

ŠTÁTNY MAJETOK ZÁVADKA n./HRONOM, š. p.

mat' o

OSOBNÝ MIKROPOČÍTAČ

POPIS A NÁVOD NA OBSLUHU

O B S A H

Strana:

1. ÚVOD	3
2. POPIS MIKROPOČÍTAČA	4
3. PRIPOJITEĽNÉ ZARIADENIA	7
4. UVEDENIE DO PREVÁDZKY	7
5. OBSLUHA MIKROPOČÍTAČA	9
6. MONITOR	17
7. PROGRAMY MODULU MONITOR	25
8. PROGRAMOVANIE V JAZYKU BASIC - G	33
9. KALKULÁTOROVÝ REŽIM	37
10. PROGRAMOVÝ REŽIM	39
11. GRAFICKÝ REŽIM	65
12. MODIFIKOVANIE SYSTÉMU	80
13. CHYBOVÉ HLÄSENTIA	80
14. ZOZNAM PRÍKAZOV A FUNKCIÍ BASIC - G	81
15. ÚDRŽBA MIKROPOČÍTAČA	83
16. TECHNICKÉ ÚDAJE	84
17. ZÁVER	84
18. PRÍLOHY / SCHÉMY ZAPOJENIA /	

1. ÚVOD

Osobný mikropočítač MAŤO je určený pre uľahčenie práce, vyplnenie voľných chvíľ všetkým záujemcom o vniknutie do tajov počítačovej techniky. Predovšetkým je určený mládeži, ktorá sa v školách vyučuje základom výpočtovej techniky, a doma si chce precvičovať v škole prebraté učivo.

Dobre poslúži rôznym záujmovým krúžkom v staniciach mladých technikov, domoch pionierov a všade tam, kde širšiemu používaniu výpočtovej techniky doteraz bránili cenové bariéry.

mikropočítač MAŤO sa svojou architektúrou a výkonnosťou zaraďuje medzi počítače s grafickým spracovaním informácie. Pri jeho konštrukcii bolo sledované zabezpečenie plnej kompatibility na najrozšírenejší československý mikropočítač PMD-85, ktorý je na maloobchodnom trhu nedostupný, ale v hojnej mieri používaný u rôznych organizácií a hlavne v školách SSR /v školách ČSR je používaný počítač DIDAKTIK ALFA, s ktorým je MAŤO v princípe kompatibilný/.

Kompatibilita počítača je zaručená na úrovni operačného systému a vyššieho programovacieho jazyka BASIC-G. Z dôvodov jednoduchosti technického riešenia /a z toho vyplývajúcej ceny mikropočítača/ bol realizovaný u počítača MAŤO iným spôsobom styk s magnetofónom pre záznam dát a programov ako u počítača PMD-85. Z tohto dôvodu nie je priama prenositeľnosť programov nahratých protokolom PMD-85 na počítač MAŤO a naopak. Pomocou vhodného konverzného programu je možné odstrániť aj tento nedostatok. Ináč programy v jazyku BASIC-G ako aj programy v assembliere 8080 napísané pre PMD-85 budú správne pracovať aj na počítači MAŤO

Treba upozorniť, že kódy kláves s významom K0 - K11 sú ponechané úmyselne také, ako sú u modelu PMD-85.1 - prvá sada teda D0 - DA hexa a nie ako u modelu PMD-85.2 /od 80H/. Z toho dôvodu, že väčšina hier vznikla pre prvý model a predpokladáme, že začiatočník bude mať záujem práve o hry, a kódy od 80 hexa sú v počítači MAŤO využívané pre grafické znaky.

2. POPIS MIKROPOČÍTAČA

2.1 Všeobecný popis

Mikropočítač je vybudovaný na základe mikroprocesora MHB 8080A a jeho podporných obvodov s výstupom na bežný čiernobielý televízny prijímač nalaďený na 9. kanál pásma VHF. Komunikácia s mikropočítačom je orientovaná na tento televízny prijímač a vstavanú alfanumericko-grafickú klávesnicu.

Celkovo je mikropočítač riešený ako kompaktný celok s klávesnicou a sieťovým napájacím zdrojom. Príslušenstvo tvorí prepojovacia šnúra na prepojenie televízneho prijímača s vlastným počítačom. Konštrukčne pozostáva z troch samostatných častí:

2.2 Mikropočítač s videoprocesorom

Je na samostatnej doske plošného spoja a tvorí základ celého zariadenia. Prvá časť obsahuje klasický mikropočítač s mikroprocesorom MHB 8080A s jeho podpornými obvodmi, interfejsom pre styk s klávesnicou, aplikačným konektorm systémovej zbernice, pamäťou typu ROM, kde je umiestnený operačný systém, operačnou pamäťou typu RWM v rozsahu 48 kB a modulátorm TV-signálu.

Druhú časť tvorí obrazový procesor umožňujúci nepretržité zobrazovanie tzv. videostránky na televíznej obrazovke. Rozhranie medzi týmito časťami tvorí RAM v rozsahu 48 kB, ktorej spomínaná videostránka je spoločná z hľadiska prístupu pre mikropočítač aj videoprocesor s obvodmi zabezpečujúcimi správnu spoluprácu oboch podsystémov tak, aby nedochádzalo ku kolíziam na zbernici.

2.3 Klávesnica

Druhý samostatný diel zariadenia je klávesnica kontaktného typu na báze vodivého elastomeru, ktorá sa vyznačuje bezhluchným a mäkkým chodom s vysokou životnosťou. Okamih správneho zopnutia tlačítka je indikovaný akusticky s možnosťou vypnutia. Na doske klávesnice je umiestnený akustický menič. Prepojenie s doskou mikropočítača je 30-pólovým konektorm FRB. Obsahuje 55 tlačítok, z ktorých každé má tri významy volené

súčasným stlačením žiadaneho tlačítka s jedným z dvoch tlačítok označených **SHIFT** alebo **CONT**. Okrem štandardných alfanumerických znakov má ešte 22 grafických znakov, 12 softwarových klúčov s označením K0 - K11 a 18 tlačítok s riadiacimi funkciami. Klávesnica s kódmi prislúchajúcimi jednotlivým znakom je nakreslená na obr. č. 1.

2.4 Sieťový napájač

Sieťový napájač tvorí tretí diel zariadenia. Obsahuje sieťový transformátor a stabilizátory potrebných napäti +5V a -5V sistením proti skratu alebo preťaženiu. Vďaka dostačnému dimenzovaniu zdroja a dokonalému vetraniu nedochádza ani pri dlhodobej nepretržitej prevádzke k prehrievaniu dôležitých uzlov a k zlyhávaniu mikropočítača ako je to známe z používania iných typov mikropočítačov tejto kategórie.

