejto príručke nie sú použité všetky možné tvary týchto symbolov, ale len tie najpoužívanéjšie.

Kvôli rýchlejšej orientácii je uvedený na nasledujúcej strane prehľad všetkých príkazov a funkcií jazyka BASIC-G, ktoré sú usporiadané do tabuľky na základe ich logických významov.

EDITOVANIE		NEW, AUTO, LIST, LLIST, REN	
SPUSTENIE PROGRAMU		RUN, GOTO, CONT	
(PO) ZASTAVENIE		STOP, PAUSE, ?, END, WAIT	
POZNÁMKA V PROGRAME		REM	
PREMENNÉ	PRIRADENIE	LET, CLEAR	
	POLIA	DIM, CLEAR, DSAVE, DLOAD	
RIADENIE TOKU PROGRAMU	SKOK	GOTO	
	ROZHODOVANIE	IF-THEN	
	CYKLUS	FOR-TO-NEXT-STEP	
	VETVENIE	ON-GOTO, ON-GOSUB, ON-ERR	
BASIC- FUNKCIE	MATEMATICKÉ	ABS, INT, SGN, RND, SQR, LOG, EXP SIN, COS, TAN, ATN	
	REŤAZCOVÉ	CHR\$, ASC, STR\$, VAL, LEFT\$, RIGHT\$, MID\$, LEN, HEX\$	
	OSTATNÉ	FRE, ADR, INKEY, TAB, SPC, AT, INK PEEK, USR, INP, BIT, STATUS, POS	
REŽIM GONIOM. FUNKCIÍ		DEG, RAD	
VYTVORENIE PROCEDÚRY	PODPROGRAM	GOSUB, RETURN	
	FUNKCIA	DEF FNC	
VSTUP	KLÁVESNICA	INPUT, INKEY, ?, DISP, PAUSE	
	PROGRAM	READ, DATA, RESTORE	
	MAGNETOFÓN	LOAD, DLOAD, CHECK	
	KANÁL	ENTER, INPUT#, INP	
V Ý S T U P	TEXTOVÝ	OKNO SYOG	GCLEAR, GCLEARA, WINDOWA, PRINT, AT, INK, TAB, SPC
		DR	DISP, ?, TAB, SPC
	GRAFICKÝ	OKNO SUOG	GCLEAR, GCLEARG, WINDOWG, SCALE, MOVE, AXES, PLOT LABEL, FILL, PEN
		OKNO BG	GCLEAR, BMOVE, BPLOT
	AKUSTICKÝ		BEEP
	MAGNETOFÓN		SAVE, DSAVE
	TLAČIAREŇ		CONTROL, OUTPUT, PRINT#, LIST#, LLIST#
	KANÁL		CONTROL, PRINT#, OUT, OUTPUT
NASTAVENIE V/V KANÁLU		CONTROL, STATUS, BIT, INP, OUT, WAIT	
PRÁCA S PAMÄŤOU		PEEK, POKE, APEEK, APOKE, ROM, ADR, CLEAR	
POUŽITIE STROJ. KÓDU		USR, CODE, ROM	

# ABS

Funkcia ABS

Opis: Vracia absolútnu hodnotu parametra.

Formát: ABS(e)

Parameter: e je numerická konštanta, numerická premenná alebo aritmetický výraz.

Príklad:

Výpis hodnoty premennej A a jej absolútnej hodnoty:

```
100 A = -5
110 PRINT A;ABS(A)
RUN
-5 5
```

# ADR

Funkcia ADR

Opis: Vracia adresu operačnej pamäte, na ktorej je uložený parameter.

Formát: ADR(v)

Parameter: v je číselná alebo reťazcová premenná.

Poznámka: Pri reťazcovej premennej funkcia vracia adresu 4-bajtovej informácie o premennej (1. bajt - dĺžka reťazca (0..255), 2. bajt - nepoužitý, 3. a 4. bajt - adresa začiatku reťazca).

Príklad:

Tento fragment programu vypíše obsah premennej A\$:

```
100 A$="TEXT" : DLZKA=PEEK(ADR(A$))
120 A=APEEK(ADR(A$)+2)
130 FOR I=1 TO DLZKA
140 PRINT CHR$(PEEK(A+I-1))
150 NEXT I
RUN
T
E
X
T
```

Príklad:

Rezervovanie oblasti 4000 bajtov a uloženie adresy jej začiatku do premennej AD:

```
1000 DIM P(999)
1010 AD=ADR(P(0))
```

# APEEK

## Funkcia APEEK

**Opis:** Vracia hodnotu dvoch po sebe nasledujúcich bajtov v operačnej pamäti (chápu sa ako 16-bitové číslo so znamienkom).

**Formát:** APEEK(adr)

**Parameter:** adr je výraz, ktorého hodnota reprezentuje adresu v operačnej pamäti.

**Poznámka:** APEEK nahrádza funkciu PEEK, použitú na dve po sebe idúce adresy.

### Príklad:

Do premennej A uložíme adresu začiatku reťazca A\$ (porovnajte s príkladom pre funkciu ADR):

```
120 A=APEEK(ADR(A$)+2)
```

# APOKE

Príkaz APOKE

Opis: Slúži pre zápis 2-bajtovej hodnoty na zadanú adresu v operačnej pamäti.

Formát: APOKE adr,z

Parametre: adr je výraz, ktorý sa vyhodnotí ako adresa.  
z je zoznam výrazov s hodnotou z intervalu  
-32768..32767, t.j. sú to dvojбайtové hodnoty.

Príklad:

Do štyroch po sebe idúcich bajtov premennej A sa zapíšu hodnoty 0 a -32256, t.j. vnútorná reprezentácia čísla 2 (nová hodnota premennej A):

```
100 A=1
110 AD=ADR(A)
120 V1=APEEK(AD)
130 V2=APEEK(AD+2)
140 PRINT AD;V1;V2;A
150 APOKE AD,0,-32256
160 V1=APEEK(AD)
170 V2=APEEK(AD+2)
180 PRINT AD;V1;V2;A
RUN
9357 0 -32512 1
9357 0 -32256 2
```

# ASC

## Funkcia ASC

**Opis:** Vracia hodnotu prvého znaku zadaného reťazca, ktorá je jeho reprezentáciou vo vnútornom kóde PMD 85-3, prekonvertovaná na decimálne číslo.

**Formát:** ASC(a\$)

**Parameter:** a\$ je reťazcová konštanta alebo reťazcový výraz.

### Príklad:

Výpis prvého znaku reťazca TR\$, prekonvertovaného na decimálne číslo. (Reprezentácia znaku "A" vo vnútornom kóde PMD 85-3 je 41H (pozri prílohu C), dekadický ekvivalent je 65).

```
100 TR$="ABCDEF"
110 PRINT TR$;ASC(TR$)
RUN
ABCDEF 65
```



# ATN

Funkcia ATN

Opis: Vracia arctangens parametra.

Formát: ATN(e)

Parameter: e je numerická konštanta, numerická premenná alebo aritmetický výraz.

Príklad:

Výpis hodnoty arctangens druhej odmocniny 3 v radiánoch a stupňoch:

```
100 PRINT ATN(SQR(3))
110 RD=57.295779
120 PRINT RD*ATN(SQR(3))
RUN
1.0472
60
```

# AUTO

Organizačný príkaz AUTO

Opis: Umožňuje automatické číslovanie riadkov programu.

Formát: AUTO m,n

Parametre: m je počiatočná hodnota číslovania riadkov.  
n je prírastok číslovania.

Poznámka: Ak sa n nezadá, prírastok sa nastaví na poslednú aktuálnu hodnotu (ak predtým nebola zadaná, platí 10). Ak sa nezadá žiaden parameter, čísluje sa od poslednej aktuálnej hodnoty (ak predtým nebola zadaná, čísluje sa od 10). Čísla riadkov sú z intervalu 0..[32769](#).  
Prerušenie režimu AUTO je možné postupným stlačením kláves CLR a EOL.

Príklad:

Čísla riadkov sa majú začínať od 300 a narastať po 15:

AUTO 300,15

# AXES

Opis: Slúži na zobrazenie súradných osí s priesečníkom v zadanom bode.

Formát: AXES m,n

Parametre: m je x-ová súradnica zadaného bodu.  
n je y-ová súradnica zadaného bodu.

Príklad:

V mierke -1,1,-1,1 sa majú v bode 0,0 vykresliť osi x,y:

```
100 SCALE -1,1,-1,1
110 AXES 0,0
```

Stred súradníc bude uprostred obrazovky. (Pozri aj príkaz SCALE.)

Príklad:

V mierke 0,10,0,10 sa majú v bode 0,0 vykresliť osi x,y:

```
100 SCALE 0,10,0,10
110 AXES 0,0
```

Stred súradníc bude v ľavom dolnom rohu obrazovky (pozri aj príkaz SCALE).

# BEEP

## Príkaz BEEP

Opis: Mikropočítač sa ohlásí akustickým signálom so zadanou frekvenciou a dĺžkou tónu.

Formát: BEEP e1,e2

Parametre: e1 je výraz pre frekvenciu tónu v intervale 20..32767 (Hz).  
e2 je výraz pre dĺžku tónu v intervale 0..255.

## Príklad:

Program, ktorý zahrá základnú oktávu zdola nahor a potom zhora nadol:

```
100 DIM FR(8)
110 DL = 8
120 FOR I=1 TO 8
130 READ FR(I)
140 NEXT I
150 DATA 261.63,293.67,329.63,349.23
160 DATA 392.00,440.00,493.89,523.25
170 PAUSE 10
180 FOR I=1 TO 8
190 BEEP FR(I),DL
200 PAUSE 1
210 NEXT I
220 PAUSE 5
230 FOR I=8 TO 1 STBP -1
240 BEEP FR(I),DL
250 PAUSE 1
260 NEXT I
270 PAUSE 20
RUN
```

# BIT

Funkcia BIT

Opis: Vracia hodnotu zadaného bitu zo zadaného bajtu.

Formát: BIT v,i

Parametre: v je premenná alebo konštanta, ktorej hodnota je z intervalu (0..255)  
i je číslo daného bitu (0..7), pri číslovaní sprava doľava.

Príklad:

Výpis dekadického, hexadecimálneho a binárneho reprezentácie znaku "E" vo vnútornom kóde PMD 85-3 (pozri prílohu C):

```
100 A$="E"
110 A=ASC(A$)
120 PRINT A$;A;HEX$(A);" ";
130 FOR I=7 TO 0 STEP -1
140 PRINT BIT A,I;
150 NEXT I
160 PRINT
RUN
E 69 0045 0 1 0 0 0 1 0 1
```

# BMOVE

Grafický príkaz BMOVE

Opis:           Nastavuje kurzor binárnej grafiky do zadaného stĺpca a linky v okne binárnej grafiky.

Formát:        BMOVE e1,e2

Parametre:    e1    je výraz, určujúci stĺpec nastavenia kurzora (stĺpce z intervalu 0..47).  
                e2    je výraz, určujúci linku nastavenia kurzora (linka z intervalu 0..242).

Poznámka:     Pozícia kurzora sa nastaví absolútne, t.j. bez ohľadu na jestvujúcu mierku. Bod 0,0 je v ľavom hornom rohu obrazovky.

Príklad:

```
100 GCLEAR
110 SCALE 0,10,0,10
120 BMOVE 23,141
130 A$=CHR$(12)
140 B$=CHR$(18)
150 C$=CHR$(33)
160 P$=A$+B$+C$+C$+B$+A$
170 BPLOT P$,1
180 END
```

Napriek existujúcej mierke sa kurzor nastaví doprostred obrazovky a vykreslia sa zadané znaky.

# BPLOT

Grafický príkaz BPLOT

**Opis:** Slúži na kreslenie v okne binárnej grafiky. Každý znak zadaného reťazca sa zobrazí do 6 bodov vedľa seba podľa binárnej reprezentácie znaku. Udá sa aj počet znakov v jednom riadku.

**Formát:** BPLOT a\$,n

**Parametre:** a\$ je reťazcová konštanta alebo reťazcový výraz.  
n je počet znakov, ktoré sa zobrazia v jednom riadku (v intervale 0..dĺžka reťazca a\$). Pre n=0 sa binárne zobrazí každý znak reťazca a\$ v jednom riadku.

**Poznámka:** Kreslí sa od pozície kurzora, nastavenej príkazom BMOVE. Pre n=1 sa zobrazí každý znak na novom riadku, pre n=2 sa zobrazia prvé dva znaky v prvom riadku, ďalšie dva v druhom riadku, atď. Pri n=3 sa zobrazujú po tri znaky v riadku, atď.

**Príklad:**

Potrebuje vykresliť trojuholník so základňou 6 bodov a s výškou 3 body. V troch riadkoch po sebe majú teda svietiť postupne 2, 4 a 6 bodov. Znak, ktoré majú potrebnú binárnu reprezentáciu, sú: 001100 (decimálne 12), 011110 (decimálne 30) a 111111 (decimálne 63).

```
100 T$ = CHR$(12) + CHR$(30) + CHR$(63)
110 BMOVE 2,55
120 BPLOT T$,1
```

Trojuholník bude vykreslený v ľavom hornom rohu obrazovky.

# CHECK

Organizačný príkaz CHECK

Opis: Preverí nahrávku záznamu súboru na magnetofónovej kazete podľa kontrolného súčtu.

Formát: CHECK f

Parameter: f číslo súboru na magnetofónovej kazete.

Poznámka: Príkaz je vhodný aj na vyhľadanie konca záznamu pre ďalší zápis na kazetu (nemení obsah pamäte programu). Ak je súbor nahratý zle, príkaz ohlásí chybu č. 23 (File error).

Pre f=0 preverí najbližší záznam na kazete.

Príklad:

Uloženie programu na magnetofónovú kazetu a kontrola správnosti zápisu:

SAVE 11

CHECK 11

Program sa uloží na magnetofónovú kazetu ako súbor s číslom 11 a preverí sa správnosť zápisu.



# CHR\$

Funkcia CHR\$

Opis: Z hodnoty argumentu vracia jeho znakový tvar daný vnútorným kódom PMD 85-3.

Formát: CHR\$(e)

Parameter: e je numerická konštanta, numerická premenná alebo aritmetický výraz z intervalu (0..255).

Poznámka: Funkcia robí prevod dekadickéj hodnoty na znak vo vnútornom kóde PMD 85-3 (pozri prílohu C).

Príklad:

Výpis časti vnútorného kódu PMD 85-3 v grafickej, dekadickej, hexadecimálnej a binárnej forme:

```
100 FOR I=40 TO 42
110 GOSUB 140
120 NEXT I
130 END
140 I$=CHR$(I)
150 PRINT I$;I;TAB(5);
160 PRINT HEX$(I);" "
170 FOR J=7 TO 0 STEP -1
180 PRINT BIT I,J;
190 NEXT J
200 PRINT
210 RETURN
RUN
( 40 0028  0  0  1  0  1  0  0  0
) 41 0029  0  0  1  0  1  0  0  1
* 42 002A  0  0  1  0  1  0  1  0
```

# CLEAR

Príkaz CLEAR

Opis: Slúži na znovuinicializáciu premenných programu.

Formát: CLEAR

Poznámka: Nuluje všetky numerické premenné a zásobník návratových adries príkazu GOSUB. Ruší ďalej platnosť všetkých predchádzajúcich príkazov DIM a nastaví hodnoty všetkých reťazcových premenných na "".

Príklad:

```
100 DIM A(15),A$(25)
110 B(7)=3
120 A$(21)="TEXT"
130 A(12)=21
140 I=1
150 PRINT I;B(7);A(12);A$(21)
160 CLEAR
170 PRINT I;B(7)
RUN
1 3 21 TEXT
0 0
```

# CODE

Príkaz CODE

Opis: Umožňuje napísať program v strojovom kóde priamo počas písania programu v jazyku BASIC-G.

Formát: CODE a1\$,a2\$,...

Parameter: ai\$ reťazcová konštanta vytvorená z hexadecimálnych číslíc (0..9,A,B,C,D,E,F).

Poznámka: Inštrukcie strojového kódu mikroprocesora MHB8080 (pozri prílohu D) sa zapisujú v tvare reťazca hexadecimálnych číslíc. Takto vytvorený podprogram sa pri interpretácii príkazu CODE uloží do operačnej pamäte od adresy BE00H a vykoná sa. Ak je podprogramov viac (t.j. ďalšie parametre oddelené čiarkami), vykonajú sa postupne.

Príklad:

Vytvorenie podprogramu v strojovom kóde pre výpis znaku A (pozri kap.5.4 podprogram DCHAR1) a jeho trojnásobné vykonanie (symbolický zápis podprogramu v strojovom kóde je vpravo):

```
100 A$="3E41CD00E5C9"           MVI  A,41H
110 CODE A$,A$,A$               CALL 0E500H
RUN                             RET
AAA
```

# CONT

Organizačný príkaz CONT

**Opis:** Pokračuje sa vo vykonávaní prerušeného programu od miesta prerušenia, vyvolaného príkazom STOP.

**Formát:** CONT

**Poznámka:** Po príkaze STOP nie je možné program meniť. Keby používateľ program predsa len zmenil, príkazom CONT by sa už nedalo pokračovať vo vykonávaní.

**Príklad:**

Program, ktorý sa preruší príkazom STOP (po ktorom sa používatel môže pýtať na stav premenných, prípadne meniť ich obsah) a znovu pokračuje po prerušení:

```
100 A=5 : B=7 : S=2
110 PRINT A;B
120 STOP
130 PRINT A;B
140 FOR I=1 TO 5 STEP S
150 PRINT I
160 STOP
170 PRINT I
180 NEXT I
190 GOTO 110
RUN
5 7
+ + + Stop at line 120 + + +
A=3:PRINT S:CONT
2
3 7
1
+ + + Stop at line 160 + + +
```

# CONTROL

## Príkaz CONTROL

Opis: Slúži na zápis hodnôt do v/v registra vo zvolenom kanále.

Formát: CONTROL k,r;e1,e2,...

Parametre: k je číslo v/v kanálu (0..7).  
r je adresa v/v registra vo zvolenom kanále (pozri tabuľku B.3).  
ei je numerická konštanta alebo aritmetický výraz z intervalu (0..255), ktorého hodnota sa zapíše do v/v registra.

Poznámka: Príkaz slúži na jednoduchý zápis jedného alebo viacerých hodnôt (bajtov) do zvoleného v/v registra.  
Podrobnejšie o práci s v/v kanálmi pozri prílohy A a B.

## Príklad:

Nulovanie kanálu 1 (interfejs IRPS) a jeho následná inicializácia na činnosť v asynchrónnom, duplexnom režime so 7-bitovým kódovým rámcom, jedným STOP-bitom, bez parity:

```
100 CONTROL 1,1; 64
```

```
110 CONTROL 1,1; 73,37
```

# COS

## Funkcia COS

Opis: Vracia kosínus uhla (zadaného v radiánoch alebo po príkaze DEG v stupňoch).

Formát: COS(e)

Parameter: e je numerická konštanta, numerická premenná alebo aritmetický výraz.

### Príklad:

Výpis cos 30° a cos 45° :

```
100 RD=57.29577951308
110 DG=0.01745329252
120 R30=30*DG
130 PRINT R30,COS(R30),SQR(3)/2
140 DEG
150 PRINT 30,COS(30),SQR(3)/2
160 RAD
170 R45=45*DG
180 PRINT R45,COS(R45),SQR(2)/2
190 DEG
200 D45=R45*RD
210 PRINT D45,COS(D45),SQR(2)/2
RUN
.523599          .866025          .866026
30               .866025          .866026
.785398          .707107          .707107
45               .707107          .707107
```

# DATA

Príkaz DATA

Opis: Definuje údaje pre príkaz READ.

Formát: DATA z

Parameter: z je zoznam jedného alebo viac údajov (údaj je numerická alebo reťazcová konštanta).

Poznámka: Príkaz definuje údaje, ktoré sa neskôr budú priradovať premenným pomocou príkazu READ. Ak reťazcový údaj obsahuje čiarku alebo dvojbodku, musí byť uzavretý v úvodzovkách.

Riadiť čítanie údajov z riadkov DATA je možné pomocou príkazu RESTORE. Štandardne sa číta prvý riadok DATA v programe.

Príklad:

Program na výpočet súčtu čísel:

```
100 DATA SUCET,0," CISEL: "  
110 READ S$,S,C$  
120 PRINT S$+C$;  
130 FOR I=1 TO 5  
140 READ N  
150 IF I>1 AND N<0 THEN PRINT "-";  
160 IF I>1 AND N>=0 THEN PRINT "+";  
170 S=S+N  
180 PRINT ABS(N);  
190 NEXT I  
200 PRINT "=";S:S=0  
210 END  
220 DATA 2,4,6,8,10
```

```
230 DATA 1,3,5,7,9
240 DATA 3,-2,7,0,-1
RUN
```

```
SUCET CISEL:  2 + 4 + 6 + 8 + 10 = 30
```

```
RESTORE 240
```

```
GOTO 120
```

```
SUCET CISEL:  3 - 2 + 7 + 0 - 1 = 7
```

```
GOTO 120
```

```
SUCET CISEL:
```

```
  + + + Data exhaust at line 140 + + +
```

```
RESTORE 230
```

```
GOTO 120
```

```
SUCET CISEL:  1 + 3 + 5 + 7 + 9 = 25
```



# DEF FNC

Príkaz DEF FNC

Opis: Definuje funkciu s menom FNCm. Funkcia môže, ale nemusí mať parameter.