Obr. č. 1 - Klávesnice a kódy tlačítek

21	!	22	“	23	”	24	#	25	\$	26	%	27	&	28	*	29	(5F)	3D	=	5E					
1	2	3	4	5	6	7	8	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
D0	31	D1	32	D2	33	D3	D4	34	D5	35	D6	36	D7	37	D8	38	D9	30	DA	2D	DB	5C					
71	Q	W	E	R	T	Z	U	I	O	6F	69	6F	70	60	Q	P	89	50	8A	40							
80	51	81	57	82	45	83	52	84	54	85	5A	86	55	87	49	88	4F	89	50	8A	40						
61	A	73	64	66	67	68	6A	6B	6C	K	J	I	L	2B	+	2A	*					STOP					
RST	8B	41	8C	8D	D44	F46	8F	47	90	H	G	F	91	4A	92	4B	93	4C	94	3B	95	3A					
09	14	!	79	78	63	C	V	B	N	M	6D	3C	<	3E	>	3F	/	2	1A	19							
0C	11	0A	13	0B	59	17	58	1E	43	1F	56	1B	42	1D	4E	1C	4D	12	2C	01	2E	0F	2F	10	08	03	18
																		UD	E0I	E0I	OD						
																		OD	OD	OD	OD						
																		20	SPACE	20							
																		20	CONT	SHIFT							

3. PRIPOJITEĽNÉ ZARIADENIA

K mikropočítaču je možné pripojiť v štandardnom prevedení nasledovné zariadenia:

- televízny čiernobiely prijímač nalaďaný na 9. kanál pásmá VHF
- bežný kazetový alebo cievkový magnetofón
- štandardné dosky interfejsov počítača PMD-85 /ROM - modul, dosku interfejsov, atď./
- perspektívne pákový ovládač s interfejsem typu CENTRONICS pre tlačiareň

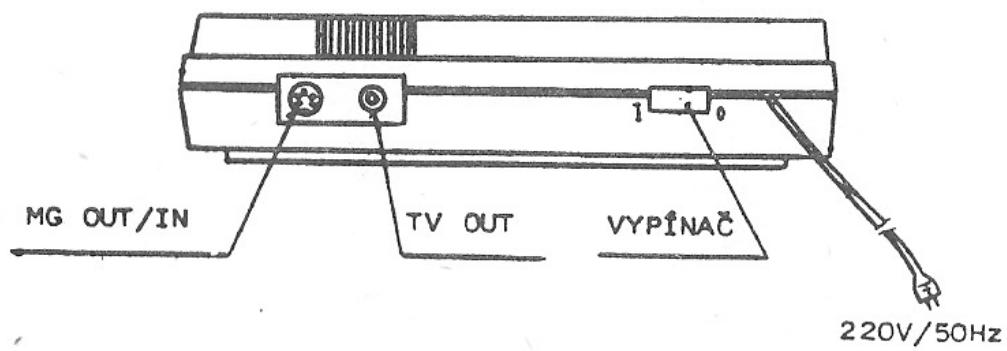
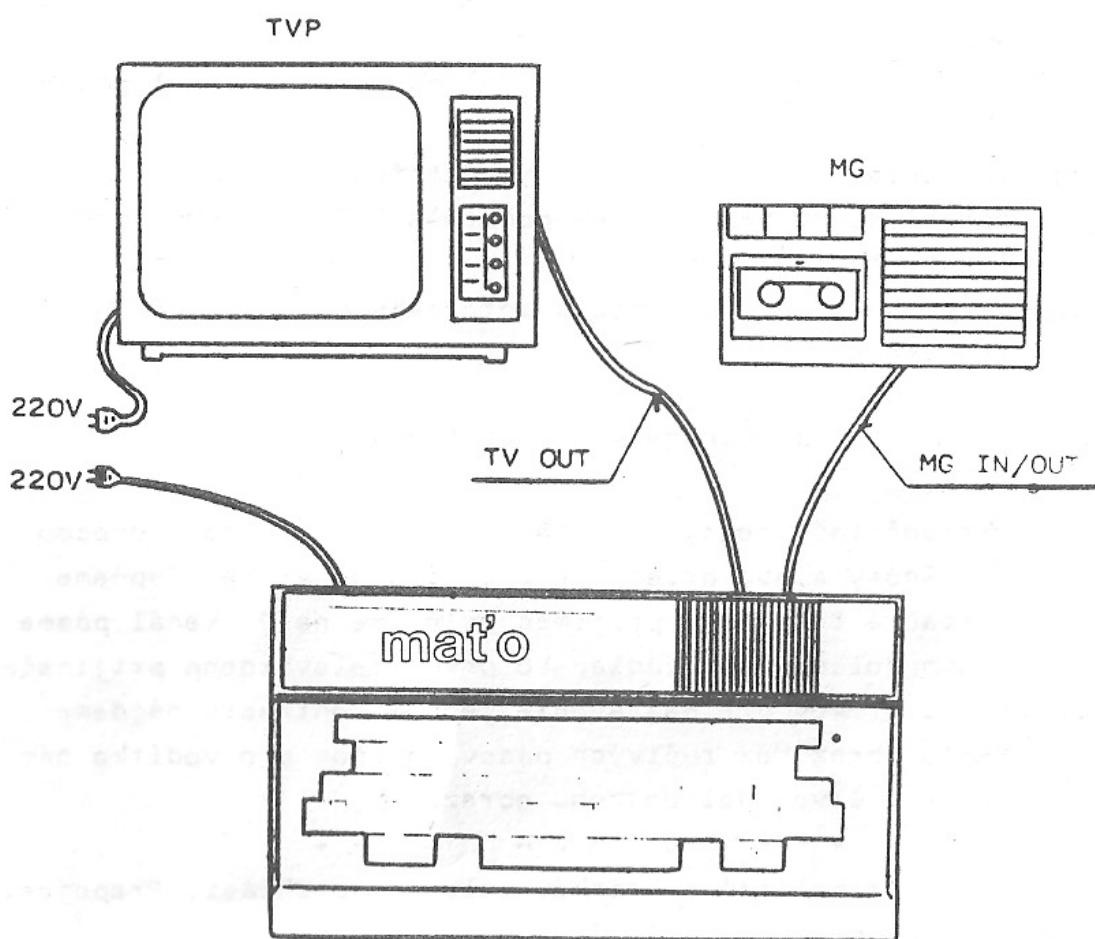
4. UVEDENIE DO PREVÁDZKY

Mikropočítač prepojíme s televíznym prijímačom pomocou dodávanej šnúry a oba prístroje zapojíme do siete. Zapneme mikropočítač a televízny prijímač nalaďime na 9. kanál pásmá VHF. Jemným dolaďovaním ladiaceho prvku televízneho prijímača a za pomoci prvkov pre nastavenie jasu a kontrastu nájdeme najostrejší obraz bez rušivých pásov, pričom ako vodítko nám slúži nápis v ľavom dolnom rohu obrazovky

+ + + O S R E A D Y + + +

ktorým sa mikropočítač s krátkou melódiou prihlásí. Prepojenie je znázornené na obrázku č. 2.

Obr. č. 2 Prepojenie mikropočítača



5. OBSLUHA MIKROPOČÍTAČA

Riadok, v ktorom je umiestnený spomínaný nápis sa nazýva "dialógový" a v tomto riadku s počítačom budeme komunikovať. Nad týmto riadkom je umiestnený prázdný priestor obrazovky nazývaný "pracovná oblast" pre výstup alfanumerickej alebo grafickej informácie. Formáty tohto priestoru vám vysvetlime neskôr. Teraz si vysvetlime, akým spôsobom budeme vkladať jednotlivé prikazy mikropočítaču.

5.1 Vkladanie a editovanie prikazov

Každý prikaz je zostavený ako reťazec znakov ASCII /to je reťazec zložený z písmen alebo číslic/, ktorý je ukončený /odoslaný na spracovanie/ stlačením najpoužívanejšej klávesy s označením **EOL**. Pred nápisaním prvého prikazu si vysvetlime, čo editovanie vlastne je.

Iste ste si už všimli, že na začiatku riadku je písmeno podtrhnuté vodorovnou čiarkou. Táto čiarka sa nazýva "kurzor". Označuje miesto, kde bude nápisany nasledujúci znak.

Nie vždy sa nám podarí /hlavne zo začiatku/ nápišať prikaz úplne správne a často potrebujeme vymazať alebo vložiť písmeno do textu, zmeniť písmená a pod. Za týmto účelom je mikropočítač vybavený radou tlačítok so skratkami rôznych funkcií zabezpečujúcich činnosti, ktoré uľahčujú editovanie v dialógovom riadku alebo vykonávajú službu so softwarovými klúčmi. Ich význam platí pri súčasnom stlačení žiadanej funkcie alebo znaku a tlačítka **CONT**.

Ich význam je nasledovný:

CONT + klávesa

BGNL	↑	kurzor na začiatok riadku
INVR	↓	prepinač inverznej tlače na obrazovku
WORK	Y	text v dialógovom riadku sa zapíše do klúča
RCL	X	vypíše sa obsah posledne odosланého dialógového riadku
C-D	C	vypíše sa posledný systémový oznam do dialógového riadku
CLS	✓	vymaže obrazovku

CLL	<u>B</u>	vymaže dialógový riadok
DELT	<u>N</u>	vymaže znak na mieste kurzoru
INST	<u>M</u>	vloží znak na mieste kurzoru
MON	<u>9</u>	návrat do operačného systému MONITOR
DELL	<u>*</u>	vymaže text od kurzoru do konca dialógového riadku
PTL	<u>/</u>	vypne - zapne kopírovanie dialógového riadku do pracovnej časti obrazovky
BEEP	<u>←</u>	vypne - zapne akustickú signalizáciu stlačenia klávesy
ENDL	<u>→</u>	kurzor na koniec riadku
KØ - K11		softwarové klúče
RES		systémové nulovanie mikropočítača

SHIFT + klávesa

<u>→</u>	riadok doľava
<u>←</u>	riadok doprava

klávesa

<u>→</u>	kurzor o jeden znak doprava
<u>←</u>	kurzor o jeden znak doľava

Pre úplnosť si uvedme, že pre výber funkcie alebo znaku z klávesnice platí nasledujúce pravidlo:

SHIFT + klávesa horný znak alebo malé písmená
CONT + klávesa dolný význam alebo grafické znaky
 klávesa význam alebo znak v strede klávesy

Povedzme si ešte, že súčasné stlačenie SHIFT + CONT + klávesa má význam uvedený v strede klávesy. Prehľadne je to uvedené na obrázku č. 3.

Obr. č. 3 Význam na klávese

<u>SHIFT</u> + znak	
<u>CONT</u> + znak	znak

Na základe týchto informácií môžete skúsiť pracovať s textom v dialógovom riadku. Vykonajte teraz nasledovné činnosti:

klávesa

dialógový riadok

CONT	+	CLL
B		_
A		B_
P		BA_
D		BAP_
-		BAPO_
1		BAPO-_

_
B_
BA_
BAP_
BAPO_
BAPO-_
BAPO-1_

Skúste si prepisovať, vkladať, vymazávať znaky za pomoci funkcií kláves popísaných v predchádzajúcom teste. Pamäťajte, že miesto, na ktoré ukazuje kurzor je miestom, na ktorom bude požadovaná funkcia vykonaná. Nacvičte si túto činnosť!

A teraz si ukážeme, ako pracujú "softwarové klúče". Klávesy s označením K0 - K11 nám spolu s funkciou WORK umožnia zapisať libovoľný text z dialógového riadku do miesta v pamäti mikropočítača označenej číslom klúča. Skúste si nasledovný príklad.

klávesa

dialógový riadok

SKUSKA	KLUCA	C	KØ
CONT	+	WORK	
CONT	+	K7	
CONT	+	CLL	
CONT	+	K7	

SKUSKA KLUCA C.KØ_
SKUSKA KLUCA C.KØ_
SKUSKA KLUCA C.KØ_
_
SKUSKA KLUCA C.KØ

Skúste sa uložiť iné texty do iných klúčov. Určite ste pochopili, akú výhodu vám poskytuje používanie klúčov, nemusíte zákaždým písať často sa opakujúci text odznovu, ale poštečí si ho na začiatku práce uložiť do príslušného klúča a v prípade potreby si ho vypisať na jedno stlačenie klávesy.

Môže sa vám stať, že ak máte dlhý program a obsadených mnoho klúčov s textami, že pri vkladani ďalšieho textu systém vypisuje správu

+ + + Memory Overflow + + +

čo značí, že už nemá miesto pre uloženie textu. Musíte skrátiť program alebo sa uspokojíť bez ďalšej služby softwarového klúča. To prirodzene platí aj naopak pri písaní programu.

Funkciu klávesy s označením [RCL] si ukážeme na nasledovnom príklade:

klávesa	dialógový riadok
CONT + CLL	-
MEM B Ø Ø Ø EOL	SUB 8ØØØ ØØ C3 ØØ 8Ø
EOL	SUB 8Ø1Ø -
CONT + CLL	-
CONT + RCL	SUB 8ØØØ ØØ C3 ØØ 8Ø

Vidíme, že klávesa [RCL] skutočne vrátila obsah posledne odoslaného dialógového riadku. Využijeme to vtedy, ak sme odoslali riadok, u ktorého nám systém zistil a oznámil chybu, napr. ďalším z oznamov.

+ + + No command + + +

A posledná klávesa [CONT + C-D] nám vypíše posledné systémové hlásenie mikropočítača, čo bude najskôr predchádzajúce uvedené, ak ste však pri doterajšej obsluhe neurobili chyby, ktoré by vám mikropočítač oznámil, bude to:

+ + + OS READY + + +

Ak ste pri študovaní tejto príručky dospeli až sem, určite je vám význam funkcií jednotlivých kláves jasný. Vyskúšajte si aj klávesy s grafickými symbolmi - pri odoslani klávesou [EOL] vám však systém bude vypisovať správu:

+ + + No command + + +

čo však vás nemusí trápiť, nakoľko mikropočítač pozná len

obmedzený počet príkazov, o ktorých si povieme v nasledovnom.

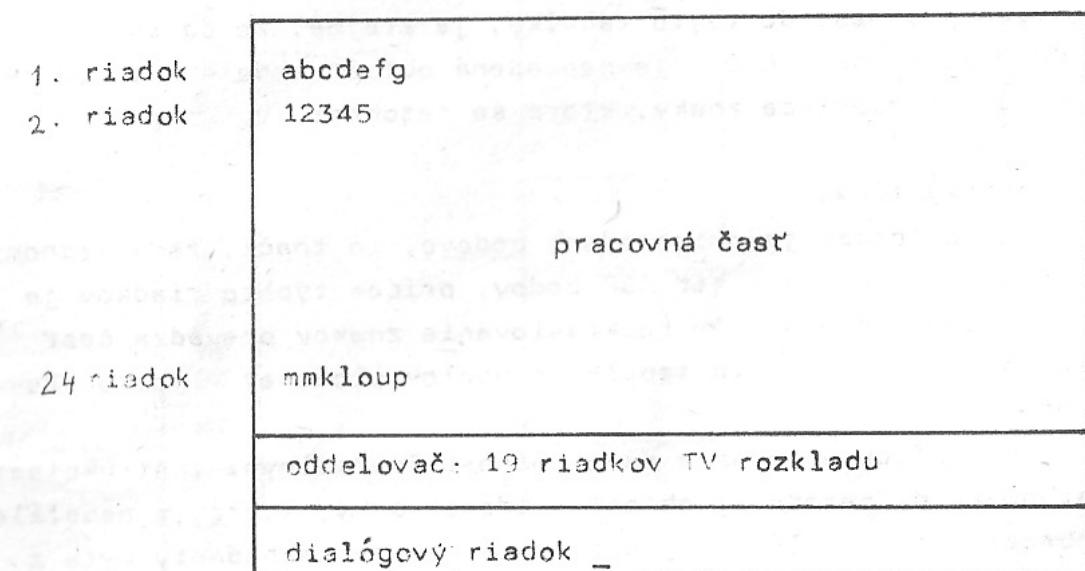
Teraz si ešte vysvetlime činnosť mikropočítača z hľadiska zobrazovania údajov na obrazovke.