Formát: DEF FNC a\$(par) = e

Parametre: a\$ je meno funkcie (prvé tri znaky mena sú vždy FNC).  
e je výraz, ktorý funkcia vypočítava.

Poznámka: Tento príkaz nie je možné použiť v priamom móde, iba v programovom.

Príklad:

Program na definovanie hyperbolických funkcií a výpisy ich hodnôt:

```
100 REM DEFINOVANIE HYPERBOLICKYCH FUNKCII
110 DEF FNC M(X)=EXP(X)-EXP(-X)
120 DEF FNC P(X)=EXP(X)+EXP(-X)
130 DEF FNC S(H)=FNC M(H)/2 : DEF FNC C(H)=FNC P(H)/2
150 DEF FNC T(H)=FNC M(H)/FNC P(H)
160 FOR I=-0.4 TO 0.3 STEP 0.1
165 IF ABS(I)<0.1 THEN I=0
170 PRINT I;FNC S(I),FNC C(I),FNC T(I)
180 NEXT I
RUN
-.4 -.410752 1.08107 -.379949
-.3 -.30452 1.04534 -.291313
-.2 -.201336 1.02007 -.197375
0 0 1 0
.1 .100167 1.005 .099668
.2 .201336 1.02007 .197375
.3 .30452 1.04534 .291313
```

# DEG

Príkaz DEG

Opis: Prepína režim výpočtu goniometrických funkcií (SIN, COS, TAN), ktorých parametre sa po tomto príkaze budú chápať v stupňoch.

Formát: DEG

Poznámka: Po príkaze DEG sa parametre goniometrických funkcií chápu v stupňoch. Príkaz DEG je inverzný k príkazu RAD. Po zapnutí mikropočítača platí režim RAD.

Príklad:

Výpis `tg 30° uhla`, zadaného najprv v radiánoch a potom v stupňoch:

```
100 PRINT 30*0.01745329,TAN(30*0.01745329),SQR(3)/3
```

```
110 DEG
```

```
120 PRINT 30,TAN(30),SQR(3)/3
```

```
RUN
```

.523599	.57735	.57735
---------	--------	--------

30	.57735	.57735
----	--------	--------

# DIM

Príkaz DIM

Opis: Deklaruje veľkosť numerických alebo reťazcových polí.

Formát: DIM z

Parameter: z je zoznam deklarovaných polí. Prvok zoznamu má formát v(zi), kde v je meno jedno- až trojrozmerného poľa a zi je zoznam jedného až troch najvyšších indexov tohto poľa.

Poznámka: V príkaze DIM sa môže deklarovať viac polí, ktoré majú rozmer nanajvýš 3. Prvky polí sú číslované od 0. Ak je najvyšší index menší než 10, nie je potrebné pole deklarovať.

Príklad:

Deklarácia trojrozmerného poľa A1, potom deklarácia dvojrozmerného poľa A2 a B2, ktorých hranice rozmerov sa predtým načítajú a deklarácia znakového poľa a\$:

```
100 DIM A1(10,20,30)
110 INPUT N
120 DIM A2(N,N),B2(N,N)
130 DIM A$(10)
```

# DISP

Príkaz DISP

Opis: Výpis zoznamu parametrov do dialógového riadka.

Formát: DISP z

Parameter: z je zoznam výrazov, ktoré sa vypíšu do DR (pozri príkaz PRINT).

Príklad:

Fragment programu, v ktorom sa načítavajú vstupné hodnoty: názov, cena položky a počet kusov:

```
100 DISP "ZADAJ POCET POLOZIEK:"
110 INPUT N
120 DIM NZ$(N),CP(N,2)
130 FOR I=1 TO N
140 DISP "POLOZKA";I;": ZADAJ NAZOV,CENU,POCET"
150 INPUT NZ$(I),CP(I,1),CP(I,2)
160 NEXT I
170 FOR I=1 TO N
180 NZ$(I)=LEFT$(NZ$(I),20)
190 PRINT I;NZ$(I);TAB(27),CP(I,1),CP(I,2)
200 NEXT I
```

# DLOAD

Príkaz DLOAD

Opis: Načíta hodnoty všetkých prvkov poľa zo zadaného súboru na magnetofónovej kazete.

Formát: DLOAD f;v

Parametre: f je číslo súboru na magnetofónovej kazete (max. hodnota 99).  
v je prvý prvok načítavaného poľa.

Poznámka: Hodnoty načítavaného poľa boli predtým uložené na kazetu príkazom DSAVE. Pri načítavaní hodnôt sa kontrolujú rozmery poľa na kazete s rozmermi nahrávaného poľa. Použitie spolu s príkazom DSAVE je vhodné napr. pri triedení poľa.  
Pre f=0 sa načíta prvý záznam údajového typu z kazety do poľa v.

Príklad:

Do poľa A s rozmermi 2,6 načítame hodnoty z magnetofónovej kazety. Tieto hodnoty boli uložené na kazetu ako súbor 4 v príklade príkazu DSAVE:

```
100 DIM A(2,6)
110 ? "Priprav magnetofón a stlač kláves!"
120 DLOAD 4;A(0,0)
130 FOR I=1 TO 2
140 FOR J=1 TO 6
150 PRINT A(I,J);
160 NEXT J: PRINT: NEXT I
RUN
12  13  14  15  16  17
13  14  15  16  17  18
```

# DSAVE

## Príkaz DSAVE

**Opis:** Ukladá všetky hodnoty zadaného poľa na magnetofónovú kazetu do súboru so zadaným číslom.

**Formát:** DSAVE f;v"text"

**Parametre:** f je číslo súboru, pod ktorým sa hodnoty poľa uložia na magnetofónovú kazetu (z intervalu 1..99).

v je prvý prvok ukladaného poľa.

"text" max. 8 znakov, slúžiacich na označenie záznamu na kazete menom.

**Poznámka:** Súbor môže, ale nemusí byť označený na kazete zadaným menom.

## Príklad:

Vytvorenie poľa A s rozmermi 2,6 a jeho zápis na magnetofónovú kazetu do súboru 4. Súbor bude na kazete označený menom "DATA".

```
10 DIM A(2,6)
20 FOR I=1 TO 2
30 FOR J=1 TO 6
40 A(I,J)=I+J+10
50 NEXT J
60 NEXT I
RUN
DSAVE 4;A(0,0)"DATA"
```

# END

Príkaz END

Opis: Ukončuje výkon programu.

Formát: END

Príklad:

Program na výpočet plochy kruhu:

```
100 PRINT "PLOCHA KRUHU S POLOMEROM: ";
110 PI = 3.1415927
120 DISP "ZADAJ POLOMER KRUHU"
130 INPUT R
140 P = PI*R*R
150 PRINT R,P
160 END

200 PRINT "OBVOD KRUHU S POLOMEROM: ";
210 PI=3.1415927
220 DISP "ZADAJ POLOMER KRUHU"
230 INPUT R
240 O=2*PI*R
250 PRINT R,O
260 STOP
270 GOTO 100

RUN
PLOCHA KRUHU S POLOMEROM:    2          12.5664
OK
RUN 200
OBVOD KRUHU S POLOMEROM:    2          12.5664
+ + + Stop at line 260 + + +
CONT
PLOCHA KRUHU S POLOMEROM:    1          3.14159
```

# ENTER

## Príkaz ENTER

Opis: Slúži na vstup údajov zo zadaného v/v kanálu a registra do premenných podľa zadaného zoznamu.

Formát: ENTER kr;v1,v2,...

Parametre: k je číslo v/v kanálu (0..7).  
r je číslo, ktoré pre konkrétny kanál udáva, z ktorého v/v registra a akým spôsobom budú údaje čítané (pozri tabuľku B.4).  
vi je premenná, ktorej sa priradí načítaná hodnota.

Poznámka: Podrobnejšie o práci s v/v kanálmi pozri prílohy A a B.

## Príklad:

Vykoná sa inicializácia kanálu 4 tak, aby brány A a B boli vstupné v režime 0, brána C výstupná a následný vstup údajov do premennej X\$ (z brány A) a jeho zobrazenie:

```
100 CONTROL 4,3; 146
```

```
110 ENTER 400; X$
```

```
120 PRINT X$
```



# EXP

Funkcia EXP

Opis: Vracia hodnotu prirodzeného čísla (2,71) umocneného na hodnotu parametra.

Formát: EXP(e)

Parameter: e je numerická konštanta, numerická premenná alebo aritmetický výraz.

Príklad:

Výpis hodnoty prirodzeného čísla:

```
100 PRINT EXP(1)
```

```
RUN
```

```
2.71828
```

# FILL

## Grafický príkaz FILL

Opis: Vykreslí od aktuálneho bodu zadaný počet bodov v smere osí x,y podľa zadanej masky.

Formát: FILL m,n;e

Parametre: m je výraz pre počet bodov v smere osi x (hodnota z intervalu 0..255).  
n je výraz pre počet bodov v smere osi y (hodnota z intervalu 0..242).  
e je výraz, reprezentujúci bitovú masku, podľa ktorej sa vykresľuje (hodnota z intervalu 0..63).

Poznámka: Príkazom sa vykreslí obdĺžnik so zvolenými rozmermi. Obdĺžnik sa vyplní podľa masky, ktorej hodnota sa rozloží na 8-ciferné binárne číslo. Vykreslia sa body, ktorým zodpovedá binárna 1 v maske. Vykresľovanie prebieha zdola nahor, takže po splnení príkazu zostáva kurzor SUOG na mieste posledného vykresleného bodu.

### Príklad:

Vykreslenie zvislej čiary s hrúbkou 1 bod, po posune kurzora vykreslenie 3 zvislých čiar s hrúbkou 2 body, po posune kurzora vykreslenie vodorovnej čiary, posun kurzora do ľavého dolného miesta a vykreslenie celého priestoru:

```
100 GCLEAR
110 SCALE 0,255,0,242
120 MOVE 20,0
130 FILL 1,242;1
140 MOVE 100,0
150 FILL 2,242;21
160 MOVE 120,120
170 FILL 255,1;1
180 MOVE 0,0
190 FILL 255,242;1
```

Príklad:

Ilustrácia použitia bitovej masky:

```
100 GCLEAR
110 SCALE 0,100,0,100
120 FOR I=0 TO 31
130 MOVE 10,I*3: LABEL 1,1;I
140 MOVE 20,I*3: FILL 9,6;I
150 NEXT I
```

# FOR

Príkaz FOR

Opis: Vykonáva v cykle skupinu príkazov medzi FOR a NEXT.

Formát: FOR v=e1 TO e2 STEP e3

Parametre: v je riadiaca premenná cyklu,  
e1 je výraz (začiatočná hodnota),  
e2 je výraz (koncová hodnota). Ak premenná v nadobudne túto hodnotu (alebo väčšiu), cyklus sa končí.  
e3 výraz (krok cyklu); ak je 1, netreba uvádzať. Po vykonaní skupiny príkazov medzi FOR a NEXT sa hodnota premennej zvýši o hodnotu e3 a cyklus pokračuje.

Príklad:

Výpočet a výpis súčtu čísiel:

```
100 REM SUCET CISEL ARITMETICKEJ POSTUPNOSTI
110 PRINT "ZADAJ A1,AN,D" : INPUT A1,AN,D : N=0 : S=0 : SN=0
140 FOR X=A1 TO AN STEP D : N=N+1 : S=S+X : NEXT X
180 AP=A1+(N-1)*D
190 IF AN<AP THEN PRINT "CHYBNE AN":AN=AP
200 SN=N*(A1+AN)/2
210 PRINT "PRVY CLEN: ",A1;TAB(25);"POSLEDNY CLEN:",AN
230 PRINT "DIFERENCIA: ",D;TAB(25);"POCET CLENOV: ",N
250 PRINT "SUCET CLENOV: ",S;TAB(25);"KONTROLA: ",SN
RUN
ZADAJ A1,AN,D
4,8,2
PRVY CLEN:      4          POSLEDNY CLEN:      8
DIFERENCIA:     2          POCET CLENOV:       3
SUCET CLENOV:  18          KONTROLA:          18
```

# FRE

Funkcia FRE

Opis: Funkcia vracia kapacitu voľnej pamäte v bajtoch na uloženie programu alebo údajov, resp. reťazcov.

Formát: FRE(v) alebo FRE(a\$)

Parametre: v je numerická premenná alebo numerická konštanta.  
A\$ je reťazcová premenná alebo reťazcová konštanta.

Príklad:

Výpis voľného miesta na uloženie programu a reťazcov:

```
100 PRINT FRE(0)
110 PRINT FRE("")
RUN
31191
4095
```

# GCLEAR

Príkaz GCLEAR

Opis: Vymaže celú obrazovku

Formát: GCLEAR

Príklad:

100 GCLEAR

# GCLEARA

Príkaz GCLEARA

Opis: Vymazáva obsah okna SYOG obrazovky.

Formát: GCLEARA

Poznámka: Okno SYOG sa vytvorí príkazom WINDOWA

Príklad:

100 GCLEARA

# GCLEARG

Príkaz GCLEARG

Opis: Vymazáva okno SUOG obrazovky.

Formát: GCLEARG

Poznámka: Okno SUOG sa nastavuje príkazom WINDOWG.

Príklad:

100 GCLEARG



# GOSUB

Príkaz GOSUB

Opis: Vykoná basicovský podprogram, začínajúci sa na zadanom riadku.

Formát: GOSUB e

Parametre: e je celé číslo alebo aritmetický výraz, ktorý sa nezačína číslícou (hodnota z intervalu 0..32769).

Poznámka: Parameter e sa interpretuje ako číslo riadka, na ktorom sa podprogram začína. Ak je to výraz, je povolená forma napr.  $X*10$ , alebo  $(10*X)$ , ale nie je povolené  $10*X$ .

Príklad:

Súčasťou programu sú podprogramy, ktoré vypíšu na obrazovku text. Aby sme si nemuseli pamätať čísla riadkov, kde sa tieto podprogramy začínajú (1000, resp. 2000), na začiatok programu napíšeme priradenia (pozri riadok 100) a potom v príkaze GOSUB uvedieme už len príslušné názvy:

```
100 VPRED = 1000: VZAD = 2000
200 GOSUB VPRED
300 PAUSE 10
400 GOSUB VZAD
500 PAUSE 10
600 GOSUB 32769
700 N=200:PAUSE 10
800 GOSUB (10*N)
900 PAUSE 10
999 END
1000 REM PODPROGRAM
1010 PRINT "VPRED"
1020 RETURN
2000 PRINT "VZAD"
2002 RETURN
32769 PRINT "POSLEDNY RIADOK":RETURN
RUN
VPRED
VZAD
POSLEDNY RIADOK
VZAD
```

# GOTO

Príkaz GOTO

Opis: Program pokračuje príkazom v zadanom riadku.

Formát: GOTO e

Parametre: e je celé číslo alebo aritmetický výraz, ktorý sa nezačína číslicou.

Poznámka: Parameter e sa interpretuje ako číslo riadka. Ak je to výraz, povolená forma je napr.  $X*10$  alebo  $(10*X)$ , ale nie  $10*X$ .

Príklad:

Opakovanie akustického signálu príkazom skoku približne po 1 sec. Program je možné prerušiť stlačením klávesu STOP.

```
100 BEEP 800,4
```

```
110 PAUSE 7
```

```
120 GOTO 100
```

```
RUN
```

```
+ + + Stop at line XXX + + +
```

# HEX\$

Funkcia HEX\$

Opis: Prevedie celú časť parametra do reťazca, ktorý udáva hodnotu parametra v šestnástkovej sústave.

Formát: HEX\$(i)

Parameter: i je numerický výraz alebo konštanta v intervale -32768..32767.

Poznámka: Funkcia vracia vždy reťazec dĺžky 4, podľa potreby doplnený nulami zľava. Ak je i mimo daného intervalu, ohlásí sa chyba.

Príklad:

Výpis hodnôt 4095 a 32767 prevedených na šestnástkové.

```
100 PRINT 4095,HEX$(4095)
110 PRINT 32767,HEX$(32767)
RUN
4095          0FFF
32767         7FFF
```

# IF

## Príkaz IF

**Opis:** Podmienkové vetvenie programu. Ak je podmienka splnená, vykoná sa buď skok na riadok, uvedený za THEN, alebo príkaz za THEN. Ak nie je splnená, vykonáva sa nasledujúci riadok.

**Formát:** IF c THEN n alebo IF c THEN s alebo IF c GOTO n

**Parametre:** c je testovaný výraz, ktorého hodnota je buď TRUE alebo FALSE.  
n je číslo riadka, od ktorého pokračuje vykonávanie programu, ak je podmienka splnená.  
s je príkaz, ktorý sa vykoná, ak je podmienka splnená.

## Príklad:

Program, v ktorom sa do premennej K ukladá počet kladných načítaných čísel a do premennej Z počet načítaných záporných čísel:

```
100 DISP "ZADAJ CISLO! 0 - UKONCUJE VSTUP"
110 INPUT X
120 IF X=0 THEN 150
130 IF X>0 THEN K=K+1 : PRINT X,"KLADNE" : GOTO 100
140 IF X<0 THEN Z=Z+1 : PRINT X,"ZAPORNE" : GOTO 100
150 PRINT
160 PRINT "POCET KLADNYCH CISEL:";K
170 PRINT "POCET ZAPORNYCH CISEL:";Z
100 END
RUN
```

# INKEY

Funkcia INKEY

Opis:           Testuje stlačenie programových klávesov K0...K11. Ak bol niektorý z nich stlačený, vracia príslušnú hodnotu 0...11, inak 255.

Formát:        INKEY

Príklad:

Program vypíše na obrazovku číslo stlačeného programového kľúča:

```
110 DISP "STLAC KLUC K0-K11!  STOP ZASTAVI PROGRAM"
120 A=INKEY
130 IF A=255 THEN 120
150 PRINT A
170 GOTO 120
RUN
```

# INP

Funkcia INP

Opis: Vracia hodnotu, načítanú zo zadaného v/v registra.

Formát: INP(p)

Parameter: p je numerická konštanta, numerická premenná alebo aritmetický výraz z intervalu (0..255) reprezentujúci fyzickú adresu v/v registra (pozri tabuľku B.1).

Poznámka: Podrobnejšie o práci s v/v kanálmi pozri prílohy A a B.

Príklad:

Načítanie hodnoty z brány s adresou 4CH (pozri tabuľku B.1) do premennej PA. Na začiatku sa v/v obvod inicializuje tak, že všetky brány sú vstupné v režime 0:

```
50 OUT '4C,'9B
100 AD = '4C
110 PA = INP(AD)
120 PRINT PA
```

# INPUT

## Príkaz INPUT

Opis: Priraduje hodnoty načítané z klávesnice, prípadne zo zadaného kanálu, premenným v udanej postupnosti.

Formát: INPUT v1,v2,... alebo INPUT# k;v1,v2,...

Parametre: vi je numerická alebo reťazcová premenná.  
k je číslo v/v kanálu (pozri tabuľku B.2).

Poznámka: Tento príkaz nie je možné použiť v priamom móde, iba v programovom.

## Príklad:

Program pre prevod kcal na jouly. Hodnota kcal sa zadá z klávesnice:

```
100 DISP "ZADAJ HODNOTU KCAL"
110 INPUT K
120 J = K*4186.8
130 PRINT K; "KCAL JE"; J; "JOULOV"
RUN
ZADAJ HODNOTU KCAL 1
1 KCAL JE 4186.8 JOULOV
```



# INT

Funkcia INT

Opis: Vracia najväčšie celé číslo, ktoré je menšie, alebo rovné parametru funkcie.

Formát: INT(e)

Parameter: e je numerická premenná alebo aritmetický výraz.

Príklad:

Program na výpis INT hodnôt premenných A a B:

```
100 A=2.8
110 B=-2.2
120 PRINT INT(A);INT(B)
RUN
2 -3
```

# LABEL

Grafický príkaz LABEL

Opis: Určuje násobok zväčšenia grafického výpisu textu.

Formát: LABEL m,n;z alebo LABEL \*;z

Parametre: m je násobok zväčšenia v smere osi x.  
n je násobok zväčšenia v smere osi y.  
z je zoznam vypisovaných výrazov.  
\* slúži na zachovanie predchádzajúcich hodnôt m a n.

Poznámka: Príkaz pracuje v okne SUOG podobne ako príkaz PRINT. Znaky sú normálne zobrazené v rastrí 6x8 bodov, z toho samotný znak zaberá 5x7 bodov. Príkaz LABEL 2,2 teda znamená zobrazenie znaku v rastrí 12x16 bodov. Ak nechceme opakovať hodnoty násobkov zadané už predtým, vo formáte príkazu uvedieme hviezdičku.

Príklad:

Zobrazenie reťazca znakov "PMD85" s dvojnásobným rozmerom uprostred obrazovky a hodnoty výrazu  $\cos(A) \cdot 3$ . Dvojnásobný rozmer z predchádzajúceho príkazu LABEL ostane zachovaný:

```
100 GCLEAR
110 SCALE -10,10,-10,10
120 MOVE -4,0
130 LABEL 2,2;"PMD85"
140 PAUSE 5
150 MOVE -2.5,-3
160 LABEL *;COS(A)*3
```

# LEFT\$

Funkcia LEFT\$

Opis: Vracia zo zadaného reťazca zľava zadaný počet znakov.

Formát: LEFT\$(a\$,i)

Parametre: a\$ je reťazcová konštanta alebo reťazcová premenná.  
i je výraz pre počet znakov (i musí byť väčšie, ako 0). Ak i je väčšie než dĺžka reťazca a\$, funkcia vracia celý reťazec a\$.