5.2 Zobrazovanie

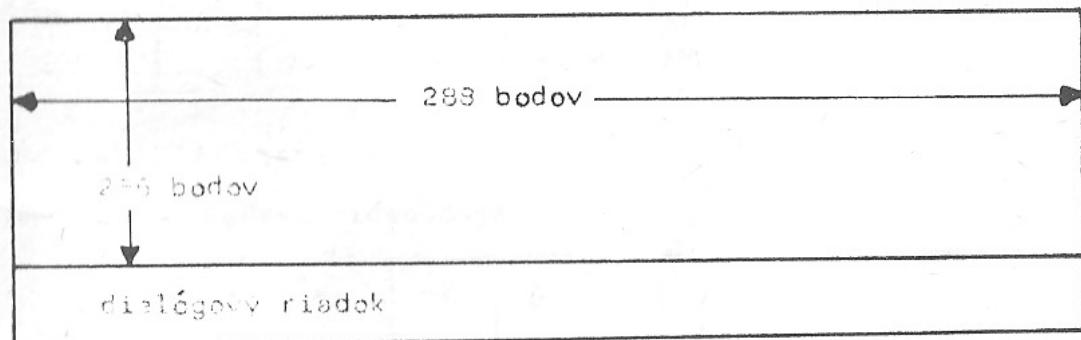
Na obr. č. 4 a č. 5 sú uvedené dva základné formáty pri zobrazovaní údajov v pracovnej časti obrazovky televízneho prijímača.

Obr. č. 4 Alfanumerický formát

48 znakov = 288 bodov



Obr. č. 5 Grafický formát



- Alfanumerický formát

Umožňuje zobrazovanie alfanumerických a grafických znakov uvedených na jednotlivých tlačítkach klávesnice /včitane malých písmen, ktoré ako bolo spomenuté sa radia cez [SHIFT]/, ktoré sú organizované do 25 riadkov po 48 znakov. Rozkreslovanie štandardných znakov je robené podľa tabuľky, ktorá je súčasťou základného programového vybavenia zvaného MONITOR.

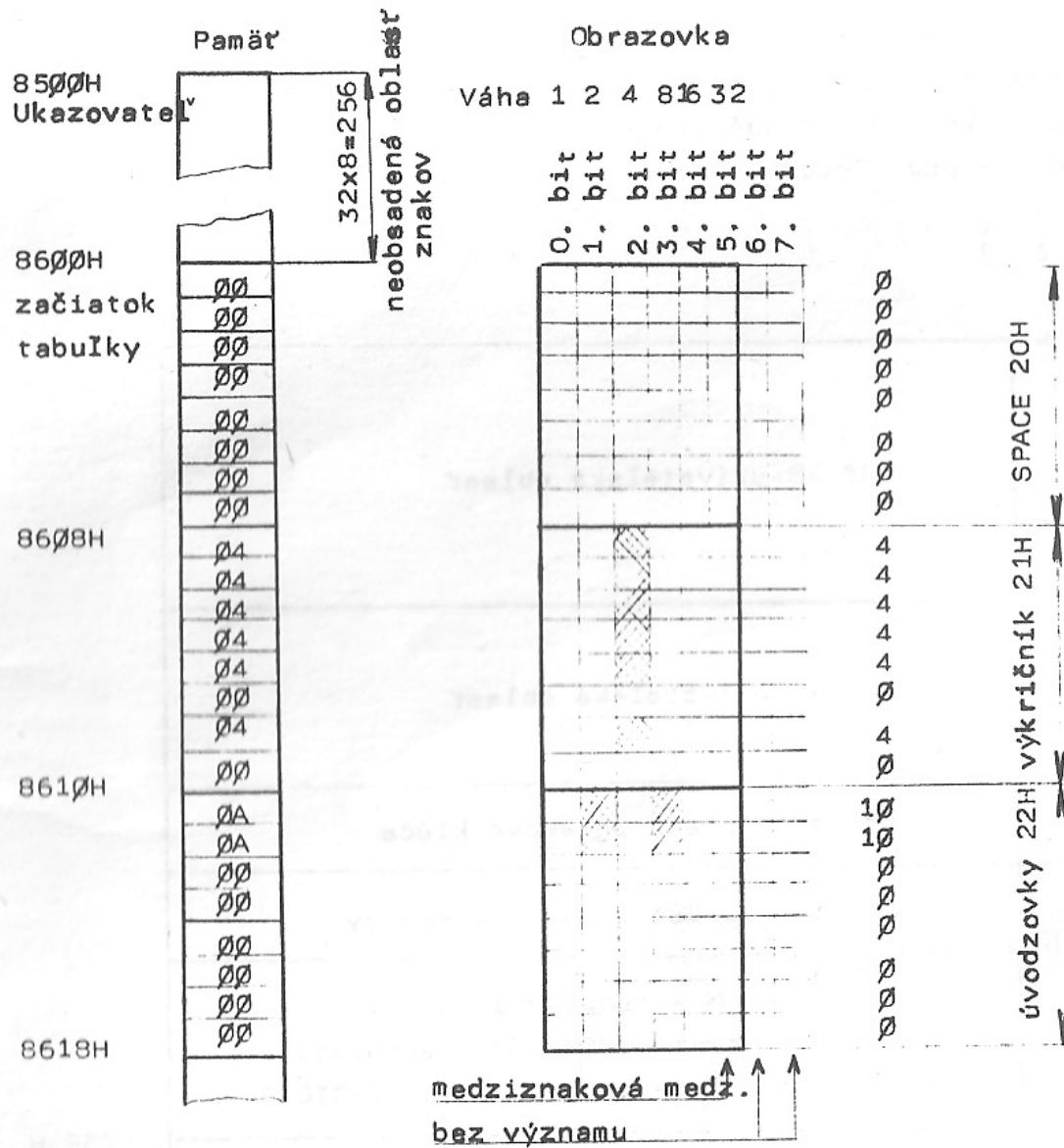
Vzťah medzi zobrazovaným byte v náväznosti na tabuľku znakov, kde je uložený kód pre rozkreslenie, je prehľadne znázorený na obrázku č. 6. Pretože kód znaku reprezentuje parameter nepriamej adresy do tejto tabuľky, je zrejmé, že od adresy 8500 H po adresu 8600 H je neosadená oblasť, kde ASCII kód je určený pre riadiace znaky, ktoré sa nezobrazujú.

- Grafický formát

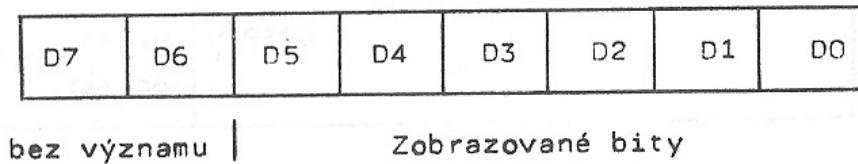
Tento formát je orientovaný bodovo, to značí, že v jednom riadku je možné zobraziť 288 bodov, pričom týchto riadkov je 256. Môžeme povedať, že rozkreslovanie znakov prevádzka časť programu MONITOR podľa tabuľky symbolov uloženej v tomto programe.