Príklad:

Program, demonštrujúci použitie funkcie LEFT\$:

```
100 A$ = "PMD85"  
110 B$ = LEFT$(A$,99)  
120 PRINT B$,LEN(B$)  
130 PRINT  
140 FOR I = 1 TO LEN(A$)  
150 B$ = LEFT$(A$,I)  
160 PRINT B$  
170 NEXT I  
RUN  
PMD85          5  
P  
PM  
PMD  
PMD8  
PMD85
```

# LEN

Funkcia LEN

Opis: Vracia dĺžku zadaného reťazca.

Formát: LEN(a\$)

Parameter: a\$ je reťazcová konštanta alebo reťazcová premenná.

Príklad:

Výpis súčtu dĺžok dvoch reťazcov:

```
100 A$ = "PMD-85" : B$ = "PMD-85/3"
```

```
110 PRINT LEN(A$)+LEN(B$)
```

```
RUN
```

```
14
```

# LET

## Príkaz LET

Opis: Priradí premennej na ľavej strane operátora priradenia (znak "=") hodnotu výrazu na pravej strane.

Formát: LET v = e

Parametre: v je aritmetická alebo reťazcová premenná.  
e je výraz, ktorý musí byť po vyhodnotení rovnakého typu ako premenná v.

Poznámka: Kľúčové slovo LET nie je v priradovacom príkaze povinné uvádzať. Ak je na ľavej strane priradovacieho príkazu numerická premenná, na pravej strane môže byť aj logický výraz, Vtedy sa priradí hodnote FALSE 0, hodnote TRUE -1.

## Príklad:

```
100 A=10 : B=1
110 LET C = A>B
120 LET D = B>A
130 PRINT "C=";C,"D=";D
RUN
C=-1          D=0
```

# LIST

Organizačný príkaz LIST

Opis: Vypíše text programu (celý alebo od zvoleného riadka) na obrazovku, prípadne na zadaný kanál.

Formát: LIST r alebo LIST# k;r

Parametre: r riadok programu, od ktorého sa začína výpis.  
k číslo kanálu, cez ktorý sa vypisuje.

Poznámka: Pri vykonávaní príkazu LIST sa riadky, ktoré majú viac než 48 znakov, vypisujú do dvoch riadkov obrazovky. Začiatok druhého riadka je odsunutý od ľavého okraja tak, aby bolo viditeľné číslo riadka.

Príklad:

Vkladanie a výpis programu na výpočet plochy kruhu. Jednotlivé riadky sa vkladajú do dialógového riadka. Po vložení programu sa program vypíše príkazom LIST. Príkazom LIST# 403; sa program vypíše na pripojenej tlačiarňi.

NEW

```
10 PRINT "PLOCHA" : PI=3.14159 : DISP "POLOMER= "
```

```
20 INPUT R : PRINT R, PI*R*R
```

LIST

```
10 PRINT "PLOCHA" : PI*3.14159 : DISP "POLOMER= "
```

```
20 INPUT R : PRINT R, PI*R*R
```

```
LIST #403;
```

# LLIST

Organizačný príkaz LLIST

**Opis:** Vypisuje do dialógového riadka obrazovky program postupne po riadkoch (od začiatku alebo od zadaného riadka programu) na editovanie.

**Formát:** LLIST r

**Parameter:** r je číslo riadka, od ktorého sa má začať výpis. Ak sa neudá, vypíše sa prvý riadok programu a po stlačení klávesu EOL sa vypíše ďalší riadok programu.

**Poznámka:** Riadok programu vypísaný do dialógového riadka obrazovky môže používateľ editovať. Po editovaní riadka sa nasledujúci riadok nevypíše.

**Príklad:**

V riadku programu s číslom 10 pridáme k textu "PLOCHA" text " KRUHU" (pozri príklad pri príkaze LIST).

LLIST 10

Po tomto príkaze sa do dialógového riadka vypíše:

```
10 PRINT "PLOCHA" : PI=3.14159 : DISP "POLOMER= "
```

Presunieme kurzor za text "PLOCHA, šesťkrát stlačíme kláves INS a dopíšeme text KRUHU. V dialógovom riadku bude nakoniec text:

```
10 PRINT "PLOCHA KRUHU" : PI=3.14159 : DISP "POLOKER= "
```

Editovanie riadka ukončíme klávesom EOL.

# LOAD

## Organizačný príkaz LOAD

**Opis:** Z magnetofónovej kazety uloží do operačnej pamäte zadaný súbor s programom (v zdrojovom alebo binárnom tvare). Program z kazety sa dá aj prihrať za existujúci iný program v operačnej pamäti, a to aj s prípadným prečíslovaním riadkov.

**Formát:** LOAD f alebo LOAD CODE f alebo LOAD END f,r

**Parametre:** f je číslo súboru na kazete (z intervalu 0..99). Ak je 0, uloží sa do operačnej pamäte nasledujúci súbor z kazety bez ohľadu na číslo. Pri LOAD CODE ide o binárny súbor (napr. program v strojovom kóde).  
r je číslo prvého riadka v ukladanom súbore. Riadky ukladaného programu sa adekvátne prečísľujú.

**Poznámka:** Pri vykonaní príkazu LOAD END sa program z kazety uloží za program v operačnej pamäti. Musí platiť, že číslo posledného riadka programu v operačnej pamäti je menšie než prvé číslo riadka programu z kazety. Ak treba riadky programu z kazety prečíslovať, stačí zadať ako ďalší parameter, akú hodnotu má mať číslo prvého riadka.

### Príklad:

Za program v operačnej pamäti uložíme z kazety zo súboru 11 ďalší program. Posledný riadok programu v pamäti má číslo 990, takže program z kazety treba prečíslovať od 1000:

```
LOAD END 11,1000
```



# LOG

Funkcia LOG

Opis: Vracia prirodzený logaritmus výrazu, ktorého hodnota je väčšia ako 0.

Formát: LOG(e)

Parameter: e je numerická konštanta, numerická premenná alebo aritmetický výraz.

Príklad:

Výpis hodnôt prirodzeného logaritmu a desiatkového logaritmu:

```
100 REM PRIRODZENY LOGARITMUS
110 PRINT LOG(1E-38)
120 PRINT LOG(EXP(1))
130 PRINT
140 REM DESIATKOVY LOGARITMUS
150 MOD=0.434295
160 FOR I=0 TO 3
170 X=10^I
180 PRINT I,X,LOG(X)*MOD
190 NEXT I
```

RUN

-87.4982

1

0	1	0
---	---	---

1	10	1
---	----	---

2	100	2
---	-----	---

3	1000	3
---	------	---

# MID\$

Funkcia MID\$

Opis: Zo zadaného reťazca vracia od udaného znaku zadaný počet znakov alebo zvyšok do konca reťazca.

Formát: MID\$(a\$,i,j)

Parametre: a\$ je reťazec alebo reťazcová premenná.  
i je poradové číslo znaku, od ktorého sa začína zvyšok reťazca,  
j je počet vrátených znakov.

Poznámka: Ak chceme, aby funkcia vrátila celý zvyšok reťazca, nemusíme zadať počet znakov.

Príklad:

Rozdeľovanie reťazca pomocou funkcie MID\$:

```
100 T$ = "PMD 85-3"  
110 T=LEN(T$)  
120 FOR I=1 TO T-1  
130 PRINT TAB(I);MID$(T$,I,2);TAB(T+4);T$  
140 NEXT I
```

RUN

PM	PMD 85-3
MD	PMD 85-3
D	PMD 85-3
8	PMD 85-3
85	PMD 85-3
5-	PMD 85-3
-3	PMD 85-3

# MOVE

Grafický príkaz MOVE

Opis: Presunie kresliace pero do bodu so zadanými súradnicami.

Formát: MOVE e1,e2

Parametre: e1 je výraz pre x-ovú súradnicu bodu,  
e2 je výraz pre y-ovú súradnicu bodu.

Poznámka: Pero sa premiestni do bodu so zadanými súradnicami podľa mierky zadanej predtým príkazom SCALE (pozri príkaz SCALE).

Príklad:

Vykreslenie priebehu funkcie  $\sin(x)$ :

```
100 GCLEAR
110 PI = 3.1415927
120 SCALE 0,2*PI,-2,2
130 AXES 0,0
140 FOR X=0 TO 2*PI STEP 0.2
150 MOVE X,0
160 PLOT X,SIN(X)
170 NEXT X
```

# NEW

Organizačný príkaz NEW

Opis: Vymazáva z operačnej pamäte program alebo jeho časť a inicializuje uvoľnenú pamäť.

Formát: NEW n1,n2      alebo      NEW RETURN

Parametre: n1      je číslo riadka, od ktorého sa program vymazáva.  
n2      je číslo riadka, po ktorý sa program vymazáva.

Poznámka: Ak sa neudá žiadny parameter, vymaže sa celý program. Ak sa udá iba n1, vymažú sa riadky od n1 do konca. Ak je prvý parameter 0, vymažú sa riadky od začiatku programu po n2. Ak je n1 väčšie než n2 nevymaže sa nič.  
Pokiaľ bol vymazaný celý program alebo jeho koniec, príkaz NEW RETURN ho obnoví.

Príklad:

Z daného programu sa vymažú riadky 130-150:

```
100 PRINT "PLOCHA"  
110 PI = 3.14159  
120 DISP "POLOMER = "  
130 INPUT R  
140 F = PI*R*R  
150 PRINT R,F  
160 END  
NEW 125,155
```

Po vykonaní príkazu NEW ostane v pamäti program:

```
100 PRINT "PLOCHA"  
110 PI = 3.14159  
120 DISP "POLOMER = "  
160 END
```

# NEXT

Príkaz NEXT

Opis: Ukončuje cyklus so zadanou riadiacou premennou, resp. zadanými riadiacimi premennými. Hodnota riadiacej premennej sa zvýši o krok cyklu.

Formát: NEXT z

Parameter: z je zoznam riadiacich premenných, pre ktoré sa ukončuje cyklus, začatý príkazom FOR.

Poznámka: Parameter z sa môže vynechať. V takom prípade sa ukončí cyklus, začatý najbližším predchádzajúcim príkazom FOR.

Príklad:

Výpis matice A s rozmermi 2x3:

```
100 DIM A(2,3): DATA 1,0,0,0,1,0
110 FOR I=1 TO 2: FOR J=1 TO 3: READ A(I,J): NEXT J,I
120 PRINT "    I";TAB(8);1;TAB(16);2;TAB(24);3
130 PRINT "-----"
140 FOR I=1 TO 2 : PRINT I;"I";
150 FOR J=1 TO 3 : PRINT TAB(J*8);A(I,J);
160 NEXT J: PRINT: NEXT I
```

RUN

	I	1	2	3
1 I	1	0	0	
2 I	0	1	0	

# ON

## Príkaz ON

**Opis:** Podľa hodnoty zadaného výrazu sa skáče alebo vykoná podprogram na príslušnom riadku. V prípade ON ERR sa po prvej chybe programu vykonajú príkazy v riadku za ON ERR.

**Formát:** ON e GOSUB z alebo  
ON e GOTO z alebo  
ON ERR s

**Parametre:** z je zoznam čísiel riadkov, na ktoré sa má skočiť, resp. na ktorých sa začína podprogram, ktorý sa má vykonať,  
s je jeden alebo viac príkazov, ktoré sa majú vykonať po chybe programu,  
e je testovaný výraz.

**Poznámka:** Pri príkazoch ON e GOTO, resp. ON e GOSUB sa začne vykonávať ten riadok v postupnosti riadkov, ktorého poradie určuje hodnota parametra e.

Pri príkaze ON ERR platí, že po vykonaní tohto príkazu sa príkazy v riadku ON ERR zatiaľ ignorujú, ale vykonajú sa až po prípadnom výskyte chyby. Ak však dôjde k ďalšej chybe, program sa už zastavuje (ak nie je iný príkaz ON ERR).

Pri vzniku chyby po ON ERR sa dá zistiť riadok, kde došlo k chybe (2 bajty od adresy 'AE5D), napr. `RIADOK=APEEK('AE5D)` a aj kód chyby (1 bajt na adrese 38, napr. `KODCH=PEEK('26)`. (Pozri prílohu G).

Príklad:

Výpis hodnoty výrazu  $1/(X*X-9)$ . Chceme zabrániť zastaveniu programu pri chybe ( $X = 3$  alebo chybe pri zadaní vstupného údaj):

```
100 ON ERR GOTO 500
110 DISP "ZADAJ X"
120 INPUT X
130 PRINT X,1/(X*X-9)
140 GOTO 110
500 DISP "CHYBNY VSTUP, ZADAJ ZNOVA"
510 ON ERR GOTO 500
520 GOTO 120

RUN
ZADAJ X
7
7          .025
ZADAJ X
A
CHYBNY VSTUP, ZADAJ ZNOVA
2
2          -.2
ZADAJ X
3
3          CHYBNY VSTUP, ZADAJ ZNOVA
0
0          -.111111
ZADAJ X
```

# OUT

## Príkaz OUT

Opis:           Zapíše hodnotu výrazu do zadaného v/v registra.

Formát:       OUT p,e

Parametre:   p     je numerická konštanta, numerická premenná alebo aritmetický výraz z intervalu (0..255) reprezentujúci fyzickú adresu v/v registra (pozri tabuľku B.1).

              e     je konštanta, premenná alebo výraz, ktorého hodnota sa má zapísať.

Poznámka:    Podrobnejšie o práci s v/v kanálmi pozri prílohy A a B.

### Príklad:

    Inicializácia časovača TIMER 0 tak, aby na jeho výstupe OUT0 bol signál s frekvenciou 1 kHz (pozri prílohu A.5). Na vstup CLK0 bude privedená frekvencia systémových hodín  $\Phi 2$ , deliaci pomer, ktorý je potrebné zapísať do časovača je 2000, pretože vstupná frekvencia je 2 MHz.

10 OUT 95,39

20 OUT 92,32

Poznámka:    Privedenie systémových hodín na vstup CLK0 je možné urobiť drôtenou prepojkou na konektore K2 (pozri obrázok A.6).



# OUTPUT

Príkaz OUTPUT

Opis: Zapisuje zoznam údajov na zadaný v/v kanál.

Formát: OUTPUT kr;z

Parametre: k je číslo kanálu (0..7).  
r je číslo, ktoré pre konkrétny kanál udáva, do ktorého v/v registra a akým spôsobom budú údaje zapísané (pozri tabuľku B.4).  
z je zoznam výrazov, ktoré sa zapíšu do v/v kanála, ich špecifikácia je rovnaká ako v príkaze PRINT.

Poznámka: Podrobnejšie o práci s v/v kanálmi pozri prílohy A a B.

Príklad:

Ak máme na v/v kanál 4 pripojenú tlačiareň cez interfejs GPIO/1 (pozri tabuľku B.4), môžeme vytlačiť text:

```
100 A=5
110 B=3*5
120 OUTPUT 43;A;SPC(2);"TEXT",B;TAB(30);"LIST"
RUN
```

```
5      TEXT      15      LIST
```

# PAUSE

Príkaz PAUSE

Opis: Zastaví vykonávanie programu na zadaný počet desatín sekundy.

Formát: PAUSE e

Parameter: e je výraz, udávajúci počet desatín sekundy (hodnota z intervalu 1..255). Ak e=0, vykoná sa ako e=255. Ak sa e neudá, príkaz sa vyhodnotí ako PAUSE 0.

Poznámka: Čakanie programu možno ukončiť stlačením klávesu "medzera".

Príklad:

Generovanie akustických signálov s prestávkami, ktorých dĺžka závisí od parametra I:

```
100 FOR I=0 TO 7
110 READ FR(I)
120 NEXT I
130 DATA 246.94,261.63,293.67,329.63,349.23
140 DATA 392.00,440.00,493.89
150 FOR I=1 TO 90 STEP 5
160 DISP " PAUSE ";I:PAUSE I
170 FOR J=0 TO 7
180 BEEP FR(J),8
190 NEXT J
200 NEXT I
```

# PEEK

Funkcia PEEK

Opis: Vracia hodnotu (1 bajt) zo zadanej adresy v operačnej pamäti.

Formát: PEEK(adr)

Parameter: adr je výraz, ktorý sa vyhodnotí ako adresa.

Príklad:

Po chybe programu zistíme funkciou PEEK kód chyby (pozri prílohu G a príkaz ON):

```
100 ON ERR GOTO 500
110 DISP "ZADAJ X"
120 INPUT X
130 PRINT X,1/(X*X-9) : GOTO 110
500 ER = PEEK('26)
510 IF ER=8 THEN DISP "PRETECENIE, ZADAJ ZNOVA" : GOTO 570
520 IF ER=12 THEN DISP "DELENIE NULOOU, ZADAJ ZNOVA"
570 ON ERR GOTO 500
580 GOTO 120
RUN
ZADAJ X
2
2          -.2
ZADAJ X
2.25E+19
2.25E+19    PRETECENIE, ZADAJ ZNOVA
10
10          .010989
ZADAJ X
+ + + Stop at ...
```

# PEN

## Grafický príkaz PEN

Opis: Nastaví spôsob grafického zobrazenia podľa zadaného výrazu.

Formát: PEN e

Parameter: e je výraz, ktorého hodnota je súčtom niektorých nasledujúcich hodnôt:

	VIDEO	RGB	ČB
0 -	biela	zelená	biela
1 -	zelená	červená	\\\\\\\\\\
2 -	červená	modrá	////////
3 -	hnedá	fialová	XXXXXX
4 -	inverzia (kreslenie na farebný podklad).		
8 -	negácia kreslených bodov bez ohľadu na podklad.		
16 -	mazanie svietiacich bodov bez ohľadu na farbu.		

Poznámka: Pri použití hodnoty 0 až 3 sa kreslí na čierny podklad vybranou farbou.

Pri použití hodnoty 4 až 7, t.j. súčtu  $4+(0..3)$  sa kreslí čierna čiara na farebný podklad.

Pri použití hodnoty 8 až 11, t.j. súčtu  $8+(0..3)$  sa negujú kreslené body bez ohľadu na farbu. (Svietiace body zhasne a zhasnuté sa rozsvietia.)

Pri použití hodnoty 16 až 19 sa kreslí čierna čiara bez ohľadu na podklad.

Pri zapnutí počítača platí implicitne PEN 8.

Príklad:

Vykreslenie vzorkovníka farieb na obrazovke:

```
100 GCLEAR
110 SCALE 0,255,0,242
120 FOR I=0 TO 16
130 PEN I:PRINT AT5+I,3;"PEN";I
140 MOVE 60,190-I*9:PEN I:FILL 60,8;1
150 NEXT I
```

# PLOT

## Grafický príkaz PLOT

**Opis:** Vykreslí úsečku od polohy kurzora do zadaného bodu (prípadne pokračuje ďalšími úsečkami). Vykresľovať sa môžu aj samostatné body.

**Formát:** PLOT z

**Parameter:** z je zoznam skupín údajov o zadanom bode alebo zadaných bodoch. Skupiny údajov o bodoch sú oddelené bodkočiarkami. Jedna skupina je zoznamom troch výrazov m,n,e, kde:

m je x-ová súradnica bodu,

n je y-ová súradnica bodu,

e ak je rôzne od 0, znamená, že sa vykreslí len zadaný bod (nie celá úsečka). Ak sa nezadá, platí 0.

**Poznámka:** Ak zadáme viac bodov, vykresľovanie pokračuje úsečkou k nasledujúcemu bodu atď., takže jedným príkazom PLOT môžeme nakresliť lomenú čiaru. Kurzor ostáva na mieste posledného uvedeného bodu.

### Príklad:

Vykreslenie obdĺžnika s bodkami v rohoch:

```
100 GCLEAR
110 SCALE 0,2.3,0.65,2.6
120 MOVE 1,1
130 PLOT 1,2; 2,2; 2,1; 1,1
140 MOVE 1.1,1.1
150 PLOT 1.1,1.9,1;1.9,1.9,1;1.9,1.1,1;1.1,1.1,1
160 END
```

# POKE

Príkaz POKE

Opis: Od zadanej adresy zapíše do operačnej pamäte zadané 1-bajtové hodnoty.

Formát: POKE adr,z

Parametre: adr je výraz, vyhodnotený ako adresa.  
z je zoznam výrazov, ktorých hodnota sa zapisuje do pamäte.

Príklad:

Od adresy '7000H zapíšeme hodnoty 1,2,3,4:

```
100 POKE '7000,1,2,3,4
```

# POS

Funkcia POS

Opis: Vracia aktuálnu pozíciu (aktuálny stípec) kurzora SYOG.

Formát: POS(x)

Parametre: x formálny parameter.

Príklad:

Program napíše na vymazanú obrazovku podčiarknutý nadpis, ktorý používateľ zadal:

```
200 DISP "Zadaj názov:"
210 INPUT N$
220 GCLEAR
230 PRINT AT 5,10;N$;
240 K=POS(Y)
250 PRINT AT 6,10;
260 FOR I=10 TO K-1
270 PRINT "-";
280 NEXT
RUN
```

Zadaj názov:

Oponentský posudok

Oponentský posudok

-----



# PRINT

## Príkaz PRINT

**Opis:** Vypíše na obrazovku alebo na zadaný kanál hodnoty podľa zoznamu. Pri výpise na obrazovku sa môže špecifikovať miesto výpisu, farba vykreslenia, posuny kurzora a vkladanie medzier.