V grafickom formáte máme možnosť špeciálnymi inštrukciami vstupovať do pamäťovej oblasti videostránky, ktorá je neustále zobrazovaná na televíznej obrazovke zapísaním hodnoty byte z ktorého sa nám zobrazí len prvých 6. bitov a to tie, v ktorých je hodnota bitu rovná 1. Prečítaný údaj z videostránky má formát podľa obrázku č. 7.

Obr. č. 6 Rozkreslenie znaku



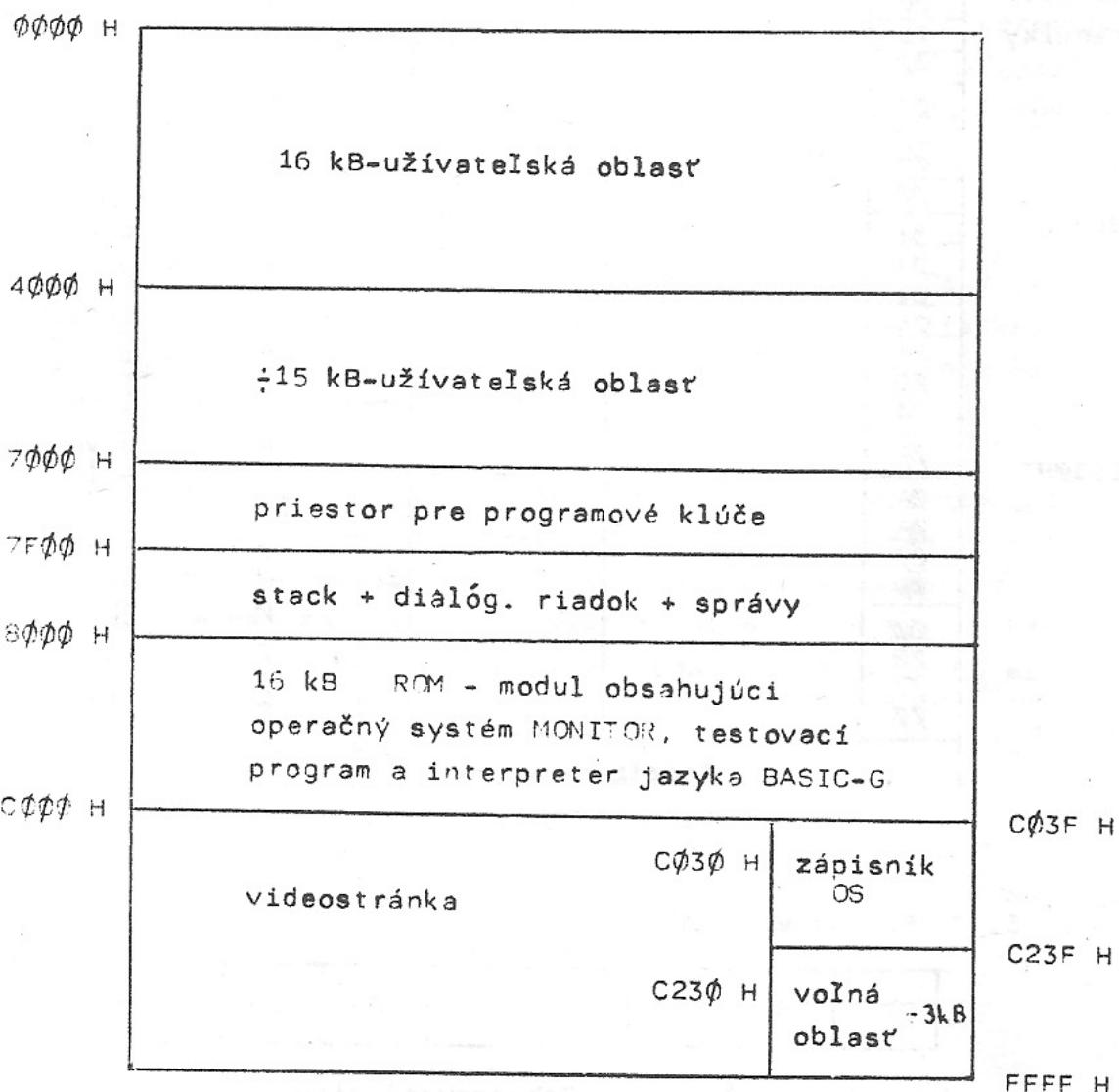
Obr. č. 7 Formát videoúdaja



Potrebaté dva bity sú bezvýznamné.

A na záver tejto časti si uvedieme prehľad rozdelenia a využívania operačnej pamäte mikropočítača, aby ste mali predstavu o tom, kde je váš program, kde je videostránka, zápisník systému a pod. Rozdelenie je na obrázku č. 8.

Obr. č. 8



6. MONITOR

Programové vybavenie nazývané MONITOR patrí medzi základné a umožňuje vykonávať úkony nad pamäťovým priestorom mikropočítača. Na obrázku č. 8 je vidieť ako je pamäťový priestor mikropočítača využívany. Príkazy programu MONITOR umožňujú programovanie mikropočítača na úrovni strojových inštrukcií v hexadecimálnom tvaru. K tomu potrebujeme prevodnú tabuľku hexadecimálnych kódov jednotlivých inštrukcií mikroprocesoru MHB 8080 A.

6.1 Tabuľka inštrukcií mikroprocesoru 8080 A

Inštrukcie presunu:			MOV ciel, zdroj	✓
MOV A,A	7F	MOV B,A	47	MOV C,A
MOV A,B	78	MOV B,B	40	MOV C,B
MOV A,C	79	MOV B,C	41	MOV C,C
MOV A,D	7A	MOV B,D	42	MOV C,D
MOV A,E	7B	MOV B,E	43	MOV C,E
MOV A,H	7C	MOV B,H	44	MOV C,H
MOV A,L	7D	MOV B,L	45	MOV C,L
MOV A,M	7E	MOV B,M	46	MOV C,M
MOV D,A	57	MOV E,A	5F	MOV H,A
MOV D,B	50	MOV E,B	58	MOV H,B
MOV D,C	51	MOV E,C	59	MOV H,C
MOV D,D	52	MOV E,D	5A	MOV H,D
MOV D,E	53	MOV E,E	5B	MOV H,E
MOV D,H	54	MOV E,H	5C	MOV H,H
MOV D,L	55	MOV E,L	5D	MOV H,L
MOV D,M	56	MOV E,M	5E	MOV H,M
MOV L,A	6F	MOV M,A	77	
MOV L,B	68	MOV M,B	70	
MOV L,C	69	MOV M,C	71	
MOV L,D	6A	MOV M,D	72	
MOV L,E	6B	MOV M,E	73	

MOV L,H	6C	MOV M,H	74
MOV L,L	6D	MOV M,L	75
MOV L,M	6E	XCHG	EB

Inštrukcie zápisu 8-bitového operandu: MVI ciel, operand

MVI A, byte	3E	MVI E, byte	1E
MVI B, byte	06	MVI H, byte	26
MVI C, byte	0E	MVI L, byte	2E
MVI D, byte	16	MVI M, byte	36

Inštrukcie zápisu 16-bitového operandu: ✓ LXI ciel, operand

LXI B, dbl	01	LXI D, dbl	11
LXI H, dbl	21	LXI SP, dbl	31
LDHI byte	28	LDI byte	38

Inštrukcie čítania a zápisu do pamäte:

LDAX B	0A	LDAX D	1A
LHLD adr	2A	LDA adr	3A
STAX B	02	STAX D	12
SHLD adr	22	STA adr	32

Inštrukcie sčítania 8-bitových operandov: ADD reg

ADD A	87	ADC A	8F
ADD B	80	ADC B	88
ADD C	81	ADC C	89
ADD D	82	ADC D	8A
ADD E	83	ADC E	8B
ADD H	84	ADC H	8C
ADD L	85	ADC L	8D
ADD M	86	ADC M	8E

Inštrukcie odčítania 8-bitového operandu: SUB reg

SUB A	97	SBB A	9F
SUB B	90	SBB B	98
SUB C	91	SBB C	99
SUB D	92	SBB D	9A

pokračovanie

SUB E	93	SBB E	9B
SUB H	94	SBB H	9C
SUB L	95	SBB L	9D
SUB M	96	SBB M	9E

Inštrukcie sčítania 16-bitových operandov:

DAD dbl ✓

DAD B	09	DAD D	19
DAD H	29	DAD SP	39
D SUB B	08		

Inštrukcie inkrement-dekrement register:

INR A	3C	DCR A	3D
INR B	04	DCR B	05
INR C	0C	DCR C	0D
INR D	14	DCR D	15
INR E	1C	DCR E	1D
INR H	24	DCR H	25
INR L	2C	DCR L	2D
INR M	34	DCR M	35

Inštrukcie inkrement-dekrement dvojice registrov:

INX B	03	DCX B	0B
INX D	13	DCX D	1B
INX H	23	DCX H	2B
INX SP	33	DCX SP	3B

Inštrukcie logické:

ANA A	A7	XRA A	AF	ORA A	B7	CMP A	BF
ANA B	A0	XRA B	A8	ORA B	B0	CMP B	B8
ANA C	A1	XRA C	A9	ORA C	B1	CMP C	B9
ANA D	A2	XRA D	AA	ORA D	B2	CMP D	BA
ANA E	A3	XRA E	AB	ORA E	B3	CMP E	BB

pokračovanie

ANA H	A4	XRA H	AC	ORA H	B4	CMP H	BC
ANA L	A5	XRA L	AD	ORA L	B5	CMP L	BD
ANA M	A6	XRA M	AE	ORA M	B6	CMP M	BE

5

Inštrukcie aritmetické a logické s 8-bitovým operandom:

ADD byte	C6	ACI byte	CE
SUB byte	D6	SBI byte	DE
ANI byte	E6	XRI byte	EE
ORI byte	F6	CPI byte	FE

Funkcie rotácie a špeciálne:

RLC	07	RRC	0F	RAL	17	RAR	1F
DAA	27	CMA	2F	STC	37	CMC	3F

3 ARITH 10

EDZL 18

Inštrukcie skokov, volania podprogramov a návratov z podprogramov:

JMP adr	4	C3	CALL adr	5	CD	RET	3	C9
JNZ adr	4	C2	CNZ adr	4	C4	RNZ	3	C0
RZ adr	3	CA	CZ adr	3	CC	RZ	2	C8
JNC adr	4	D2	CNC adr	4	D4	RNC	3	D0
JC adr	3	DA	CC adr	3	DC	RC	2	D8
JPO adr	4	E2	OP0 adr	4	E4	RPO	3	E0
JPE adr	4	EA	CPE adr	4	EC	RPE	3	E8
JM adr	3	FA	CM adr	3	FC	RM	2	F8
JP adr	3	F2	CP adr	3	F4	RP	2	F0
PCHL	4	E9	SWX5	5	DD	SWX5	5	DD

Inštrukcie zápisníku, vstupu, výstupu a riadenia:

PUSH B	6	C5	POP B	5	C1	OUT byte	6	D3
PUSH D	6	D5	POP D	5	D1	IN byte	3	DB
PUSH H	6	E5	POP H	5	E1	DI	3	F3
PUSH PSW	6	F5	POP PSW	5	F1	EI	2	FB
XTBL	6	E3	SPHL	5	F9	NOP	3	00
RIM	7	Z0	SWH	5	30	HLT	3	76

Inštrukcie prerušenia programu:

RST 0	00	C7	RST 1	08	CF	RST 2	10	D7
RST 3	18	DF	RST 4	20	E7	RST 5	28	EF
RST 6	30	F7	RST 7	38	FF	RSTV		CB

Úplné vysvetlenie funkcie jednotlivých inštrukcií mikroprocesoru 8080 A by zabralo samostatnú knihu. Preto pre podrobnejšie štúdium doporučujeme literatúru uvedenú na konci tejto príručky.

6.2 POPIS MONITORU MIKROPOČÍTAČA

Nové pojmy: ADRESA - štvorica hexadecimálnych číslic. V prípade chyby v zadani adresy /nesprávny počet, iný ako hexa znak a pod./ mikropočítač vypisuje správu
+++ ERROR IN ADDRESS +++

DATA - dvojica hexadecimálnych číslic. Ak je chyba v zadani, bude vypísaná správa
+++ ERROR IN DATA +++, v prípade, že dáta nebudú zadané vôbec, vypíše sa
+++ NO DATA +++.

Nie je potrebné, aby ste jednotlivé polia v príkaze oddeľovali medzerami. Program MONITOR to rozpozná a vy si zjednodušíte prácu. Ak však chcete mať prehľad v písaných textoch, kludne môžete medzery ako oddelovače použiť, pretože počítač toleruje ľubovoľný počet medzier.

Priaz SUB - modifikácia pamäťového miesta RWM

Tvar prikazu: SUB adresa data data

Ak chcete vkladať priamo ASCII znaky, môžete ich vkladať rovnako ako dáta, ale na to musíte mikropočítač upozorniť pred prvým znakom a za posledným znakom /ak ešte mienite vkladať hexadecimálne data/. Urobíte to zvláštnym znakom /' - apóstrof. Tento prikaz treba používať zo začiatku opatrne, aby ste neprepísali niektoré dôležité pamäťové miesta, ktoré mikropočítač potrebuje k práci. Preto doporučujeme používať v začiatkoch priestor od adresy 0000 H - 7000 H.

PRÍKLAD: **S U B 1 0 0 0 0 1 0 2 0 3 EOL**
SUB 1003 A B C D EOL
SUB 1007 -

Pričaz MEM - zobrazenie 16 byte dát od zadanej adresy do dialógo-vého riadku

Tvar pričazu: MEM adresa

Pričaz zobrazí 16 po sebe nasledujúcich byte ako dvojice hexadecimálnych číslic, oddelených medzou a automaticky prechádza do módu pričazu SUB, takže pomocou editačných kláves môžete priamo modifikovať príslušné pamäťové miesto.

PRÍKLAD: **CONT + CLR**

M E M 1 D 0 D EOL a v dialógovom riadku uvidíme
SUB 1000 31 32 33 41 42 43 44

Pričaz DUMP - výpis pamäťovej oblasti od zadanej adresy

Tvar pričazu: DUMP adresa

Pričaz zobrazuje obsah zadanej pamäťovej oblasti v tvare:

ADRESA 8 x DATA 8 x ASCII TVAR

Po zaplnení obrazovky nastane tzv. rolovanie jej obsahu, kde sa riadky presúvajú smerom hore. Rolovanie možno pozastaviť stlačením klávesy **SHIFT**, prípadne ukončiť funkciu DUMP definitívne stlačením klávesy **STOP**.