**Formát:** PRINT AT r,s;z alebo PRINT# k;z

**Parametre:**

- r je riadok, do ktorého sa má vypisovať (z intervalu 0..25).
- s je stĺpec, od ktorého sa má vypisovať (z intervalu 0..47).
- k je číslo kanálu, na ktorý sa vypisuje.
- z je zoznam výrazov, ktorých hodnoty sa vypisujú. Prvkami zoznamu môžu byť numerické alebo reťazcové výrazy a špecifikátory TAB, SPC a INK. Prvky sa oddeľujú čiarkou (stĺpcová tabelácia), alebo bodkočiarkou (výpis bez medzier). Ak za posledným prvkom zoznamu nie je oddeľovač, ďalší PRINT sa začína od začiatku nasledujúceho riadka.

**Špecifikátory:** TAB(e) posunie kurzor na e-tý stĺpec (hodnota z intervalu 0..47)

SPC(e) vloží e medzier

INK(e) mení vykresľovanie do okna SYOG:

	VIDEO	RGB	ČB
e = 0	biela	zelená	biela
1	zelená	červená	\\\\\\\\\\
2	červená	modrá	/////
3	hnedá	fialová	XXXXXX
4	inverzia		

Parameter e je výraz. Pri INK je jeho hodnota súčtom niektorých z uvedených hodnôt. Implicitne platí INK(0).

**Príklad:**

Do 11. riadka a od 6. stĺpca sa vypíše podľa typu zobrazovacej jednotky príslušnou farbou slovo "BASIC", inverzne text "V3.0" a príslušnou farbou text "PMD 85-3":

```
100 GCLEAR
110 FOR I=1 TO 26
120 PRINT I
130 NEXT I
140 PRINT AT 10,5; INK(3); "BASIC"; INK(7); TAB(12); "V3.0"; INK(0);
150 PRINT SPC(2); "PMD 85-3"
```

**Príklad:**

Vykreslenie približných uhlopriečok z hviezdíček:

```
100 GCLEAR
110 FOR I=0 TO 23
120 PRINT AT I, (2*I)-(0.3*I); "*"
130 PRINT AT I, 46-(2*I)+(0.3*I); "*"
140 NEXT I
```

# RAD

Príkaz RAD

Opis: Predefinuje jednotky parametrov funkcií SIN, COS, TAN, ATN na radiány.

Formát: RAD

Poznámka: Po odštartovaní programu sa jednotky parametrov trigonometrických funkcií implicitne chápu v radiánoch až do zmeny na stupne.

Príklad:

Výpis hodnoty  $\sin(x)$ . Parameter je najprv v radiánoch, potom v stupňoch a nakoniec opäť v radiánoch:

```
100 DG=0.01745329252
110 PRINT 45*DG;SIN(45*DG)
120 DEG
130 PRINT SIN(45);SQR(2)/2
140 RAD
150 PRINT SIN(60*DG);SQR(3)/2
RUN
.785398 .707107
.707107 .707107
.866025 .866026
```

# READ

## Príkaz READ

**Opis:** Postupne priradí hodnoty zadaným premenným podľa poradia údajov v príkazoch DATA.

**Formát:** READ z

**Parameter:** z je zoznam premenných.

**Poznámka:** Každý ďalší príkaz READ pokračuje v priradovaní ďalšou hodnotou v príkaze DATA. Ak sú vyčerpané v programe už všetky príkazy DATA, ohlásí sa chyba v programe. Príkaz úzko súvisí s príkazom RESTORE.

## Príklad:

Súčet čísel, zadaných v príkaze DATA:

```
100 DATA SUCET,0," CISEL: "
```

```
110 READ S$,S,C$
```

```
120 PRINT S$+C$;
```

```
130 FOR I=1 TO 5
```

```
140 READ N
```

```
150 S=S+N
```

```
160 NEXT I
```

```
170 PRINT "=";S:S=0
```

```
180 END
```

```
190 DATA 1,3,5,7,9
```

```
RUN
```

```
SUCET CISEL: =25
```

# REM

Príkaz REM

Opis: Text uvedený za príkazom REM slúži ako poznámka.

Formát: REM text

Parameter: text je sled ľubovoľných znakov.

Poznámka: Dvojbodka v texte sa ignoruje, aj prípadné nasledujúce príkazy v riadku s REM.

Príklad:

Príkaz REM obvykle slúži na pomenovanie a dokumentovanie programov a ich dôležitých častí. Napríklad program na prevod z kcal na jouly (príkaz PRINT v riadku 100 sa ignoruje):

```
100 REM KCAL/JOUL: PRINT "P R E V O D   K C A L   -   J O U L"
110 DISP "ZADAJ HODNOTU KCAL   0 - KONIEC"
120 INPUT K: REM NACITANA HODNOTA KCAL Z KLAVESNICE
130 J = 4186.8*K
140 IF K=0 THEN END
150 PRINT K; "KCAL JE"; J; "JOULOV"
160 GOTO 110

RUN

ZADAJ HODNOTU KCAL   0 - KONIEC
1
  1 KCAL JE 4186.8 JOULOV
ZADAJ HODNOTU KCAL   0 - KONIEC
2
  2 KCAL JE 8373.6 JOULOV
ZADAJ HODNOTU KCAL   0 - KONIEC
0
OK
```

# REN

Organizačný príkaz REN

Opis: Prečísluje zadaný úsek programu.

Formát: REN r1,r2,n,m

Parametre: r1 je riadok, od ktorého sa má prečíslovať úsek programu. Ak sa neuvedie, prečíslováva sa od prvého riadka programu.

r2 je riadok, po ktorý sa má prečíslovávať. Ak sa neuvedie, prečísluje sa po posledný riadok programu.

n je nové číslo prvého riadka daného úseku programu. Ak sa neuvedie, ostáva mu pôvodné číslo (=r1), ak ani to nie je uvedené, platí 10.

m je krok číslovania. Ak sa neuvedie, platí 10.

Poznámka: Takýmto spôsobom prečíslovaný úsek programu sa nesmie prekrývať s iným úsekom, inak dôjde k ohláseniu chyby.

Vo všetkých príkazoch IF, GOTO, ON ERR, GOSUB sa príslušné čísla riadkov tiež prečíslujú. Ak sú čísla riadkov definované ako premenné (pozri príkaz GOSUB), neprečíslujú sa.

Príklad:

Prečíslovanie riadkov 120-150 v programe (pozri príkaz IF). Nové riadky majú začiatočnú hodnotu 152 s krokom 2:

```
REN 120,150,152,2
```

```
LIST
```

# RESTORE

Príkaz RESTORE

Opis: Nastaví čítanie údajov (pomocou READ) na prvý príkaz DATA v programe alebo na riadok so zadaným číslom, v ktorom je príkaz DATA.

Formát: RESTORE e

Parameter: e je výraz pre číslo riadka, obsahujúci príkaz DATA, z ktorého sa bude čítať nasledujúcim príkazom READ.

Poznámka: Pre potreby čítania pomocou príkazu READ si program udržiava vnútorný ukazovateľ údajov (ukazuje na nasledujúci údaj z DATA, ktorý sa bude čítať príkazom READ). Po každom čítaní sa tento ukazovateľ aktualizuje. Príkaz RESTORE (bez parametrov) slúži na nastavenie tohto ukazovateľa na prvý údaj v prvom príkaze DATA v programe alebo (s parametrom) na prvý údaj v príkaze DATA so zadaným číslom riadka.

Príklad:

Výpočet hodnôt SIN a COS uhlov 0,90,180,270 a 360:

```
100 READ T$
110 DATA SIN(X)
120 DATA "COS(I)"
130 DATA 0,90,180,270,360
140 DEG
150 RESTORE 130
160 GOSUB 200
170 RESTORE 120: READ T$
180 GOSUB 200
190 END
200 PRINT " X",T$
210 READ X
220 PRINT X,VAL(T$)
230 IF X=360 THEN RETURN
240 GOTO 210
```

RUN

X	SIN(X)
0	0
90	1
180	0
270	-1
360	0
X	COS(X)
0	1
90	0
180	-1
270	0
360	1

Použitie príkazu RESTORE v príkazovom režime ilustruje príklad pri príkaze DATA.



# RETURN

Príkaz RETURN

Opis:           Návrat z podprogramu do hlavného programu, kde sa pokračuje vykonávaním príkazu za tým príkazom GOSUB, ktorý skok do podprogramu vyvolal.

Formát:        RETURN

Príklad:

Skoky do podprogramu a návraty späť:

```
100 DISP "ZADAJ CISLO PODPROGRAMU (1-3) "  
110 INPUT A : IF A>3 OR A<1 THEN 100  
120 GOSUB A*200  
130 END  
200 PRINT " * * * "  
210 RETURN  
400 PRINT " + + + "  
410 RETURN  
600 PRINT " - - - "  
620 RETURN  
RUN  
ZADAJ CISLO PODPROGRAMU (1-3)  
2  
+ + +
```

# RIGHT\$

Funkcia RIGHT\$

Opis: Vracia z daného reťazca zadaný počet posledných znakov.

Formát: RIGHT\$(a\$,i)

Parametre: a\$ je reťazcová konštanta alebo reťazcová premenná.  
i je počet znakov v a\$ sprava (i väčšie alebo rovné 1).

Poznámka: Pre i väčšie než počet znakov reťazca funkcia, vracia celý reťazec.

Príklad:

Výpisy pravých častí reťazca "PMD 85-3" a ich dĺžok:

```
100 A$="PMD 85-3"
110 A=LEN(A$)
120 FOR I=A TO 1 STEP -1
130 J=J+1
140 B$=RIGHT$(A$,J)
150 PRINT TAB(I-1); B$,LEN(B$)
160 NEXT I : END
```

RUN

3	1
-3	2
5-3	3
85-3	4
85-3	5
D 85-3	6
MD 85-3	7
PMD 85-3	8

# RND

Funkcia RND

Opis: Vracia pseudonáhodné číslo z intervalu (0,1), prípadne predchádzajúce pseudonáhodné číslo alebo hodnotu prvého člena radu pseudonáhodných čísel.

Formát: RND(i)

Parameter: i>0 generuje sa nové pseudonáhodné číslo.  
i=0 vracia sa predchádzajúce pseudonáhodné číslo.  
i<0 vracia sa hodnota prvého člena radu pseudonáhodných čísel.

Príklad:

A sa mení od 0 po 100 opakovane, B sa náhodne generuje od 0 po 1. Program zisťuje, či A=22 a súčasne je B v rozsahu od 0.3 po 0.5:

```
100 B = RND(1)
110 IF A=22 THEN GOSUB 160
120 A = A+1
130 IF A = 100 THEN LET A=0
140 GOTO 100
150 END
160 IF (B>.3 AND B<.5) THEN PRINT A;B: BEEP 1045.51,8
170 RETURN
RUN
22 .361614
22 .462626
22 .33546
22 .462338
22 .306027
```

# ROM

## Príkaz ROM

Opis: Načíta oblasť o veľkosti 1 KB z ROM-modulu do operačnej pamäte od adresy B000H a na túto adresu odovzdá riadenie.

Formát: ROM i

Parameter: i je poradové číslo 1 KB-ovej oblasti v ROM-module z rozsahu od 0 po 21, podľa nasledujúcej tabuľky:

adresa v ROM-module	i	poznámka
0H - 27FFH		BASIC-G/V3.0
2800H - 2BFFH	0	Pamäte typu ROM s programami a údajmi, osadené používateľom.
2C00H - 2FFFH	1	
3000H - 33FFH	2	
3400H - 37FFH	3	
... - ...	...	
7C00H - 7FFFH	21	

Poznámka: Pre prenesenie väčšieho množstva bajtov ako 1 KB je možné využiť podprogram MONITOR-a TRANSF (pozri kap. 5.4).

## Príklad:

Prenesenie programu o veľkosti 1 KB z ROM-modulu do operačnej pamäte, ktorý je trvalo uložený v druhej polovici pamäte EPROM (typu 2716) umiestnenej v prvej pozícii za interpreterom BASIC-G:

100 ROM 1

# RUN

Organizačný príkaz RUN

Opis: Po predbežnej inicializácii začne vykonávať program od zadaného riadka.

Formát: RUN alebo RUN r

Parameter: r je číslo riadka, od ktorého sa má program vykonať.

Poznámka: Pri inicializácii sa vynulujú všetky premenné a textovým premenným sa priradia reťazce nulovej dĺžky. Príkaz RUN r je teda ekvivalentný so sledom príkazov CLEAR :  
RESTORE : GOTO r.  
Ak sa nezadá číslo riadka, program sa začne vykonávať od prvého príkazu.

Príklad:

Ilustrácia príkazov RUN:

```
100 A=1 : GOSUB 170
120 A=2 : GOSUB 170
140 A=3 : GOSUB 170
160 END
170 PRINT A
180 RETURN
```

RUN

1

2

3

RUN 140

3

# SAVE

Organizačný príkaz SAVE

**Opis:** Uloží program na magnetofónovú kazetu a označí ho zadá-  
ným číslom a menom. Ak sa meno začína znakom "\*", ulože-  
ný program je samoštartovateľný s ochranou.

**Formát:** SAVE f "meno

**Parametre:** f je číslo z intervalu 1..99, pod ktorým sa program  
uloží na magnetofónovú kazetu  
"meno je reťazec znakov (max. 8) a reprezentuje meno  
súboru, pod ktorým sa program uloží na magnetofó-  
novú kazetu.

**Poznámka:** Ak chceme mať chránený program, uložíme ho na kazetu  
tak, že prvý znak mena je "\*". Keď neskôr zavádzame ten-  
to program do operačnej pamäte príkazom LOAD, program sa  
ihneď odštartuje (ako keby bol zadáný príkaz RUN) a nedá  
sa zastaviť klávesom STOP. Ak prerušíme program klávesom  
RST, znovu sa odštartuje BASIC-G a program sa vymaže.

**Príklad:**

Program v operačnej pamäti sa uloží na magnetofónovú kazetu pod  
číslom 11. Bude uložený s ochranou a menom "PROG"

```
SAVE 11"*PROG
```

# SCALE

Grafický príkaz SCALE

Opis: Slúži na nastavenie mierky okna SUOG obrazovky. Parametrami sú rozsahy zobrazovania hodnôt premenných.

Formát: SCALE e1,e2,e3,e4

Parametre: e1 je výraz, reprezentujúci minimálnu hodnotu v smere osi x (ľavá hranica),  
e2 je výraz, reprezentujúci maximálnu hodnotu v smere osi x (pravá hranica).  
e3 je výraz, reprezentujúci minimálnu hodnotu v smere osi y (dolná hranica).  
e4 je výraz, reprezentujúci maximálnu hodnotu v smere osi y (horná hranica).

Poznámka: Zobrazenie funkcie ďalšími grafickými príkazmi sa deje v zadanej mierke v celom okne SUOG. Napríklad pri zadaní mierky príkazom:

```
SCALE -1,1,-1,1
```

bude stred súradníc v strede okna SUOG, bod -1,-1 vľavo dole, bod 0,1 uprostred hore atď.

e1 musí byť rôzne od e2 a e3 musí byť rôzne od e4.

Príklad:

Zobrazenie histogramu percentuálnych hodnôt. Zobrazených bude 10 hodnôt po celej obrazovke, t. j. výsledkom bude 10 zafarbených stĺpčekov s maximálnou výškou až po horný okraj (ak je zobrazovaná hodnota = 100%):

```
100 REM H I S T O G R A M
110 GCLEAR
120 SCALE 0,10,0,242
130 FOR I = 0 TO 9
140 READ A
150 IF A<1 THEN 180
160 MOVE I,0
170 FILL 15,2*A;1
180 NEXT I
190 AXES 0,0
190 DATA 23,13,0,58,64,80,25,60,22,2
```



# SGN

Funkcia SGN

Opis: Vracia znamienko parametra (-1, ak je hodnota parametra záporná, 0, ak je hodnota nulová, a 1, ak je hodnota parametra kladná).

Formát: SGN(e)

Parameter: e je numerická premenná alebo aritmetický výraz.

Príklad:

Výpočet odmocniny zadaného čísla. Vopred sa testuje, či parameter nie je menší ako 0:

```
100 DISP "ZADAJ X PRE SQR (DRUHU ODMOCNINU) "  
110 INPUT K  
120 IF SGN(K)=-1 THEN GOTO 170  
130 PRINT K,SQR(K)  
140 IF SGN(K)=0 THEN 160  
150 GOTO 100  
160 END  
170 PRINT K;"ZAPORNE CISLO. SQR NEEEXISTUJE"  
180 GOTO 100
```

RUN

```
ZADAJ X PRE SQR (DRUHU ODMOCNINU)  
-9  
-9 ZAPORNE CISLO. SQR NEEEXISTUJE  
ZADAJ X PRE SQR (DRUHU ODMOCNINU)  
9  
9 3  
ZADAJ X PRE SQR (DRUHU ODMOCNINU) 0  
0 0
```

# SIN

## Funkcia SIN

Opis: Vracia sínus zadaného uhla. Parameter je v radiánoch alebo v stupňoch.

Formát: SIN(e)

Parameter: e je numerická konštanta, numerická premenná alebo aritmetický výraz.

Poznámka: Voľbu, či má byť parameter v stupňoch alebo v radiánoch, uskutočníme príkazmi DEG alebo RAD ešte pred príkazom SIN(e). Štandardne platí RAD.

### Príklad:

Graf funkcie  $\sin(x)$  (parameter je v stupňoch):

```
100 GCLEAR
110 SCALE -360,360,-2,2
120 AXES 0,0
130 MOVE -360,0
140 DEG
150 FOR X=-360 TO 360 STEP 5
160 PLOT X,SIN(X),1
170 NEXT X
```

# SQR

Punkcia SQR

Opis: Vracia druhú odmocninu nezáporného parametra.

Formát: SQR(e)

Parameter: e je nezáporná numerická konštanta alebo nezáporný aritmetický výraz.

Príklad:

Výpis odmocniny zadaného čísla:

```
100 DISP "ZADAJ ARGUMENT PRE SQR (DRUHU ODMOCNINU) "  
110 INPUT K  
120 PRINT K,SQR(K)  
130 END  
RUN  
ZADAJ ARGUMENT PRE SQR (DRUHU ODMOCNINU)  
9  
9          3
```

# STATUS

## Funkcia STATUS

Opis: Vracia obsah zvoleného v/v registra zo zvoleného v/v kanálu.

Formát: STATUS k,r

Parametre: k je číslo v/v kanálu (0..7).  
r je adresa v/v registra vo zvolenom kanále (pozri tabuľku B.3)

Poznámka: Funkcia slúži na jednoduché priradenie obsahu v/v registra do zvolenej premennej.

Podrobnejšie o práci s v/v kanálmi pozri prílohy A a B.

### Príklad:

Inicializácia kanálu 4 tak, aby všetky brány boli vstupné, priradenie obsahu brány C do premennej PC a výpis hodnoty siedmeho bitu tejto brány:

```
100 CONTROL 4,3;155: REM =9BH
```

```
110 PC=STATUS 4,2
```

```
120 PRINT BIT PC,7
```

```
RUN
```

```
0
```

Výsledok znamená, že na vstupe brány nebolo pripojené nič (čo sa prejavilo ako log. 0), log. 1 možno overiť tak, že sa táto privedie pomocou drôtenej prepojky na konektore GPIO/0 (pozri prílohu A.1 a obrázok A.3):

```
RUN
```

```
1
```

# STOP

Príkaz STOP

Opis: Zastaví vykonávanie programu a do dialógového riadka vypíše, na ktorom riadku sa program zastavil.

Formát: STOP

Poznámka: Príkaz je vhodný na testovanie programu vo zvolených bodoch. Ak chceme pokračovať v ďalšom vykonávaní programu, nesmieme pri zastavení meniť obsah programu, ale môžeme zisťovať a meniť obsah premenných.

Príklad:

Fragment programu na výpočet náhodných hodnôt. Po každom výpočte sa vykonávanie programu zastaví a náhodnú hodnotu vypíšeme:

```
100 FOR I=0 TO 99
110 RN = RND(1)
120 STOP
130 NEXT I
```

Po zastavení napíšeme do dialógového riadka:

```
PRINT RN
CONT
```

a program pokračuje príkazom v riadku 130.

Príklad:

Podobné použitie príkladu STOP pozri aj v príklade pri príkaze CONT.

# STR\$

Punkcia STR\$

Opis: Vracia reťazec, ktorý je dekadickou reprezentáciou hodnoty parametra.

Formát: STR\$(e)

Parameter: e je aritmetický výraz.

Príklad:

Program, ktorý zisťuje, či číslo zadané z klávesnice sa končí číslicou 3:

```
100 DISP "ZADAJ CISLO. 0 - KONIEC"
110 INPUT K
120 IF K=0 THEN END
130 K$ = STR$(K)
140 IF RIGHT$(K$,1) ="3" THEN 170
150 PRINT K;" CISLO NEKONCI CISLICOU 3"
160 GOTO 100
170 PRINT K;" CISLO KONCI CISLICOU 3"
180 GOTO 100
RUN
ZADAJ CISLO. 0 - KONIEC
123
123 CISLO KONCI CISLICOU 3
ZADAJ CISLO. 0 - KONIEC
234
234 CISLO NEKONCI CISLICOU 3
ZADAJ CISLO. 0 - KONIEC
0
```

# TAN

## Funkcia TAN

Opis: Vracia tangens uhla, ktorý je zadaný v radiánoch alebo v stupňoch.

Formát: TAN(e)

Parameter: e je numerická konštanta, numerická premenná alebo aritmetický výraz.

Poznámka: Voľbu, či je parameter v radiánoch alebo v stupňoch, vykonáme pred použitím príkazu TAN(e) príkazmi RAD alebo DEG. Štandardne platí RAD.

## Príklad:

Výpis tangensu hodnoty zadanej z klávesnice a kontrola ohyby:

```
100 DEG : DISP "ZADAJ UHOL:" : INPUT U : W=-90
110 IF ABS(U)>1500 THEN END
120 FOR I=1 TO 8
130 IF U=W OR U=ABS(W) THEN 180
140 W=W-180 : IF ABS(U)<ABS(W) THEN 160
150 NEXT I
160 PRINT U,TAN(U)
170 IF TAN(U)=0 THEN END
175 GOTO 100
180 PRINT "TAN(";U;")    NIE JE DEFINOVANY" : GOTO 100
RUN
ZADAJ UHOL PRE VYPOCET TANGENS
630
TAN( 630 )    NIE JE DEFINOVANY
ZADAJ UHOL PRE VYPOCET TANGENS
0
0                0
```

# USR

## Funkcia USR

**Opis:** Vyvolá podprogram v strojovom kóde uložený od zadanej adresy v operačnej pamäti. Vracia obsah akumulátora.

**Formát:** USR(e)

**Parameter:** e je numerická konštanta alebo aritmetický výraz reprezentujúci adresu v operačnej pamäti, od ktorej je podprogram v strojovom kóde uložený.

### Príklad:

Inštrukcie podprogramu v strojovom kóde vytvoríme v príkaze DATA (dekadickým vyjadrením) a prenese ich do zvolenej časti operačnej pamäte (adresa 7000H). Potom sa tento podprogram vykoná. Ide o podprogram na výpis znaku (porovnaj s príkladom pri príkaze CODE).

```
120 A=28672: REM =7000H
130 FOR I=0 TO 5
140 READ D
150 POKE A+I,D
160 NEXT I
170 DATA 62,65,205,0,229,201
180 X=USR(A)
RUN
A
```



# VAL

Funkcia VAL

Opis: Vracia hodnotu ľubovoľného výrazu, ktorý sa zadá ako reťazec znakov.

Formát: VAL(a\$)

Parameter: a\$ je reťazec, ktorý obsahuje ľubovoľný výraz v BASIC-G.

Príklad:

Program na vytlačenie tabuľky hodnôt 1 až 5 funkcie, zadanej používateľom:

```
100 DISP "ZADAJTE FUNKCIU PREMENNEJ X" : INPUT F$
110 GCLEAR : PRINT "X", F$
120 FOR X=1 TO 5
130 PRINT X,VAL(F$)
140 NEXT X
```

Ak zadáme programu reťazec X\*X, dostaneme výstup:

X	X*X
1	1
2	4
3	9
4	16
5	25

# WAIT

Príkaz WAIT

Opis: Čaká na nastavenie zadaného bitu (z log.0 na log.1) v zadanom v/v registri.

Formát: WAIT p,b,m

Parametre: p je výraz reprezentujúci fyzickú adresu v/v registra  
b je aritmetický výraz s hodnotou z intervalu (0..255) vyjadrujúci polohu testovaných bitov v bajte.  
m je maska, reprezentovaná aritmetickým výrazom (0..255), udávajúca, ktoré bity majú byť testované opačne (t.j. na nastavenie z log.1 na log.0). V podstate ide o vykonanie XOR pred testovaním.

Príklad:

Čakanie, kým sa na bráne C interfejsu GPIO/0 nastaví bit 4 alebo 5 z hodnoty 0, na hodnotu 1. Binárny profil testovaných bitov je 00110000, čo je dekadický 48.

```
100 CONTROL 4,3;155:PC=78
110 PRINT "Čakám na nastavenie bitu."
120 WAIT PC,48
130 PRINT "Bit sa nastavil !"
RUN
Čakám na nastavenie bitu.
Bit sa nastavil !
```

Poznámka: Fyzické nastavenie bitu na úroveň log.1 možno urobiť pomocou drôtenej prepojky na konektore K3 (pozri prílohu A.1), s využitím invertora.

# WINDOWA

Príkaz WINDOWA

Opis: Nastavuje rozmery okna SYOG obrazovky. Príkaz bez uvedenia parametrov ruší predchádzajúcu definíciu okna SYOG.

Formát: WINDOWA e1,e2,e3,e4

Parametre: e1 je výraz pre ľavý okraj okna SYOG (hodnota z intervalu 0..47).  
e2 je výraz pre pravý okraj okna SYOG (hodnota z intervalu 0..47).  
e3 je výraz pre horný okraj okna SYOG (hodnota z intervalu 0..255).  
e4 je výraz pre dolný okraj okna SYOG (hodnota z intervalu 0..255).

Poznámka: Ľavý a pravý okraj okna môžu byť len v niektorom stĺpci (po 6 bitov), horný a dolný okraj v ktorejkoľvek linke obrazovky. V príkaze PRINT sa podľa šírky okna nastaví patričný počet stĺpcov na výpis (mení sa teda funkcia oddeľovača ",").

Parametre e1 a e2 sú ekvivalentné s parametrom e1, parametre e3 a e4 s parametrom e2 v príkaze BMOVE.

Príklad:

Zadefinovanie okna SYOG v pravej hornej časti obrazovky:

```
100 WINDOWA 40,47,0,42
```

# WINDOWG

Príkaz WINDOWG

**Opis:** Definuje okno SUOG na obrazovke podľa zadaných parametrov. Príkaz bez uvedenia parametrov ruší predchádzajúcu definíciu okna SUOG.

**Formát:** WINDOWG e1,e2,e3,e4

**Parametre:**

- e1 je výraz pre ľavý okraj okna SUOG (hodnota z intervalu 0..47).
- e2 je výraz pre pravý okraj okna SUOG (hodnota z intervalu 0..47).
- e3 je výraz pre horný okraj okna SUOG (hodnota z intervalu 0..255).
- e4 je výraz pre dolný okraj okna SUOG (hodnota z intervalu 0..255).

**Poznámka:** Pravý a ľavý okraj môžu byť len v niektorom zo stĺpcov (po 6 bitov), horný a dolný okraj môžu byť v ktorejkoľvek linke obrazovky. Príkaz SCALE na definovanie mierky môže byť pred príkazom, alebo po príkaze WINDOWG (pracuje sa vždy podľa posledného uvedeného príkazu). Parametre e1 a e2 sú ekvivalentné s parametrom e1, parametre e3 a e4 s parametrom e2 príkazu BMOVE.

**Príklad:**

Zadefinovanie okna SUOG v spodnej polovici obrazovky:

```
100 WINDOWG 0,47,128,255
```

?

Príkaz ?

Opis: Výpis zoznamu parametrov do dialógového riadka a čakanie na vstup ľubovoľného klávesu.

Formát: ? z

Parameter: z je zoznam výrazov, ktoré sa vypíšu do DR (pozri príkaz PRINT).

Príklad:

Fragment programu, v ktorom sa bude pokračovať až po stlačení niektorého klávesu:

```
330 ...  
340 ? "Priprav tlačiareň, potom stlač kláves!"  
350 PRINT#43;...  
360 ...
```

## 5. MONITOR

### 5. MONITOR

MONITOR je základné programové vybavenie, ktoré umožňuje komunikáciu s mikropočítačom na najnižšej úrovni. Vytvára isté oddelenie a súčasne prepojenie medzi technickým vybavením mikropočítača a používateľom. MONITOR je uložený rezidentne v pamätiach typu EPROM na module procesora a v rámci celého pamäťového priestoru sa nachádza v časti od adresy 57344 po 65535 (resp. od E000H po FFFFH).

Základné funkcie MONITOR-a sú:

- inicializácia systému, v rámci ktorej sa vykonáva autodiagnostika mikropočítača
- prezeranie a modifikovanie údajov v operačnej pamäti
- spustenie používateľského programu
- uloženie programu alebo údajov z operačnej pamäte na magnetofónovú kazetu
- uloženie programu alebo údajov z magnetofónovej kazety do operačnej pamäte a ich kontrola
- vytvorenie režimu kompatibility s PMD 85-2
- vytvorenie režimu terminál počítača SMEP

Prvá funkcia (inicializácia systému) sa vykoná automaticky po zapnutí mikropočítača alebo súčasným stlačením klávesov SHIFT a RST. V rámci autodiagnostiky, ktorá je súčasťou tejto funkcie, sa vykoná test pamätí ROM na kontrolný súčet a test pamätí RWM pomocou zápisu a čítania 4 vzoriek (hexadecimálne 00, 55, AA a FF). Ak sú údaje v ROM-pamäti porušené, vypíše sa **do DR** oznam:

++ Rom error ++      alebo    ++ BASIC-rom error ++

V prípade chyby dynamických pamätí RWM sa do DR vypíše správa:

```
++ Ram error ++
```

Zároveň sa do okna SYOG vypíše adresa RWM, kde sa chyba detekovala a čaká sa na stlačenie klávesy. Po stlačení klávesy sa pokračuje v teste RWM.

Ostatné funkcie sa aktivujú zadávaním príslušných príkazov. Tieto sú opísané v kapitole 5.1.

Komunikovať s programom MONITOR (t.j. zadávať príkazy) je možné vtedy, keď je tento program aktívny. Tento prípad nastane po zapnutí mikropočítača, ak nebol zasunutý ROM-modul (pozri kap. 2.2) alebo v prípade aktívneho interpretera BASIC-G súčasným stlačením kláves SHIFT a RCL.

Štruktúra programu MONITOR sa vyznačuje dôsledným použitím techniky podprogramov. V náročnejších aplikáciách je možné využiť rôzne služby, ktoré sú v nich už realizované, volaním príslušného podprogramu z používateľského programu. Opis jednotlivých podprogramov a spôsob ich volania sú v kapitole 5.4.

## 5.1. Príkazy MONITOR-a

Príkazy sa zadávajú z klávesnice mikropočítača zadaním mena príkazu a príslušných parametrov (presne tak, ako to vyžaduje jeho formát uvedený v opise príkazu) a príkaz sa ukončí stlačením klávesu EOL. Ak sa príkaz vykonal správne, je možné zadať ďalší príkaz. V prípade, že príkaz sa nedá korektne vykonať, vypíše MONITOR chybové hlásenie a opravený príkaz je možné opätovne zadať.

# DUMP

## Príkaz DUMP

**Opis:** Príkaz DUMP vypisuje obsah operačnej pamäte na zobrazovaciu jednotku.

**Formát:** DUMP adr

adr - štvorciferné hexadecimálne číslo udávajúce adresu začiatku výpisu

**Poznámka:** Vypisuje sa v hexadecimálnom aj znakovom tvare. Výpis prebieha dovtedy, kým ho nepozastavíme pridržením klávesu SHIFT alebo nezastavíme klávesom STOP.

**Príklad:** DUMP E000

Po odoslaní príkazu začne výpis oblasti pamäte, v ktorej je uložený MONITOR. Po dosiahnutí adresy FFFFH pokračuje výpis od 0000H.



# JOB

Príkaz JOB

Opis: Príkaz JOB načíta do operačnej pamäte oblasť pamäte z ROM-modulu a odovzdá riadenie na začiatok oblasti.

Formát: JOB romadrlengthrwmadr

operandy sú štvorciferné hexadecimálne čísla:

romadr - adresa v ROM-module

length - počet prenášaných bajtov

rwmadr - adresa v operačnej pamäti, kde sa oblasť ukladá  
a odkiaľ sa štartuje

Príklad: JOB 000004007000

V prípade, že je pripojený ROM-modul prenesie sa 1KB (1KB=1024B=0400H) do operačnej pamäte od adresy 7000H a program sa odštartuje.

# JUMP

## Príkaz JUMP

Opis: Príkaz JUMP odštartuje program od danej adresy.

Formát: JUMP adr

adr - štvorciferné hexadecimálne číslo udávajúce adresu prvej inštrukcie programu

Poznámka: Príkaz odovzdá riadenie na príslušnú adresu a dané bajty interpretuje ako inštrukcie mikroprocesora MHB8080.

Príklad: JUMP 0000

V prípade, že v pamäti je kópia BASIC-u, odštartuje sa BASIC.

# MEM

## Príkaz MEM

**Opis:** Príkaz MEM vypíše obsah 16 bajtov operačnej pamäte do dialógového riadka a pripraví ich modifikáciu.

**Formát:** MEM adr

adr - štvorciferné hexadecimálne číslo udávajúce adresu začiatku výpisu

**Poznámka:** Výpis je v hexadecimálnom tvare a začína sa kľúčovým slovom SUB, čo umožňuje dané bajty potvrdiť (klávesom EOL) alebo zmeniť. Ďalšia reakcia MONITOR-a je ako po príkaze SUB.

**Príklad:** MEM 2400

Po odoslaní príkazu sa vypíše prvých 16 bajtov z oblasti na uloženie basicovského programu.

# MGEND

## Príkaz MGEND

**Opis:** Príkaz MGEND preverí záznam na magnetofónovej kazete testovaním na kontrolnú sumu celého súboru, prípadne každého bloku (pri blokovom zázname).

**Formát:** MGEND int

int - dvojmiestne dekadické číslo (00 až 99) udávajúce číslo súboru

**Poznámka:** Vykonanie príkazu nezmení obsah operačnej pamäte, záznam sa číta "naprázdno".

Ak je záznam bezchybný, vymaže sa dialógový riadok, ak je chybný, vypíše sa oznam:

```
+++ File error +++
```

Vykonávanie príkazu môžeme prerušiť klávesom STOP. V prípade, že použijeme číslo 00, príkaz skontroluje súbor, ktorého hlavičku prvú prečíta.

**Príklad:** MGEND 23

Príkaz číta hlavičky záznamov na magnetofónovej kazete a vypisuje ich na zobrazovaciu jednotku. Keď nájde binárny záznam číslo 23, skontroluje ho a odovzdá riadenie MONITOR-u.

# MGLD

## Príkaz MGLD

**Opis:** Príkaz MGLD načíta súbor z magnetofónovej kazety do operačnej pamäte.

**Formát:** MGLD int

int - dvojmiestne dekadické číslo (00 až 99) udávajúce číslo záznamu

**Poznámka:** Príkaz načíta len súbor vytvorený príkazom MGSV. Súbory nie sú premiestniteľné a ukladajú sa do oblasti, ktorá bola uvedená v príkaze MGSV. Formát príkazu je rovnaký pre štandardný aj blokový záznam. Príkaz vykonáva aj kontrolu súboru.

**Príklad:** MGLD 00  
Do operačnej pamäte sa prepíše prvý záznam, ktorého hlavička sa prečíta.

# MGSV

## Príkaz MGSV

**Opis:** Príkaz MGSV zapíše oblasť operačnej pamäte do súboru na magnetofónovej kazete.

**Formát:** MGSV int , adr1 - adr2 sep text

int - dvojmiestne dekadické číslo (00 až 99) udávajúce číslo záznamu

adr1,adr2 - štvormiestne hexadecimálne čísla udávajúce začiatok a koniec pamäťovej oblasti

sep - oddeľovač určujúci typ záznamu:

" - štandardný (sekvenčný) záznam

: - blokový záznam

text - max. 8 znakov, ktoré sa zapíšu do hlavičky súboru

**Poznámka:** Po odoslaní príkazu sa údaje začnú prenášať bez ohľadu na chod magnetofónu.

V prípade, že vynecháme operandy sep a text, súbor sa nahrá štandardne, v hlavičke bude zapísané AS. Ak vynecháme len operand text, v hlavičke budú medzery.

**Príklad:** MGSV 04,0000-2400:BEJZIK

Ak je pripojený magnetofón, po odoslaní príkazu nahráme na magnetofónovú kazetu časť interpretera BASIC-G ako záznam číslo 04.

# SUB

Príkaz SUB

Opis: Príkaz SUB umožňuje meniť obsah operačnej pamäte.

Formát: SUB adr data data ...

adr - štvormiestne hexadecimálne číslo udávajúce začiatok oblasti

data - dvojmiestne hexadecimálne číslo alebo znak vnútorného kódu PMD 85-3 v apostrofoch. Maximálny počet operandov data je obmedzený iba dĺžkou dialógového riadka.

Poznámka: Po odoslaní príkazu máme v dialógovom riadku pripravený príkaz SUB s nasledujúcou adresou. Z režimu editovania (neustále ponúkание príkazu SUB) sa dostaneme napr. vymazaním dialógového riadka a zadaním iného príkazu.

Príklad: SUB 6E00 50 4D "DPM" 44

Od adresy 6E00H naplníme 6 bajtov hodnotami 50 4D 44 50 4D 44 (znakovo PMDPMD). V dialógovom riadku sa objaví SUB 6E06\_ .

## 5.2. Režim compatibility

**Funkcia:** MONITOR PMD 85-3 umožňuje prechod do režimu PMD 85-2, v ktorom sa dajú spúšťať programy nahrané v MONITOR-e nižšej verzii. Pri prechode do tohto režimu sa prvé 4kB MONITOR-a skopírujú od adresy 8000E, preadresujú sa volania a skoky a vykoná sa štart posunutého MONITOR-a. Do kompatibilného režimu môžeme prejsť pred nahrávaním programu alebo pred jeho štartovaním.

Prechod do režimu compatibility: JUMP FFF0

Návrat do MONITOR-a PMD 85-3: resetovaním mikropočítača

**Poznámka:** Pre lepšiu orientáciu používateľa sa MONITOR pri vstupe do režimu compatibility ohlasuje správou

++ Os ready ++

na odlíšenie od výpisu

++ OS ready ++

v pôvodnom režime.

Na rozdiel od skutočného PMD 85-2 je v režime PMD 85-2 MONITOR umiestnený v RWM pamäti, takže môže prísť k jeho prepísaniu.



## 5.3. Režim terminálu

Funkcia: Umožňuje použiť PMD 85-3 ako terminál počítača.  
Spojenie sa realizuje cez interfejs IRPS (pozri prílohu A.3) v asynchrónnom, duplexnom režime s rýchlosťou prenosu 4800 Baud, ktorú je možno zmeniť zmenou hodnoty premennej SPEED (C07CH) podľa vzorca:

$$\text{rýchlosť prenosu} = 128000 / \text{SPEED}$$

Modul bol vyvinutý s prihliadnutím na počítače SMEP, je však možné aj spojenie s inými počítačmi.

Vstup do režimu terminálu: stlačením kláves SHIFT a C-D

Zrušenie režimu terminálu: stlačením kláves SHIFT a RCL

Komunikačný modul TERMINAL má okrem štandardných funkcií terminálu aj tri nové funkcie, ktoré umožňujú blokový prenos údajov medzi nadriadeným počítačom a PMD-85. Ovládajú ich príkazy riadiaceho počítača INBLOCK, OUTBLOCK a JUMP.

Príkaz INBLOCK

- prenesie na danú adresu v operačnej pamäti PMD 85-3 blok 128 bajtov.

Formát: 1.bajt - 04H (kód príkazu)  
2.-3.bajt - adresa v PMD 85-3  
4.-131.bajt - blok 128 prenášaných bajtov

#### Príkaz OUTBLOCK

- vyšle z danej adresy v PMD 85-3 blok 128 bajtov na výstupnú linku.

Formát:       1.bajt - 02H (kód príkazu)  
              2.-3.bajt - adresa v PMD 85-3

#### Príkaz JUMP

- odovzdá riadenie na danú adresu v PMD 85-3 a uloží návratovú adresu na vrch zásobníka.

Formát:       1.bajt - 05H (kód príkazu)  
              2.-3.bajt - adresa v PMD 85-3

V prípade spojenia s nadriadeným počítačom SMEP pracujúcim pod operačným systémom DOS RV treba nastaviť nasledujúce charakteristiky:

```
SET /SPEED=TTx:4800:4800
SET /C2111=TTx:
SET /HFILL=TTx:6
SET /CRT=TTx:
SET /BUF=TTx:48
```

## 5.4. Podprogramy MONITOR-a

Podprogramy sú použiteľné v programoch v strojovom kóde aj v jazyku BASIC-G, v druhom prípade sa však nepredpokladá ich časté využívanie, pretože väčšina služieb je pohodlnejšie zaistená na úrovni príkazov jazyka BASIC-G.

Spôsob volania:

v strojovom kóde:	inštrukciou typu CALL
v jazyku BASIC-G:	funkciou USR

Príklad použitia podprogramu DITXT2 v programe v strojovom kóde:

symbolický tvar	strojový kód
...	...
LXI H,3000	21 00 30
SHLD C074	22 74 C0
CALL EA8C	CD 8C EA
...	...

Príklad použitia podprogramu TESTKY v programe v BASIC-u

```
...
195 VPRAVO=1010
205 VLAVO=1080
210 K=USR('E9C0)
221 IF K=24 GOTO VPRAVO
222 IF K=8 GOTO VLAVO
230 GOTO 210
...
```

Prehľad podprogramov MONITOR-a

názov	adresa	význam
BELL	E8A3H	Zvukový signál
BINBCD	EE73H	Prevod z binárneho kódu do kódu BCD
BINHED	E125H	Zobrazenie bajtu v hexa tvare
BINHEM	E13BH	Prevod z hexa do vnútorného kódu PMD
DCHAR1	E500H	Zobrazenie znaku
DCHAR2	E584H	Zobrazenie znaku
DITXT1	E858H	Výpis textu
DITXT2	EA8CH	Výpis textu
DPOINT	EC7DH	Zobrazenie bodu
EDIT	E800H	Interpretovanie kódu klávesu
ENTER	EBEEH	Vstup z dialógového riadka
EXIT	EC40H	Prechod do MONITOR-a
HEBIN1	E0E0H	Prevod z hexa do binárneho kódu
HEBIN2	E0F7H	Prevod z hexa do binárneho kódu
HEBIN4	E109H	Prevod z hexa do binárneho kódu
INPOL	ECD0H	Zobrazenie úsečky
REBIG	E4F3H	Zistenie stavu šestice bodov
RESUG	E5E6H	Zistenie stavu bodu
RECALL	E855H	Výpis bafra dialógového riadka
WAITKY	E4A1H	Čakanie na stlačenie klávesu
TESTKY	E9C0H	Test klávesnice
TESTOP	EC74H	Test klávesu STOP
TPREAD	EDE2H	Čítanie z magnetofónovej kazety
TPWRTE	EF60H	Zápis na magnetofónovú kazetu
TRANSF	EC00H	Prenos údajov z ROM-modulu

Podprogram BELL

Funkcia: Aktivuje zvukový signál

Adresa: E8A3H

Vstup: žiadny

Výstup: zvukový signál

Registre: mení PSW, DE, HL

Podprogram BINBCD

Funkcia: Konvertuje binárny obsah A do kódu BCD.