PRÍKLAD: **CONT + CLR**

D U M P 1 0 0 0 EOL a na obrazovke beží výpis

Pričaz JUMP - štart programu od zadanej adresy

Tvar pričazu: JUMP adresa

Tento pričaz je veľmi dôležitý, nakoľko počítač nezodpovedá za riadenie systému. Riadenie plne preberá

Váš užívateľský program, pričom mikropočítač ešte
stihne podať správu
+++ EXECUTIVE +++.

PRÍKLAD: **JUMP 8000 EOL** spustí program MONITOR od začiatku ako
pri zapnutí

Pričaz MGLD - nahranie zadaného súboru do operačnej pamäte z magnetofónu

Tvar príkazu: MGLD file

Tento príkaz vyhľadá na páske súbor s udaným číslom "file", čo môže byť v rozsahu 00 až 99 a uloží ho od adresy, ktorá je uvedená v jeho hlavičke. Pri prehliadaní iných súborov na páske s číslom, ktoré sa nezhoduje so zadaným, vypisuje tieto do pracovnej časti obrazovky. Ak teda chceme získať výpis obsahu kazety, postačí zadať MGLD 99, (ak nemáme na páske súbor s týmto číslom) a na obrazovke sa nám postupne vypíše obsah pásky. Výpis je možné kedykoľvek zrušiť stlačením klávesy **STOP**. Pri takomto prerušení alebo v prípade, že sa program nahral späť, vypíše mikropočítač správu

+++ FILE ERROR +++.

PRÍKLAD: **M GLD 00 EOL**

Pričaz odštartujte po vložení kazety s programom po pretočení na začiatok a spustení magnetofónu na prehrávanie.

Po prečítaní hlavičky bude táto vypísaná do pracovnej časti obrazovky. Po ukončení nahrávania sa počítač prihlási krátkym tónom a vypíše "?" v dialógovom riadku. Startovaciu adresu programu nájdete v zápisníku na adrese C1B4. Vykonajte pričaz **MEMC1B4 EOL**.

V dialógovom riadku sa zobrazí SUB C1B4 00 10

Je to štartovacia adresa v poradí
nižší byte, vyšší byte teda 1000 H.
Program potom môžeme spustiť príkazmi:
CONT + **CLR**
JUMP 1 000 EOL

Pričaz MGSV - nahranie dát alebo programu na magnetofón

Tvar príkazu: MGSV file; od - do; komentár
Pričaz nahrá dát uložené v operačnej pamäti
počítača na magnetofón, kde parametre v pri-
kaze značia:
file - číslo file v rozsahu 0 až 99
od - adresa začiatku zaznamenaných dát /hexa/
do - adresa posledného miesta zaznamenaných
dát /hexa/
komentár - nepovinný reťazec max. osem ASCII
znakov s názvom programu
Po skončení nahrávania vypíše mikropočítač
+++ MG STOP +++.

PRÍKLAD:

MGSV 01;1000-0,0,0,0HRAEOL

Pričaz odštartujte po založení čistej pásky,
pretočení na začiatok a spustení nahrávania.
Dĺžku programu nájdete na adrese C1B6 spôsobom
ako v prípade štartovacej adresy. Ale pozor.
Do pričazu MGSV zadávate koncovú adresu pro-
gramu, ktorú získate sčítaním počiatočnej adresy
s dĺžkou programu, prirodzene v hexadecimálnej
sústave.

Pričaz MGEND - kontroluje správnosť nahrávky

Tvar pričazu: MGEND file
Pričaz má dve funkcie. Zistuje koniec žiadane-
ho súboru, čo je dôležité pre pridávanie ďal-
ších súborov a zistuje správnosť nahrávky na

páske. V prípade, že pri nahrávani došlo k chybe, napr. vplyvom nekvalitnej pásky, nečistote a pod., vypíše oznam +++ FILE ERROR +++

PRÍKAZ: **M G E N D Ø 1 EOL**

Prikazom si prevedieme kontrolu záznamu nahratého v predchádzajúcom. Pretočte pásku na začiatok a zapnite magnetofón na prehrávanie. Po 2 - 3 sec. odštartujte prikaz. Po prečítaní hlavičky počítač túto vypíše a po ukončení kontroly sa prihlási krátkym tónom; pokiaľ prirodzene záznam bol bezchybný. Ináč vypíše chybový oznam.

Priaz BASIC - presunie interpreter jazyka BASIC-G do operačnej pamäte a odštartuje ho.

Tvar príkazu: **B A S I C EOL**

Po vykonaní príkazu sa v pracovnej časti obrazovky prihlási interpreter výpisom +++ BASIC - G V2.1 +++ a je pripravený na prácu. Prihlási sa novým znakom v dialógovom riadku ">". Popis práce s týmto vyšším programovým vybavením je uvedený v ďalšej časti tejto príručky.

7. PROGRAMY MODULU MONITOR

Okrem už uvedených štandardných príkazov mikropočítač môže poskytnúť služby celého radu podprogramov z programového modulu MONITOR. Tieto môžete s výhodou využívať pri tvorbe vlastných programov za predpokladu, že máte znalosti z programovania a štruktúry systému 8080.

7.1 **[HEX]** - podprogram na prevod čísla ASCII do HEX. V prípade chyby, ak nejde o znak z množiny hexa znakov /Ø až F/ nastavuje sa CARRY bit.

Vstup: akumulátor

Výstup: akumulátor a príznakový bit CARRY

Volanie: 80E0 H

Mení obsah registrov: PSW

7.2 **PAIRIN** - Podprogram na prevod 2 x ASCII do HEXA. V prípade chyby nastaví sa príznakový bit CARRY.

Vstup: ukazovateľ HL

Výstup: akumulátor a príznakový bit CARRY

Volanie: 80F7 H

Mení obsah registrov: PSW, HL, B

7.3 **ADRIN** - podprogram na prevod 4 x ASCII do 2 x HEXA. V prípade chyby nastaví sa príznakový bit CARRY.

Vstup: ukazovateľ HL

Výstup: registrový par DE a príznakový bit CARRY

Volanie: 8109 H

Mení obsah registrov: všetkých

7.4 **BIN/BCD** - podprogram na prevod binárneho čísla /HEXA/ na BCD v rozsahu 0 až 99.

Vstup: akumulátor

Výstup: akumulátor

Volanie: 8E73 H

Mení obsah registrov: HL, PSW, B

7.5 **PREVOD 1** - podprogram na prevod HEXA na 2 x ASCII pri ich súčasnom zobrazení do pracovnej časti obrazovky. Využíva podprogram PRTOUT, ktorý vyžaduje nastavenie hodnôt do ukazovateľov a je popísaný ďalej.

Vstup: akumulátor

Výstup: podprogram PRTOUT /8500H/

Volanie: 8125 H

Mení obsah registrov: B, PSW

7.6 [PREVOD 2] - podprogram na prevod HEXA na 2 x ASCII pri ich súčasnom uložení do pamäte a na adresu určenú ukazovateľom HL.

Vstup: akumulátor

Výstup: pamäťové miesto určené ukazovateľom HL

Volanie: [813B] H

Mení obsah registrov: HL, PSW, B

7.7 TRANSFER - podprogram na prenos údajov z ROM modulu do operačnej pamäte mikropočítača. Argumenty sa ukladajú za volaciu inštrukciu v poradí: počiatocná adresa v ROM, dĺžka bloku, ukladacia adresa v operačnej pamäti.

Volanie: 8000 H

Mení obsah registrov: všetkých

Priklad použitia: CALL TRANSFER	CD 00 8C
DW ADR 1	00 00
DW BATCH	00 25
DW ADR 2	00 00
JMP 00 00	C3 00 D0

Tento program načíta interpreter jazyka BASIC zo štandardného ROM modulu počítača PMD 85 pripojeného na aplikačný konektor systémovej zbernice na mikropočítač MAŤO a odštartuje ho.