Adresa: EE73H

Vstup: A (s obsahom 00H - 63H)

Výstup: A (00H - 99H)

Registre: mení PSW, B, HL

#### Podprogram BINHED

Funkcia: Konvertuje binárny obsah A na dva znaky, reprezentujúce hexadecimálny zápis hodnoty akumulátora zobrazí ho v okne SYOG.

Adresa: E125H

Vstup: A

Výstup: podprogram DCHAR1 (E500H)

Registre: mení PSW,B

#### Podprogram BINHEM

Funkcia: Konvertuje binárny obsah A na dva znaky vo vnútornom kóde PMD 85-3 a ukladá ich do pamäte.

Adresa: E13BH

Vstup: A  
HL obsahujúci adresu pamäte, na uloženie 2 znakov

Výstup: pamäťové miesto určené v HL

Registre: mení PSW,B  
HL zvýši o 1

#### Podprogram DCHAR1

Funkcia: Zobrazuje obsah akumulátora ako znak na mieste, ktoré určuje kurzor okna SYOG.

Adresa: E500H

Vstup: A  
C03EH-C03FH - poloha kurzora v okne SYOG

Výstup: zobrazovacia jednotka

Registre: mení PSW

#### Podprogram DCHAR2

Funkcia: Zobrazuje obsah akumulátora ako znak na mieste, ktoré je určené v registri HL.

Adresa: E584H

Vstup: A  
HL určujúci polohu na obrazovke

Výstup: zobrazovacia jednotka

Registre: mení PSW

#### Podprogram DITXT1

Funkcia: Vypisuje do dialógového riadku reťazec znakov z pamäte.

Adresa: E858H

Vstup: 48 bajtový reťazec v operačnej pamäti  
adresa začiatku reťazca je uložená v registri HL

Výstup: dialógový riadok

Registre: mení všetky

#### Podprogram DITXT2

Funkcia: Vypisuje do dialógového riadka reťazec znakov z operačnej pamäte.

Adresa: EA8CH

Vstup: reťazec v pamäti (max.80 bajtov) ukončený znakom 0DH ad-  
resa začiatku reťazca je uložená v pamäti na adrese C074H  
(2 bajty)

Výstup: dialógový riadok

Registre: mení všetky



#### Podprogram DPOINT

Funkcia: Zobrazuje bod, na ktorý ukazuje kurzor okna súradnicovej grafiky v závislosti od farby, módu a stavu bodu.

Adresa: EC7DH

Vstup: C170H - súradnica x  
C172H - súradnica y  
C03AH - farba  
C1FAH - mód

Výstup: zobrazovacia jednotka

Registre: mení PSW

#### Podprogram EDIT

Funkcia: Interpretuje kód klávesu (z adresy C134H) ako alfanumerický znak alebo riadiaci znak dialógového riadka.

Adresa: E800H

Vstup: kód klávesu na adrese C134H

Výstup: zobrazenie znaku do dialógového riadku alebo príslušná editačná činnosť

Registre: mení PSW

#### Podprogram ENTER

Funkcia: načítava obsah dialógového riadka do kľúča RCL a do pamäte. Daný reťazec sa ukončí znakom 0DH.

Adresa: EBEEH

Vstup: dialógový riadok  
2 bajty na adrese C078H, ktoré určujú adresu pamäte na uloženie reťazca

Výstup: pamäťová oblasť určená v C078H  
kľúč RCL

Registre: mení všetky

#### Podprogram EXIT

Funkcia: Ukončuje činnosť programu a zabezpečuje vstup do MONITOR-a.

Adresa: EC40h

Vstup: žiadny

Registre: mení všetky

#### Podprogram HEBIN1

Funkcia: Konvertuje hexadecimálnu číslicu vyjadrenú ako znak vo vnútornom kóde PMD 85-3 na binárne číslo.

Adresa: E0E0H

Vstup: A obsahujúci hexa číslicu (30H - 39H, 41H - 46H)

Výstup: A  
pri chybe nasadí príznak CY

Registre: mení PSW

#### Podprogram HEBIN2

Funkcia: Konvertuje dve hexadecimálne číslice vo vnútornom kóde PMD 85-3 na binárne číslo.

Adresa: E0F7H

Vstup: HL obsahujúci adresu začiatku 2 bajtovej oblasti pamäte, v ktorej sú hexa číslice uložené

Výstup: A  
pri chybe nasadí príznak CY

Registre: mení PSW, B  
HL zvýši o 2

#### Podprogram HEBIN4

Funkcia: Konvertuje štyri hexadecimálne čísllice vo vnútornom kóde  
PMD 85-3 na binárne číslo.

Adresa: E109H

Vstup: HL obsahujúci adresu začiatku 4 bajtovej oblasti pamäte,  
v ktorej sú hexa čísllice uložené

výstup: DE  
pri chybe nasadí príznak CY

Registre: mení PSW, A, B, C, D  
HL zvýši o 4

#### Podprogram INPOL

Funkcia: Kreslí úsečku z pozície kurzora do určeného bodu - inter-  
poluje. Kurzor sa presunie do nového bodu.

Adresa: ECD0H

Vstup: C170H - x-ová súradnica kurzora  
C172H - y-ová súradnica kurzora  
C173H - x-ová súradnica cieľového bodu  
C174H - y-ová súradnica cieľového bodu

Výstup: zobrazovacia jednotka

Registre: mení všetky

#### Podprogram REBIG

Funkcia: Zistí hodnotu šestice bodov podľa polohy kurzora binárnej grafiky a uloží ju do akumulátora.

Adresa: E4F3H

Vstup: hodnota bajtu na adrese C17AH (kurzor BG)

Výstup: A

Registre: mení PSW, HL

#### Podprogram RESUG

Funkcia: Zistí stav bodu podľa polohy kurzora súradnicovej grafiky a výsledok uloží do akumulátora.

Adresa: E5E6H

Vstup: hodnoty bajtov na adresách:  
C170H - súradnica x (00H-FFH)  
C172H - súradnica y (00H-FFH)

Výstup: A  
ak bod svieti, A = 01  
ak nesvieti, A = 00

Registre: mení všetky

#### Podprogram RECALL

Funkcia: Vypisuje do dialógového riadka obsah kľúča RCL.

Adresa: E855H

Vstup: kľúč RCL

Výstup: dialógový riadok

Registre: mení všetky

#### Podprogram WAITKY

Funkcia: Čaká na stlačenie nejakého klávesu, ak sa stane, uloží jeho kód do A a na adresu C134H.

Adresa: E4A1H

Vstup: klávesnica

Výstup: A  
hodnota bajtu na adrese C134H

Registre: mení PSW

#### Podprogram TESTKY

Funkcia: Testuje klávesnicu (okrem klávesu STOP), ak nájde stlačený kláves uloží jeho kód do A a na adresu C134H, v opačnom prípade A = 0.

Adresa: E9C0H

Vstup: klávesnica

Výstup: A  
obsah bajtu na adrese C134H

Registre: mení PSW

#### podprogram TESTOP

Funkcia: Testuje kláves STOP.

Adresa: EC74H

Vstup: kláves STOP

Výstup: Akumulátor  
ak je STOP stlačený, A = 03H a nasadí príznak Z  
ak nie je stlačený, A = 40H a príznak NZ

Registre: mení PSW

#### Podprogram TPREAD

Punkcia: Načítava alebo kontroluje záznam súboru na magnetofónovej kazete v závislosti od želanej operácie a čísla súboru.

Adresa: EDE2H

Vstup: C1B0H - číslo hľadaného súboru  
C1B1H - kód operácie: 3FH načítavanie (MGLD)  
00H kontrola (MGEND)

Výstup : pamäťová oblasť C1B2H - C1BFH - (hlavička súboru)  
pri chybe sa nasadí príznak CY

Registre: mení DE,HL

#### Podprogram TPWRTE

Punkcia: Zapiše na mg. kazetu súbor podľa údajov na adresách C1B2H až C1BFH.

Adresa: EF60H

Vstup: C1B2H - číslo súboru (1 bajt)  
C1B3H - typ súboru (1 bajt)  
C1B4H - začiatočná adresa (2 bajty)  
C1B6H - dĺžka súboru (2 bajty)  
C1B8H - názov súboru (8 bajtov)

Registre: mení DE,HL



## Podprogram TRANSF

Punkcia: Prenáša časť pamäte z ROM-modulu do operačnej pamäte.

Adresa: EC00H

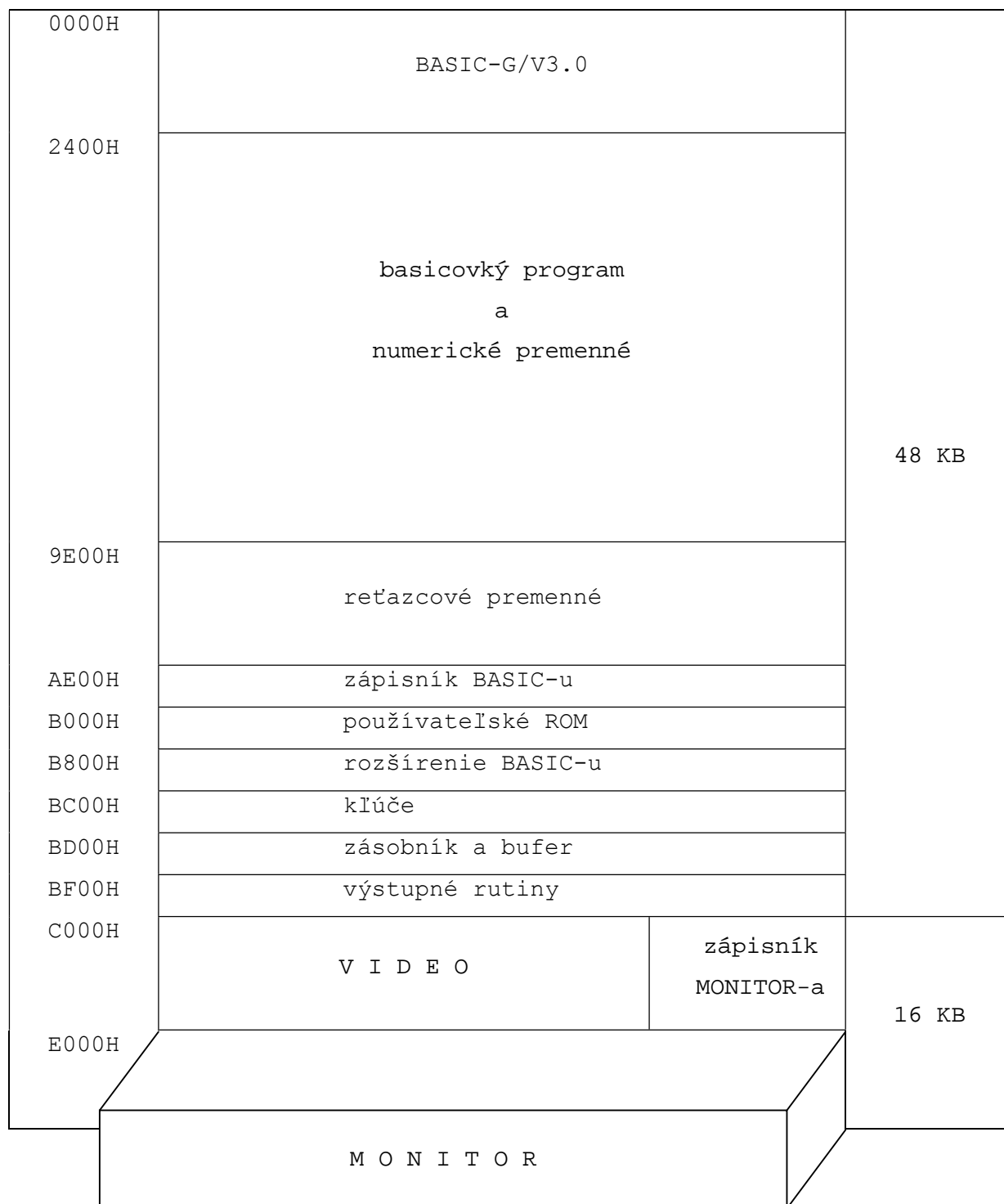
Vstup: 3 dvojбайtové slová uložené za volacou inštrukciou  
1.slovo - adresa v ROM-module, odkiaľ sa prenáša  
2.slovo - počet prenášaných bajtov **mínus 1**  
3.slovo - adresa v operačnej pamäti, kam sa prenáša

Registre: mení všetky

## 5.5. Významné premenné MONITOR-a

ADRESA	VÝZNAM
C03AH	kód farby
C03C-DH	adresa tabuľky znakov
C03E-FH	kurzor symbolovej grafiky
C074H	adresa textu na výpis do DR (pre DITXT2)
C078H	adresa vstupného bafra (pre ENTER)
C07CH	prenosová rýchlosť terminálu
C134H	kód prijatého klávesu
C137H	00H - kopírovanie DR FFH - vypnutie kopírovania (ako SHIFT + PTL)
C170-2H	kurzor súradnicovej grafiky
C178H	dĺžka cyklu autorepeat
C17A-BH	kurzor binárnej grafiky
C1B0H	číslo hľadaného záznamu na mgf. kazete
C1B1H	3FH - načítanie záznamu z mgf. kazety 00H - kontrola záznamu
C1B2H-C1BFH	hlavička záznamu na mgf. kazete :
C1B2H	číslo súboru
C1B3H	typ súboru : 3FH (?) - program v strojovom kóde 3EH (>) - program v BASIC-u 44H (D) - pole údajov pre BASIC
C1B4-5H	začiatočná adresa súboru v pamäti
C1B6-7H	dĺžka súboru
C1B8H-C1BFH	meno súboru
C1FAH	režim činnosti SUOG : A8H - negácie AFH - zhasnutie B0H - rozsvietenie
C1FEH	rýchlosť prenosu na kazetový magnetofón

# Mapa operačnej pamäte



# Príloha A

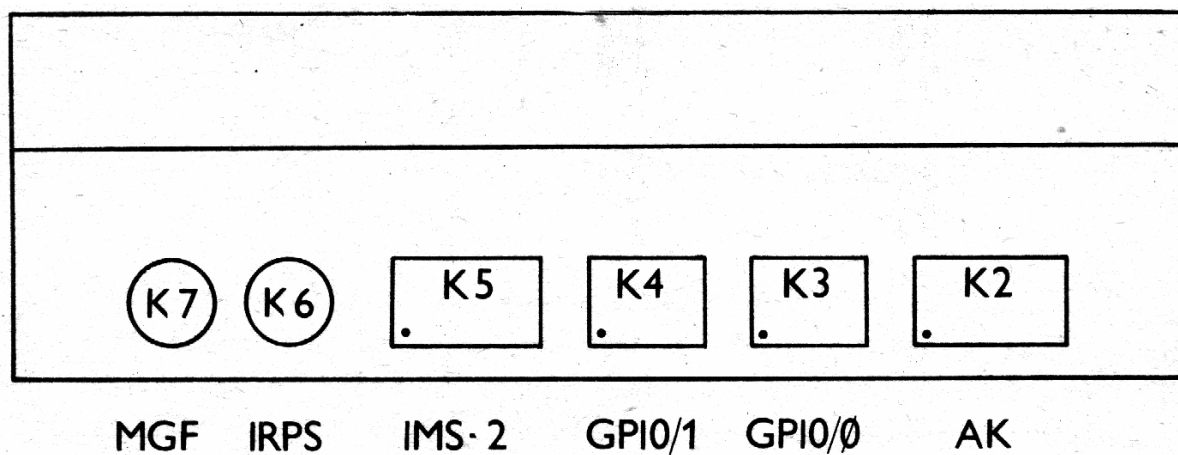
## A. OPIS MODULU INTERFEJSOV

Modul interfejsov rozširuje možnosti použitia PMD 85-3 v styku s okolím. Základné vstupno-výstupné zariadenia mikropočítača (klávesnica a zobrazovacia jednotka) sú ovládané priamo z modulu procesora. Pomocou modulu interfejsov je možné realizovať komunikáciu s rôznymi periférnymi zariadeniami paralelným aj sériovým spôsobom alebo používať časovač.

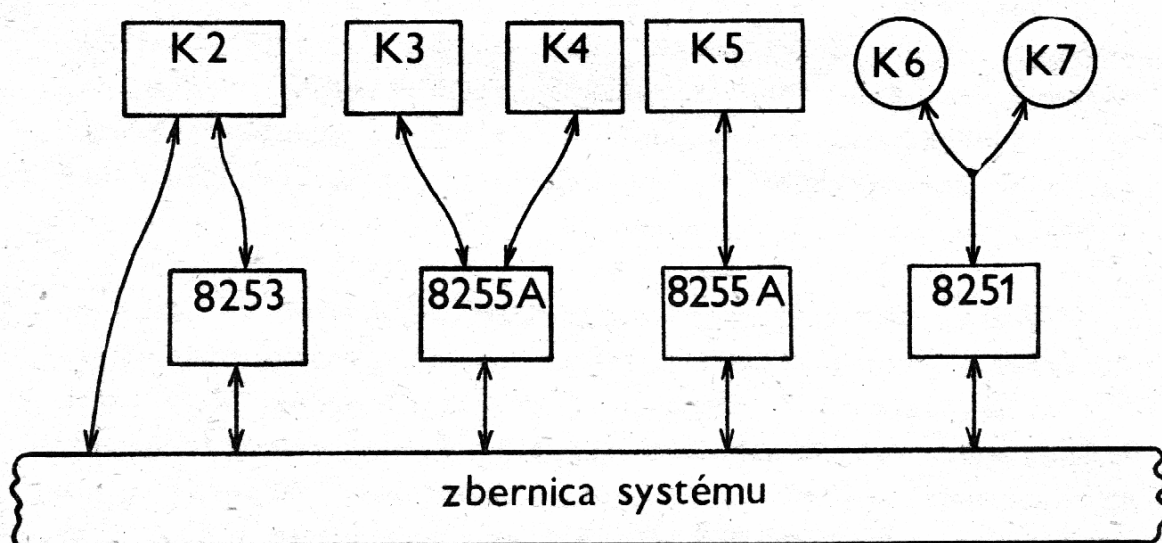
Všetky obvody na module interfejsov, cez ktoré možno komunikovať, sú označované ako kanály. Spojenie periférneho zariadenia s kanálom je cez tzv. interfejs, ktorý je definovaný konektorom, logickým významom jednotlivých signálov na konektore a ich elektrickými vlastnosťami. V tejto časti sú opísané základné možnosti použitia jednotlivých interfejsov, ich najdôležitejšie vlastnosti a rozloženie signálov na príslušných konektoroch. Spôsob ich programového ovládania cez jednotlivé kanály je v prílohe B.

Všetky interfejsy z modulu interfejsov sú vyvedené na konektoroch umiestnených v zadnej časti mikropočítača (obr. A.1). Na tomto obrázku je na konektoroch K2 až K5 v ľavom dolnom rohu bodkou označený pin č.1. Číslovanie pokračuje v dolnom rade nepárnymi číslami, horný rad sú párne čísla začínajúce sa pinom č.2 (umiestnený nad pinom č.1). Toto číslovanie nemusí byť zhodné s číslovaním pinov priamo na telesách konektorov.

Principiálna schéma modulu interfejsov je na obr. A.2.



Obr. A.1. Pohľad na zadnú časť PMD 85-3



Obr. A.2. Principiálna schéma modulu interfejsov

### A.1. Univerzálny paralelný interfejs (GPIO)

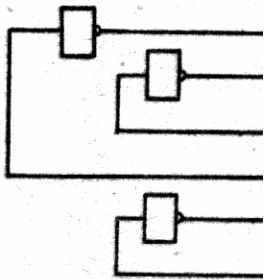
Oblasť jeho použitia je najmä v pripájaní štandardných periférií s riadeným paralelným prenosom údajov, ako napr. tlačiareň, snímač a dierovač diernej pásky, externá pamäť na pružnom disku ap. Možno ho však použiť aj na pripájanie neštandardných periférií.

Interfejs je realizovaný na báze programovateľného obvodu MHB8255A, ktorý určuje jeho vlastnosti. Keďže tento obvod umožňuje riadený paralelný prenos údajov dvoma nezávislými cestami (brána A a brána B), jeho výstupy sú rozdelené a pripojené na dva 20-pinové konektory typu FRB, označené K3 a K4. Rozmiestnenie signálov na konektoroch je na obr. A.3.

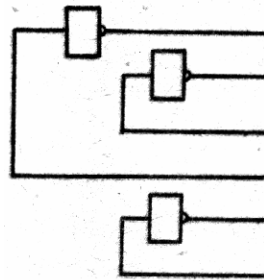
Údajové linky PA0 až PA7, resp. PB0 až PB7, sú vedené na konektory cez oddeľovacie obvody MH8286, pričom voľba smeru údajov cez ne sa ovláda privedením úrovne log.0 alebo log.1 na pin č.8 konektora. Ak je na pin č.8 privedená log.1 (resp. na pin nie je pripojené nič), tok údajov je smerom von. V prípade pripojenia log.0 (čo sa dá realizovať spojením pinu č.8 s pinom č.1) je tok údajov smerom dnu. Tok údajov je možné ovládať aj programovo, ak sa pin č.8 spojí s výstupom niektorého pinu brány C.

Výstupy brány C sú rozdelené na polovičku a pripojené na obidva konektory. V prípade riadeného prenosu majú funkciu potvrdzovacích signálov prenosu (handshake). Tieto signály nie sú výkonovo zosilnené, v prípade potreby ich možno invertovať cez obvody MH7405, ktorých vstupy a výstupy sú privedené na konektory. Invertory možno použiť aj na vytvorenie úrovni log.1 uzemnením ich vstupov na pin č.1.

Pokiaľ je pripojená disketová jednotka PMD 32 a podarí sa úspešne zaviesť systém z diskety, MONITOR inicializuje bránu A v móde 2 a bránu B ako výstupnú v móde 1. V opačnom prípade sú obe brány inicializované ako výstupné v móde 1.

Signál	Pin
GND	1
	2
	3
	4
	5
	6
	7
ovládanie v/v	8
PC7	9
PC6	10
PC5	11
PC4	12
PA1	13
PA0	14
PA3	15
PA2	16
PA5	17
PA4	18
PA7	19
PA6	20

Konektor K3 (GPIO/0)

Signál	Pin
GND	1
	2
	3
	4
	5
	6
	7
ovládanie v/v	8
PC3	9
PC2	10
PC1	11
PC0	12
PB1	13
PB0	14
PB3	15
PB2	16
PB5	17
PB4	18
PB7	19
PB6	20

Konektor K4 (GPIO/1)

Obr. A.3. Rozmiestnenie signálov na konektoroch

Signál	Pin	Brána
	1	
$\overline{\text{NRFD}}$	2	PC2
DAV	3	PC1
	4	
$\overline{\text{NDAC}}$	5	PC3
	6	
GND	7	GND
GND	8	GND
	9	
REN	10	PC6
SRQ	11	PB4
D3	12	PA3
EOI	13	PC4
D2	14	PA2
	15	
ATN	16	PC5
	17	
D1	18	PA1
	19	
IFC	20	PC7
D0	21	PA0
D7	22	PA7
D6	23	PA6
	24	
D5	25	PA5
D4	26	PA4
	27	
	28	
	29	
	30	

Obr. A.4. Rozmiestnenie signálov na konektore K5 (IMS-2)



## A.2. Paralelný interfejs IMS-2

Je to špecializovaný interfejs na prepájanie meracej a laboratórnej techniky, ktorý bol navrhnutý firmou Hewlett-Packard pod označením HP-IB (známy aj ako štandard IEEE 488).

Interfejs je realizovaný na báze programovateľného obvodu MHB8255A a niektorých ďalších obvodov, ktoré zabezpečujú výkonové prispôsobenie signálov. Je vyvedený na 30-pinovom konektore typu FRB označenom K5.

Všetky signály interfejsu sa ovládajú programovo. Programová podpora realizovaná v PMD 85-3 umožňuje prácu v režime "CONTROLLER". Viacvodičové správy sa prenášajú po údajových vodičoch D0 až D7 realizovaných cez bránu A, jednovodičové správy sa prenášajú linkami DAV, NRFD, NDAC, IFC, ATN, REN, SRQ a EOI realizovanými cez brány B a C. Rozmiestnenie signálov na konektore K5 je na obr. A.4.

Ak používateľ nemá záujem o tento interfejs, môže využiť obvod MHB8255A ako ďalšie paralelné vstupy alebo výstupy.

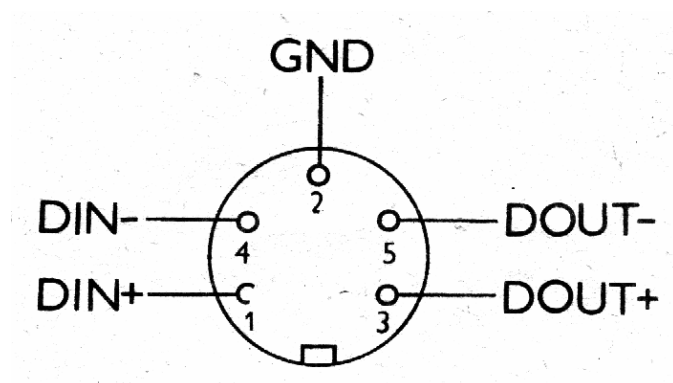
### A.3. Sériový interfejs IRPS

Slúži na pripojenie zariadení so sériovým prenosom údajov v plnoduplexnom režime, napr. terminál, iný počítač atď.

Interfejs je realizovaný univerzálnym programovateľným obvodom MHB8251 (USART), ktorý riadi aj interfejs pre kazetový magnetofón (pozri prílohu A.4). Z množiny signálov obvodu MHB8251, ktoré sú určené na vytvorenie interfejsu definovaného odporúčením CCITT V.24, sú v tomto interfejsi použité iba dva údajové: TxD a RxD. Tieto signály sa pomocou špeciálnych obvodov a oddeľovacích optočlenov transformujú na signály prúdovej slučky. Log.0 sa realizuje bezprúdovým stavom (resp.  $I < 4 \text{ mA}$ ), log.1 prúdom 20 mA. Obidve linky sú zapojené ako pasívne.

Rýchlosť prenosu údajov je daná frekvenciou hodinového signálu privedeného na vstupy TxC a RxC, ktorý je generovaný v obvode časovača (TIMER 1) a ktorý je inicializovaný MONITOR-om na hodnotu 1200 Hz (pozri prílohu A.5).

Interfejs IRPS je vyvedený na 5-kolíkový konektor označený K6, rozloženie signálov je na obr. A.5.



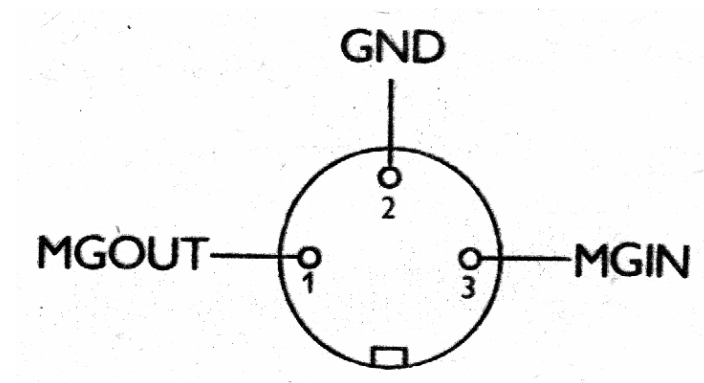
Obr. A.5. Rozloženie signálov na konektore K6

#### A.4. Interfejs pre kazetový magnetofón

Tento interfejs je určený na pripojenie kazetového magnetofónu, ktorý slúži pre PMD 85-3 ako archivačné zariadenie pre programy a údaje.

Interfejs je realizovaný obvodom MHB8251 (USART), ktorý riadi aj interfejs IRPS (pozri prílohu A.3). Vyvedený je na 5-kolíkový konektor označený K7, rozloženie signálov je na obr. A.6.

Signál z USART-u je upravený špeciálnym obvodom (tzv. magnetofónový modem), v ktorom je týmto signálom modulovaná frekvencia 1200 Hz a tento signál je privedený na vstup magnetofónu. V opačnom smere je signál z magnetofónu demodulovaný a upravený na vstup do USART-u.



A.6. Rozloženie signálov na konektore K7

#### A.5. Časovač

Tento obvod je určený na generovanie presných časových značiek, resp. ako počítadlo udalostí reprezentovaných elektrickými impulzmi.

V PMD 85-3 je ako časovač použitý programovateľný obvod typu MHB8253. V ňom sú umiestnené tri samostatné časovače, z ktorých prvý (TIMER 0) je úplne prístupný používateľovi cez aplikačný konektor K2, kde sú vyvedené jeho signály CLK0, GATE0 a OUT0 (obr. A.7). Druhý časovač (TIMER 1) je použitý na generovanie riadiacej frekvencie sériového prenosu dát cez obvod MHB8251 (pozri prílohu A.3) a po zapnutí PMD 85-3 je inicializovaný MONITOR-om na frekvenciu výstupného signálu (OUT1) 1200 Hz. Na vstup CLK1 je privedený signál systémových hodín  $\Phi 2$  (2000 kHz), signály GATE1 a OUT1 sú vyvedené na aplikačný konektor. V prípade potreby ho môže používateľ preprogramovať. Tretí časovač (TIMER 2) slúži ako hodiny reálneho času, na jeho vstup (CLK2) je privedený signál s periódou  $T = 1 \text{ s}$ .

#### A.6. Interfejs systémovej zbernice

Cez tento interfejs, ktorý je vyvedený na aplikačný konektor označený K2, je možné v náročnejších aplikáciách rozširovať možnosti PMD 85-3 pripájaním ďalších obvodov.

Interfejs obsahuje podmnožinu signálov systémovej zbernice PMD 85-3.

adresná zbernica	- dolných 8 bitov (AB0 až AB7)
údajová zbernica	- DB0 až DB7
riadiaca zbernica	- signály $\overline{\text{RESET}}$ , I/OR, I/OW, $\overline{\text{INT}}$ a $\overline{\Phi 2(\text{TTL})}$
napájacie napätia	- +5 V (max. 0,2 A), -5 V a +12 V

Všetky signály systémovej zbernice sú oddelené pomocou oddelovacích obvodov.

Rozmiestnenie signálov na konektore K2 je na obr. A.6.

Signál	Pin	Typ signálu
GND	1	napájanie
-5 V	2	napájanie
AB6	3	výstup
AB3	4	výstup
AB2	5	výstup
AB5	6	výstup
AB4	7	výstup
AB0	8	výstup
AB1	9	výstup
AB7	10	výstup
GATE1	11	vstup
OUT0	12	výstup
GATE0	13	vstup
OUT1	14	výstup
<u>INT</u>	15	vstup
CLK0	16	vstup
I/OR	17	výstup
DB5	18	v/v
<u>RESET</u>	19	výstup
DB6	20	v/v
$\Phi 2$ (TTL)	21	výstup
DB7	22	v/v
I/OW	23	výstup
DB3	24	v/v
DB1	25	v/v
DB2	26	v/v
DB0	27	v/v
DB4	28	v/v
+5 V	29	napájanie
+12 V	30	napájanie

Obr. A.6. Rozmiestnenie signálov na konektore K2



## Príloha B

### B. PROGRAMOVANIE VSTUPNO-VÝSTUPNÝCH KANÁLOV

Spolupráca PMD 85-3 s periférnymi zariadeniami, ktoré sú k nemu pripojené cez niektorý z interfejsov (pozri prílohu A), sa vykonáva prenosom údajov cez príslušný vstupno-výstupný (v/v) kanál. Kanály sú osadené štandardnými v/v obvodmi (MHB8251, MHB8253, MHB8255A), ktoré čiastočne alebo úplne riadia prenos údajov medzi nimi a perifériami cez príslušné interfejsy. Tieto obvody sú programovateľné, čo znamená, že je možné definovať a meniť ich vlastnosti. Programovanie sa vykoná zápisom jedného alebo viacerých riadiacich slov do riadiaceho registra v/v obvodu. Po jeho naprogramovaní je možné cez v/v kanál komunikovať s pripojenou perifériou (perifériami) čítaním z údajového registra alebo zápisom do údajového registra (registrov) alebo zistiť stav, v ktorom sa v/v obvod nachádza, čítaním stavového registra. Na označenie údajového registra v/v obvodu sa v niektorých prípadoch používa výraz brána (anglicky port).

Ovládanie v/v kanálu možno v mikropočítači PMD 85-3 robiť na dvoch úrovniach:

- na fyzickej úrovni, keď sa ovládanie v/v obvodu musí úplne zabezpečiť v danom programe (spravidla na úrovni strojového kódu mikroprocesora MHB8080A), pričom sa používajú fyzické adresy v/v obvodov.

- na logickej úrovni, keď sa ovládanie v/v obvodov vykonáva v časti interpretera jazyka BASIC-G (nazývanej exekutíva) na základe príkazov a funkcií jazyka BASIC-G. Namiesto fyzických adries v/v obvodov sa používajú symbolické čísla v/v kanálov.



### B.1. Fyzická úroveň

Strojový kód mikroprocesora MHB8080A má pre prácu s v/v obvodmi dva príkazy:

IN (kód DB) - čítanie údajov z v/v obvodu

OUT (kód D3) - zápis údajov do v/v obvodu

Inštrukcie sú dvoj bajtové, 1.bajt je inštrukčný kód a 2.bajt obsahuje fyzickú adresu v/v registra. Konkrétne adresy v/v registrov mikropočítača PMD 85-3 sú v tabuľke B.1.

obvod	v/v register	adresa		príslušný interfejs
		HEX	DEK	
MHB8251	údajový	1E	30	IRPS a magnetofón
	riadiaci, stavový	1F	31	
MHB8253	časovač 0	5C	92	
	časovač 1	5D	93	
	časovač 2	5E	94	
	riadiaci	5F	95	
MHB8255A	brána A	4C	76	GPIO/0 a GPIO/1
	brána B	4D	77	
	brána C	4E	78	
	riadiaci	4F	79	
MHB8255A	brána A	7C	124	IMS-2
	brána B	7D	125	
	brána C	7E	126	
	riadiaci	7F	127	

Tabuľka B.1 Fyzické adresy v/v registrov

Podpora práce v strojovom kóde je v základnom programovom vybavení MONITOR (príkazy MEM, SUB, JUMP), ako aj v interpreteri jazyka BASIC-G (príkazy CODE a USR).

K úrovni používania v/v inštrukcií v strojovom kóde možno logicky priradiť aj používanie funkcie jazyka BASIC-G INP a príkazu OUT, ktoré sú vlastne basicovské varianty príkazov strojového kódu IN a OUT.

#### Príklad:

Je potrebné zapísať riadiace slovo do obvodu MHB8255A, ktorý riadi interfejsy GPIO/0 a GPIO/1 tak, aby všetky tri brány pracovali ako vstupné v režime 0.

#### Riešenie:

Príslušné riadiace slovo má binárny tvar 10011011, čo je hexadecimálne 9B a dekadicky 155. Podprogram napísaný v strojovom kóde má nasledujúci tvar (vpravo je symbolický zápis):

3E 9B	MVI A,9B
D3 4F	OUT 4F
C9	RET

Adresa riadiaceho registra (4F) bola určená z tabuľky B.1, jej dekadický ekvivalent je 79.

Vzhľadom na to, čo sa vyššie uviedlo, tento podprogram je možné uložiť do operačnej pamäte mikropočítača a vykonať ho tromi spôsobmi:

1) podpora MONITOR-a (podprogram bude uložený v operačnej pamäti od adresy 8192 (2000H)):

```
SUB 2000 3E9BD34FC9
SUB 2005
JUMP 2000
```

2) podpora jazyka BASIC-G (príkazy sú zadané v priamom režime):

```
A$="3E9BD34FC9"
```

```
CODE A$
```

3) podpora jazyka BASIC-G (príkaz je uvedený ako segment programu):

```
200 ...
```

```
210 OUT 79,155
```

```
220 ...
```

## B.2. Logická úroveň

Na rozdiel od predchádzajúcej kapitoly jazyk BASIC-G má aj ďalšiu, efektívnejšiu možnosť pre prácu s v/v kanálmi. Túto činnosť podporujú príkazy a funkcie CONTROL, STATUS, OUTPUT, ENTER, resp. PRINT#, LIST# a INPUT#.

V syntaxi týchto príkazov sa zadáva vždy parameter k, ktorý označuje v/v kanál, cez ktorý sa bude komunikovať. Tento parameter je celé číslo, ktoré je symbolickým menom konkrétneho v/v kanálu (nie jeho fyzickou adresou). BASIC-G dovoľuje pracovať maximálne s 8 v/v kanálmi, ktoré sú označené číslami 0 až 7. Vzťah parametra k ku skutočným v/v obvodom je uvedený v tabuľke B.2.

k	obvod	príslušný interfejs
0		
1	MHB8251	IRPS a magnetofón
2		
3		
4	MHB8255A	
5		GPIO/0 a GPIO/1
6		
7	MHB8255A	IMS-2

Tabuľka B.2 logické čísla v/v kanálov

Z tabuľky B.2. vidieť, že BASIC-G v PMD 85-3 štandardne podporuje iba tri kanály (1, 4 a 7). Časovač takúto podporu nemá, možno ho ovládať iba spôsobom opísaným v kapitole B.1.

Keďže jednotlivé v/v obvody majú vnútri spravidla viacej v/v registrov, je potrebné v týchto príkazoch špecifikovať aj konkrétny v/v register, ktorý sa zadáva v príkazoch ako parameter r. Význam tohoto parametra je závislý ako od príkazu, tak aj od konkrétneho kanálu.

Pre príkazy CONTROL a STATUS je parameter r identický s vnútornou adresou v/v registra v obvode, ktorým je kanál realizovaný (tabuľka B.3).

Pre skupinu príkazov ENTER a OUTPUT predstavuje parameter r referenciu na obslužnú rutinu, ktorá vykonáva prenos údajov cez konkrétny v/v register. Význam parametra r v tomto prípade je uvedený v tabuľke B.4. Poznámka "čs" znamená, že príslušná rutina podporuje tlačiareň so slovenskou alebo českou abecedou, "invert" znamená, že údaj vysielaný na kanál je invertovaný.

k	r	príslušný v/v register
1	0	údajový register
	1	riadiaci a stavový register
4,7	0	brána A
	1	brána B
	2	brána C
	3	riadiaci register

Tabuľka B.3

ENTER			OUTPUT	
k	r	v/v brána	r	v/v brána
1		nezadáva sa		nezadáva sa
4	0	A, režim 0	0	A, režim 0
	1	B, režim 0 *	1	B, režim 0
	2	C, režim 0 *	2	B, režim 1
	3	A, režim 2	3	B, režim 1
	4	B, režim 1 *	4	A, režim 2
	5	A, režim 2	5	B, režim 1, invert
			6	B, režim 1, čs
			7	B, režim 1, čs, invert
7	0	adresa zariadenia	0	adresa zariadenia
	1	na zbernici IMS-2	1	na zbernici IMS-2
	3	vo funkcii TALKER	2	vo funkcii LISTENER
	4		4	
	...		...	

\* možno použiť po preprogramovaní obvodu MHB8255A (kanál 4)

Tabuľka B.4

# Príloha C

C. TABUĽKA VNÚTORNÉHO KÓDU PMD 85-3 (1. časť)

b <sub>7</sub>		0	0	0	0	0	0	0	0
b <sub>6</sub>		0	0	0	0	1	1	1	1
b <sub>5</sub>		0	0	1	1	0	0	1	1
b <sub>4</sub>		0	1	0	1	0	1	0	1
b <sub>3</sub> b <sub>2</sub> b <sub>1</sub> b <sub>0</sub>		0	1	2	3	4	5	6	7
0 0 0 0	0	NUL	DLE	SP	0	@	P	\	p
0 0 0 1	1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC2	"	2	B	R	b	r
0 0 1 1	3	ETX	DC3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC4	\$	4	D	T	d	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ETB	'	7	G	W	g	w
1 0 0 0	8	BS	CAN	(	8	H	X	h	x
1 0 0 1	9	HT	ET	)	9	I	Y	i	y
1 0 1 0	A	LF	SUB	*	:	J	Z	j	z
1 0 1 1	B	VT	ESC	+	;	K	[	k	{
1 1 0 0	C	FF	FS	,	<	L	\	l	
1 1 0 1	D	CR	GS	-	=	M	]	m	}
1 1 1 0	E	SO	RS	.	>	N	^	n	~
1 1 1 1	F	SI	US	/	?	O	_	o	■

TABULKA VNÚTORNÉHO KÓDU PMD 85-3 (2. časť)

b <sub>7</sub>	1	1	1	1	1	1	1	1	1
b <sub>6</sub>	0	0	0	0	1	1	1	1	1
b <sub>5</sub>	0	0	1	1	0	0	1	1	1
b <sub>4</sub>	0	1	0	1	0	1	0	1	1
b <sub>3</sub> b <sub>2</sub> b <sub>1</sub> b <sub>0</sub>	8	9	A	B	C	D	E	F	
0 0 0 0	0				Š	š	π	ō	
0 0 0 1	1				á	ä	Á	Ǻ	
0 0 1 0	2				ä	ř	Ř	Ř	
0 0 1 1	3				č	š	Č	Š	
0 1 0 0	4				ď	t	Ď	Ť	
0 1 0 1	5				ě	ú	Ě	Ú	
0 1 1 0	6				ř	β	Ř	Ť	
0 1 1 1	7				■	é	■	É	
1 0 0 0	8				ü	à	Ü	À	
1 0 0 1	9				í	ý	Í	Ý	
1 0 1 0	A				û	ž	Û	Ž	
1 0 1 1	B				í	■	Ĺ	■	
1 1 0 0	C				ř	■	Ĺ	■	
1 1 0 1	D				ö	■	Ö	■	
1 1 1 0	E				ň	■	Ň	■	
1 1 1 1	F				ó	■	Ó	■	

## Príloha D

### D. TABUĽKA INŠTRUKCIÍ ASEMLERA 8080

V nasledujúcej tabuľke sú v abecednom poradí uvedené symbolické názvy všetkých možných inštrukcií strojového kódu mikroprocesora MHB8080A. Každému symbolickému menu inštrukcie prislúcha kód, ktorý je uvedený v hexadecimálnom zápise. Ako operandy inštrukcií sú použité symboly s nasledujúcim významom:

d8	- 1-bajtový priamy operand
d16	- 2-bajtový priamy operand
adr	- 2-bajtový údaj (adresa operačnej pamäte)
port	- 1-bajtový údaj (adresa v/v registra)
A	- akumulátor
B	- register B alebo registrový pár BC
C	- register C
D	- register D alebo registrový pár DE
E	- register E
H	- register H alebo registrový pár HL
L	- register L
M	- symbol pre pamäťovú bunku (adresa je v HL)
PSW	- stavové slovo procesora
SP	- ukazovateľ zásobníka



ACI	d8	CE	CNZ	adr	C4	JMP	adr	C3
ADC	A	8F	CP	adr	F4	JNC	adr	D2
ADC	B	88	CPE	adr	EC	JNZ	adr	C2
ADC	C	89	CPI	d8	FE	JP	adr	F2
ADC	D	8A	CPO	adr	E4	JPE	adr	EA
ADC	E	8B	CZ	adr	CC	JPO	adr	E2
ADC	H	8C	DAA		27	JZ	adr	CA
ADC	L	8D	DAD	B	09	LDA	adr	3A
ADC	M	8E	DAD	D	19	LDAX	B	0A
ADD	A	87	DAD	H	29	LDAX	D	1A
ADD	B	80	DAD	SP	39	LHLD	adr	2A
ADD	C	81	DCR	A	3D	LXI	B,d16	01
ADD	D	82	DCR	B	05	LXI	D,d16	11
ADD	E	83	DCR	C	0D	LXI	H,d16	21
ADD	H	84	DCR	D	15	LXI	SP,d16	31
ADD	L	85	DCR	E	1D	MOV	A,A	7F
ADD	M	86	DCR	H	25	MOV	A,B	78
ADI	d8	C6	DCR	L	2D	MOV	A,C	79
ANA	A	A7	DCR	M	35	MOV	A,D	7A
ANA	B	A0	DCX	B	0B	MOV	A,E	7B
ANA	C	A1	DCX	D	1B	MOV	A,H	7C
ANA	D	A2	DCX	H	2B	MOV	A,L	7D
ANA	E	A3	DCX	SP	3B	MOV	A,M	7E
ANA	H	A4	DI		F3	MOV	B,A	47
ANA	L	A5	EI		FB	MOV	B,B	40
ANA	M	A6	HLT		76	MOV	B,C	41
ANI	d8	E6	IN	port	DB	MOV	B,D	42
CALL	adr	CD	INR	A	3C	MOV	B,E	43
CC	adr	DC	INR	B	04	MOV	B,H	44
CM	adr	FC	INR	C	0C	MOV	B,L	45
CMA		2F	INR	D	14	MOV	B,M	46
CMC		3F	INR	E	1C	MOV	C,A	4F
CMP	A	BF	INR	H	24	MOV	C,B	48
CMP	B	B8	INR	L	2C	MOV	C,C	49
CMP	C	B9	INR	M	34	MOV	C,D	4A
CMP	D	BA	INX	B	03	MOV	C,E	4B
CMP	E	BB	INX	D	13	MOV	C,H	4C
CMP	H	BC	INX	H	23	MOV	C,L	4D
CMP	L	BD	INX	SP	33	MOV	C,M	4E
CMP	M	BE	JC	adr	DA	MOV	D,A	57
CNC	adr	D4	JM	adr	FA	MOV	D,B	50

MOV D,C	51	MVI E,d8	1E	RST 5	EF
MOV D,D	52	MVI H,d8	26	RST 6	F7
MOV D,E	53	MVI L,d8	2E	RST 7	FF
MOV D,H	54	MVI M,d8	36	RZ	C8
MOV D,L	55	NOP	00	SBB A	9F
MOV D,M	56	ORA A	B7	SBB B	98
MOV E,A	5F	ORA B	B0	SBB C	99
MOV E,B	58	ORA C	B1	SBB D	9A
MOV E,C	59	ORA D	B2	SBB E	9B
MOV E,D	5A	ORA E	B3	SBB H	9C
MOV E,E	5B	ORA H	B4	SBB L	9D
MOV E,H	5C	ORA L	B5	SBB M	9E
MOV E,L	5D	ORA M	B6	SBI d8	DE
MOV E,M	5E	ORI d8	F6	SHLD adr	22
MOV H,A	67	OUT port	D3	SPHL	F9
MOV H,B	60	PCHL	E9	STA adr	32
MOV H,C	61	POP B	C1	STAX B	02
MOV H,D	62	POP D	D1	STAX D	12
MOV H,E	63	POP H	E1	STC	37
MOV H,H	64	POP PSW	F1	SUB A	97
MOV H,L	65	PUSH B	C5	SUB B	90
MOV H,M	66	PUSH D	D5	SUB C	91
MOV L,A	6F	PUSH H	E5	SUB D	92
MOV L,B	68	PUSH PSW	F5	SUB E	93
MOV L,C	69	RAL	17	SUB H	94
MOV L,D	6A	RAR	1F	SUB L	95
MOV L,E	6B	RC	D8	SUB M	96
MOV L,H	6C	RET	C9	SUI d8	D6
MOV L,L	6D	RLC	07	XCHG	EB
MOV L,M	6E	RM	F8	XRA A	AF
MOV M,A	77	RNC	D0	XRA B	A8
MOV M,B	70	RNZ	C0	XRA C	A9
MOV M,C	71	RP	F0	XRA D	AA
MOV M,D	72	RPE	E8	XRA E	AB
MOV M,E	73	RPO	E0	XRA H	AC
MOV M,H	74	RRC	0F	XRA L	AD
MOV M,L	75	RST 0	C7	XRA M	AE
MVI A,d8	3E	RST 1	CF	XRI d8	EE
MVI B,d8	06	RST 2	D7	XTHL	E3
MVI C,d8	0E	RST 3	DF		
MVI D,d8	16	RST 4	E7		



# PRÍLOHA E

0 1 2 3 4 5 6 7 8 9 A B C D E F

AE 00	KURZOR PRE PRINT		ČÍSLO VYKONÁVA- NEHO RIADKU	BUFER											
AE 10															
AE 20															
AE 30															
AE 40															
AE 50	PRACOV- NÝ BAJT	REAL = 0 STRING = 1	RIADOK S DATA?	PRACOVNÉ SLOVO	PRACOVNÉ SLOVO	PRVÝ VOLNÝ BAJT PRE STRING	PRACOVNÉ SLOVO	DEG = 1 RAD = 0	ČÍSLO CHYBY	NÁVESTIE CHYBNÉHO RIADKU	ADR. RIADKU ON ERR				
AE 60	NÁVESTIE RIADKU ON ERR	DĹŽKA, ZNAK REŤAZCOV + ADRESA			NÁVESTIE RIADKU PRE CONT	ADR. PRIKAZU PRE CONT	ADR. KONIEC PROGR. ZACÍATOK PREMENNÝCH	KONIEC PREMENNÝCH	NÁVESTIE RIADKU DATA	PREM = 9 FOR = 64 DEF = 80					
AE 70	PRACOV- NÝ BAJT	ADRESA PRIKAZU V RIADKU	PRACOVNÉ SLOVO	NÁVESTIE RIADKU PRE CONT	ADR. PRIKAZU PRE CONT	ADR. KONIEC PROGR. ZACÍATOK PREMENNÝCH	KONIEC PREMENNÝCH	NÁVESTIE RIADKU DATA	PREM = 9 FOR = 64 DEF = 80						
AE 80	SMERNÍK NA DATA	VÝSLEDOK VÝRAZU	EXPONENT	BUFER PRE OUT, HEX\$ DO/AFF											
AE 90															

ZÁPISNÍK BASIC V.3.0



# PRÍLOHA F

0 1 2 3 4 5 6 7 8 9 A B C D E F

ZAČIATOK BUF- RA D. R.	ADRESA D. R. NA OBRAZOVKE	KONIEC BUFRA D. R.	POZÍCIA KUR- ZORU V BUFRI	1. VYPISOVANÝ ZNAK Z BUFRA	KÓD FARBY	POZÍCIA BEEP	TABUĽKA ASCII	ADRESA KURZO- RA NA OBRAZOVKE
ADRESA VRCHO- LU ZÁSOBNÍKA	NASTAVENIE BUFRA PRI ANALÝZE	ADRESA TEXTU PRE VÝPIS DO D. R.	ADRESA NÁ- VRATU PRE ENTER	ADRESA TEXTU PO EOL	SP-PRE NÁVRAT Z ENTER	RÝCHLOSŤ PRENOSU 8251	ADRESA TABUĽKY DIREKTÍV MONITORA	
VOĽNÝ UKAZOVATEĽ	VEĽKÉ ZNAKY	DIAKRITIKA	MALÉ ZNAKY	VOĽNÝ UKAZOVATEĽ	VOĽNÝ UKAZOVATEĽ	MALÉ PÍSMENÁ + DIAKRITIKA	VEĽKÉ PÍSMENÁ + DIAKRITIKA	
4 BAJTY PRE PRÁCU S MGF	POČET ROLOVA- NÝCH TV RIADKOV	VÝŠKA RIADKU V T. V.	POLOHA KURZORA PO ROLOVANÍ	ZAČIATOK ROLOVANIA	POČET TV RIAD- KOV	POČET ZOBRA. ZN. V RIADKU	PRACOV- NÝ BYTE	POČET ZNAKOV V RIADKU
DATA PRE BEEP	TABUĽKA DIREKTÍV	NAČITA- NÝ ZNAK Z KLÁVES	ZNAK V HLAV V ZÁZNAM.	ADRESA STACKU A ULOŽENIE	ADRESA C-D BUFRA	DĹŽKA BUFRA	POMOCNÁ PRE- MENNÁ PRE ROLOVANIE	
PREMENNÁ X1 PRE INTERPO- LÁTOR	00-FF Y <sub>1</sub>	Y <sub>2</sub>	PRACOVNÉ SLOVO	PREMENNÁ PRE VSTUP ZNAKU	ADRESA KURZORA NA OBRAZOV.	ZARÁŽKA PRE KEYS	OSADENIE RAM KLÚČMI	
ČÍSLO ZÁZNAMU	ČÍSLO ZÁZNAMU	ODKAZ - KAM	KOŤKA	KOMENTÁR				
TYP ZÁZNAMU	TYP ZÁZNAMU	VÝSTUP WINDOW	PLOT PRE WINDOW	KÓD FARBY PRE WINDOW	KÓD FARBY PRE WINDOW	REŽIM KRESL. BODU	ADR. BODU PRE WINDOW	FREKVENCIA PRE MGF

Č030

Č070

Č0B0

Č0F0

Č130

Č170

Č1B0

Č1F0

ZÁPISNÍK OS V.3.0.

ZÁPISNÍK OS V.3.0.



# Príloha G

## G. CHYBOVÉ HLÁSENIA INTERPRETERA BASIC-G

Ak pri interpretácii príkazu jazyka BASIC-G dôjde k chybe, vypíše sa príslušné chybové hlásenie do dialógového riadka. Tento výpis je závislý od toho, v akom režime bol riadok interpretovaný.

1) výpis v priamom režime:

```
+ + + xxxxxxxxxxx + + +
```

2) výpis v programovom režime:

```
+ + + xxxxxxxxxxx at line yy + + +
```

Tieto výpisy sa modifikujú od prípadu k prípadu konkrétnym textom chybového hlásenia, ktorý má jednotnú dĺžku 10 znakov a je umiestnený na pozíciách xxxxxxxxxxx. Text chybového hlásenia je anglický. Na pozícii označenej yy sa v prípade programového režimu zobrazí číslo riadka v basicovskom programe, v ktorom došlo k chybe.

V prípade programového režimu sa okrem toho odpamätajú dva údaje:

- kód chyby (1-bajt)                      na adresu        38    (0026H)
- číslo riadka (2-bajty)                na adresu    44637    (AE5DH)

Tieto údaje možno použiť pri individuálnom ošetrovaní chýb v špeciálnych aplikáciách, keď sa využívajú možnosti príkazu ON ERR (pozri kapitolu 4).



Kód	Hlásenie	Význam
1	Subscr.rng	hodnota indexu poľa je mimo hraníc
2	Arr.alloc.	nesprávne alebo opakované dimenzovanie
3	Fnc.param.	chybný argument funkcie
4	Only in pg	príkaz možno použiť len v prog. režime
5	No for stm	chyba v cykle FOR
6	Data exhau	chýbajú dáta pre READ
7	Pg too big	preplnenie pamäťového priestoru
8	Overflow	prekročenie rozsahu čísiel
9	Syntax err	chybne zadaný príkaz
10	Return err	RETURN bez predchádzajúceho GOSUB
11	Numb.nonex	neexistujúce číslo riadka
12	Dv by zero	delenie nulou
13	Can't cont	CONT použitý po chybe alebo zmene
14	Strng long	reťazec je dlhší ako 256 znakov
15	No str.spc	naplnenie oblasti pre reťazce
16	Str.algrth	reťazcový výraz je dlhý alebo zložitý
17	Type conv.	text namiesto čísla alebo naopak
18	File small	chyba pri DLOAD alebo LOAD END
19	Input err	nezodpovedajúce hodnoty pre INPUT
20	Field lost	zbytočné údaje pre INPUT
21	File bound	číslo súboru nie je 0 - 99
22	Stop	zastavenie programu
23	File error	chyba pri nahrávaní

## Príloha H

```
100 REM *****
110 REM *** Program KRESLIČ ***
120 REM *****
130 DE=40:DIM H(DE),V(DE):POKE'C137,255:PEN3
140 REM * riadenie *****
150 ON ERR GOTO820:REM * ošetrenie chyby *
160 GOSUB 300:REM * úvodný výpis *
170 GOSUB 390:REM * zadanie funkcie *
180 GOSUB 440:REM * zadanie hraníc *
190 GOSUB 490:REM * výpočet hodnôt *
200 GOSUB 580:REM * vykreslenie *
210 REM *****
220 PEN1: DISP" Pokračovať ? (A/N) "
230 INPUT Z$: IF Z$<>"A" THEN 270
240 DISP" Zmeniť X alebo F(X) ? (X/F) "
250 INPUT Z$: IF Z$="F" THEN 170
260 GOTO 180
270 PEN8:POKE 'C137,00
280 END :REM * koniec programu*
290 REM *** úvodný výpis ***
300 GCLEAR:SCALE0,100,0,100:PEN8
310 MOVE0,0:PLOT0,95;100,95;100,0;0,0
320 MOVE25,80: LABEL3,3;"Kreslič"
330 MOVE24.7,79.5:LABEL3,3;"Kreslič"
340 PRINTAT9,3 "Jednoduchý program na vykreslenie grafu"
350 PRINTAT11,3"funkcie jednej premennej s možnosťou"
360 PRINTAT13,3"voľby hraníc vykreslenia alebo zmeny"
370 PRINTAT15,3"vykresľovanej funkcie."
380 RETURN
390 REM *** zadanie funkcie ***
400 PEN1:DISP" Zadanie funkcie F(X)=":INPUT A$
410 PRINTAT 19,6;INK(5);"F(X)=";A$;:PRINT INK(0)
420 DEF FNC F(X)=VAL(A$)
430 RETURN
440 REM *** zadanie hraníc ***
450 PEN 2:DISP" Zadanie hraníc X-min, X-max:"
460 INPUT XIN,XAX:PEN0
470 PRINT AT23,6;INK(6);"X min=";XIN;" X max=";XAX
480 RETURN
490 REM *** výpočet hodnôt ***
500 XP=ABS(XAX-XIN)/DE
510 FOR CNT=0 TO DE
520 H(CNT)=XIN+CNT*XP:F=H(CNT):V(CNT)=FNC F(F)
530 PEN3
540 DISP" Výpočet č.";CNT;" F(";F;")=";FNCF(F)
550 PEN0
560 NEXT:BEEP
570 RETURN
```

```

580 REM *** vykreslenie ***
590 XN=9E20:XM=9E-20:YN=9E20:YM=9E-20
600 FOR CNT=0 TO DE
610 IF H(CNT)<XN THEN XN=H(CNT)
620 IF H(CNT)>XM THEN XM=H(CNT)
630 IF V(CNT)>YM THEN YM=V(CNT)
640 IF V(CNT)<YN THEN YN=V(CNT)
650 NEXT:BEEP:GCLEAR
660 DX=ABS(XM-XN)/11:DY=ABS(YM-YN)/11
670 XD=XN-DX:XH=XM+DX:YD=YN-DY:YH=YM+DY
680 SCALE XD,XH,YD,YH:OX=XN:OY=YN
690 IF SGN(YN)<>SGN(YM) THEN OY=0
700 IF SGN(XN)<>SGN(XM) THEN OX=0
710 IF SGN(YM)=-1 THEN OY=YM
720 IF SGN(XM)=-1 THEN OX=XM
730 AXES OX,OY:MOVE H(0),V(0)
740 FOR CNT=1 TO DE: PLOT H(CNT),V(CNT): NEXT
750 MOVE OX,OY+.05:LABEL 1,1;OX
760 MOVE OX,YN-DY*.5:LABEL1,1;YN
770 MOVE XM-DX,OY+.05:LABEL1,1;XM
780 MOVE OX,YM-DY:LABEL1,1;YM
790 PEN8:SCALE0,255,0,242:MOVE0,0:FILL255,242; 1
800 PEN8
810 RETURN
820 PEN1:DISP"          CHYBA!  Nové zadanie! ":BEEP
830 BEEP:PAUSE10:GOTO150

```

```

;
;
;
;
;   UMIESTNENIE PROGRAMU Z MG PÁSKY
;   DO LUBOVOĽNEJ OBLASTI PAMÄTI
;   V PMD 85-3
;
;
;   NÁVOD: * ODŠTARTUJ PROGRAM
;           * ZADAJ ADRESU
;           * ZAPNI   MG
;
;
0132          ORG   5000H
              *C5000H
              KKK   EQU   4AH                Dolný bajt adr.
;
;
5000 21 003F  ZAC   LXI   H,3F00H           Číslo a typ
5003 22 B0C1          SHLD  0C1B0H         ulož na C1B0,1
5006 3E 21          MVI   A,21H           inštrukciu LXI H
5008 32 F0C0          STA   0C0F0H         ulož na C0F0
500B CD 1C50          CALL ENTER          nová adr. do 'HL'
500E 22 F1C0          SHLD  0C0F1H         ulož na C0F1
5011 3E C9          MVI   A,0C9H         inštrukciu RET
5013 32 F3C0          STA   0C0F3H         ulož na C0F3
5016 CD ADEB          CALL 0EBADH         LOAD
5019 C3 40EC          JMP   0EC40H        Koniec
501C E3          ENTER XTHL              Prevezme novú
501D 22 4450          SHLD  ODLOZ          adresu v hexkóde
5020 CD EEEB          CALL 0EBEEH         z klávesnice
5023 2A 4450          LHLD  ODLOZ         a uloží ju do
5026 E3          XTHL              'HL'
5027 2A 78C0          LHLD  0C078H
502A 11 4650          LXI   D,ODKIAL
502D 7E          DDD   MOV   A,M
502E 12          STAX  D
502F 36 20          MVI   M,20H
5031 23          INX   H
5032 13          INX   D
5033 7B          MOV   A,E
5034 FE 4A          CPI   KKK
5036 C2 2D50          JNZ   DDD
5039 21 4650          LXI   H,ODKIAL      Ukaz. bufer
503C CD 09E1          CALL 0E109H        preved' na hexkód
503F DA 1C50          JC   ENTER        Ak chyba : znovu
5042 EB          XCHG          ulož do 'HL'
5043 C9          RET          Návrat
5044 0000          ODLOZ DW   00H
5046 2020          ODKIAL DW  2020H
5048 2020          DW   2020H
504A 0D 0A          KON   DB   0DH,0AH

```

Na prevode tejto príručky do elektronickej podoby sa podieľali:

Scan:	Jan Křupka
OCR a export do PDF:	Martin Bórik
Sadzba, korekcie a opravy:	Roman Bórik

Text uvedený **modrým písmom** sa líši oproti originálu. Jedná sa buď o opravy uvedené v chybovníku prikladanom k pôvodnej príručke, alebo sú to opravené nepresnosti, či doplnené chýbajúce drobnosti.

---

Názov: PMD 85-3

Táto príručka bola vytvorená v Ústave výpočtovej techniky vysokých škôl v Bratislave autorským kolektívom: Anatol Filip, prom. Mat., Ing. Anton Hodál, RNDr. Gorazd Kmeť, Ing. Jozef Valentovič a Tibor Závodský, pod vedením Ing. Jána Petríka v spolupráci s pracovníkmi TESLA Bratislava, k.p. RNDr. Lorantom Šándorom a Ing. Viktorom Valentínym.

Text príručky bol napísaný a editovaný na textovom editore TEXT 01.

Vydanie:	prvé, rok 1988
Vydal:	TESLA Bratislava, k.p.
Tlač:	Tlačiarne SNP, n.p. Martin, závod Liptovský Mikuláš
Graf. úprava:	Terézia Hojzáková
Náklad:	25 000 ks

© TESLA Bratislava, k.p.