7.8 [PAUSE] - podprogram vykoná časové spozdenie dané veľkosťou parametru vloženého do registrovej dvojice DE

Vstup: parameter v registrovej dvojici DE

Výstup: časové spozdenie

Volanie: 88B5 H

Mení obsah registrov: DE

7.9 [STOP] - podprogram zabezpečuje kontrolu stlačenia tlačítka

[STOP]. V prípade, že sa tak stalo, akumulátor nadobúda hodnotu $\emptyset 3$ H.

Vstup: klávesa **[STOP]**

Výstup: akumulátor

Volanie: 8C74 H

Mení obsah registrov: PSW

7.10 RPOINT - podprogram slúžiaci ako podpora k interpolátoru jazyka BASIC-G a umožňuje zo zadaných súradnic bodu zistiť jeho stav / \emptyset alebo 1/. Využíva podprogram POINT a s ním súvisiace jeho argumenty.

Vstup: argumenty podprogramu POINT

Výstup: akumulátor nadobudne hodnoty \emptyset alebo 1

Volanie: 85E6 H

Mení obsah registrov: všetkých

7.11 **[MONIT]** - podprogram zabezpečuje vstup do monitoru z užívateľského programu

Volanie: 8C4 \emptyset H

7.12 ENTER - podprogram zabezpečuje správnu korešpondenciu medzi užívateľským programom a operačným systémom. Umožní odloženie vrcholu zásobníku a návratovej adresy do užívateľského programu a nastavuje vlastný vrchol zásobníku na hodnotu 7FFF H.

Výstup z podprogramu a pokračovanie v užívateľskom programe nastane po stlačení klávesy **[EOL]**.

Volanie: 8BEE H

7.13 **[POINT]** - podprogram na samotné zobrazenie bodu na obrazovke, ktorý vypočíta zo zadaných súradnic polohu bodu vzhľadom k strojovej nule /C000 H/ a v samotnom byte pozíciu bitu.

Vstup: zápisník na adrese

- L C~~7~~0 H súradnica X
H C~~7~~2 H súradnica Y
C1FA H mod - A~~D~~8H negácia bodu
AF H nulovanie bodu
B~~0~~ H ... nastavenie bodu

Volanie: 8C7D H

Mení obsah registrów: PSW

7.14 **INPOL** - podprogram vytvára lineárne spojenie medzi dvoma
ľubovoľnými bodmi v pracovnej časti obrazovky.
Z dôvodu väčšej rýchlosťi si mikropočítač zníži
šírku pracovnej časti obrazovky na 256 bodov.

Vstup: zápisník na adrese

- C170 H súradnica X posledného bodu
C172 H súradnica Y posledného bodu
C173 H súradnica X žiadaneho bodu
C174 H súradnica Y žiadaneho bodu

Výstup: podprogram POINT

Volanie: 8CD0 H

Mení obsah registrów: všetkých

7.15 **EDIT** - podprogram zabezpečuje interaktívnu činnosť nad
obsahom dialógového riadku. Po prijatí vstupného
znaku rozhodne, či sa jedná o alfanumerický znak
- v tom prípade ho ihned rozkreslí nad pozíciou
kurzoru, ak ide o riadiaci znak vykoná príslušnú
činnosť ako je popísaná v časti 5.1.

Vstup: znak na adrese C134 H

Volanie: 8800 H

7.16 **BELL** - podprogram zabezpečuje generovanie ľubovoľných melódií podľa parametrov zadaných tabulkou užívateľa.
Parametre predstavujú za sebou nasledujúce dvojice
HEXA čísel, pričom prvá dvojica udáva dĺžku a druhá dvojica výšku tónu. Tabuľka musí byť ukončená
znakom FF H a jej začiatok udaný v registrovanej
dvojici HL.

Vstup: tabuľka parametrov tónov na adrese zadanej v dvojici HL

Výstup: akustický signál

Volanie: 88A6 H

Mení obsah registrů: PSW, DE, HL

Príklad použitia:	LXI H,TAB	6000	21 09 60
	CALL 88A6		CD A6 88
	JMP 8C40		C3 40 8C
TAB:	DB 03H, 10H	6009	03 10
	DB 08H, 15H		08 15
	DB 10H, 20H		10 20
	DB 0FFH		FF

Program vygeneruje tri rôzne tóny s narastajúcou dĺžkou.

7.17 [ZOBRBUF] - podprogram vykoná zobrazenie reťazca znakov uloženého v pamäti, ktorého začiatok je daný ukazovateľom v registrovej dvojici HL do dialógového riadku. Reťazec musí byť ukončený znakom 0D H.

Vstup: adresa začiatku reťazca v registrovej dvojici HL

Volanie: 8858 H

Mení obsah registrů: všetkých

7.18 [PRTTEXT] - podprogram zabezpečí zobrazenie reťazca znakov uloženého v pamäti, ktorého začiatok je daný na adrese C074 H do dialógového riadku. Reťazec musí byť ukončený znakom 0D H.

Vstup: ukazovateľ začiatku reťazca uložený na adrese C074 H

Volanie: 8A89 + BELL

8A8C H bez BELL

Mení obsah registrů: všetkých

7.19 [KLAV] - podprogram zabezpečuje obsluhu klávesnice priradením ASCII znakov grafických a riadiacich znakov umiestnených v tabuľke od adresy 83F2 H.

Vstup: klávesnica

Výstup: akumulátor alebo zápisník na adrese C134 H
Volanie: 84A1 H
Mení obsah registrov: žiadnych

7.20 **PRTOUT** podprogram zabezpečuje rozkreslenie znaku do videostránky pamäte na miesto, ktoré udáva pozícia kurzoru v zápisníku na adr. C03E H. Tento podprogram prijme okrem znakov, ktoré vie rozkresliť aj tieto riadiace znaky:

20FF H prázdný znak
0D H návrat a nový riadok
1C H mazanie obrazovky

Výstup: znak v akumulátore
zápisník systému C03E H pozícia kurzoru - inicializované 0000 H

Volanie: 8500 H
Mení obsah registrov: PSW

7.21 PIXEL - podprogram prevedie vlastné rozkreslenie znaku na pozícii kurzoru. Treba upozorniť, že tabuľka znakov z dôvodov kompatibility na PMD-85 je rozdelená na dve časti. Tabuľka pre rozkreslenie veľkých znakov začína na adrese 8600 H, tabuľka pre rozkreslenie malých písmen začína na adrese 88BE H a tabuľka pre rozkreslenie grafických znakov začína na adrese 89BE H.

Volanie: B5A4 H
Mení obsah registrov: PSW

7.22 **SCAN** - podprogram "ohmatá" klávesnicu a do akumulátoru v prípade stlačenej klávesy priradí jej kód, ak nie je stlačená žiadna klávesa, priradí kód 00.

Volanie: 8C03
Výstup: akumulátor s kódom klávesy
Ničí registre: PSW

Priklad použitia: LOP: LXI H, D010	6000	21 10 D0
SHLD C03E		22 3E C0
CALL 8C03		CD 03 8C
CALL 8125		CD 25 81
DUMP LOP		C3 00 60

Priklad číta a zobrazuje kódy stlačených kláves v hexadecimálnej sústave.

Zhrnutie:

- a/ do videostránky vložte údaje svojho mena
- b/ skontrolujte tento obsah prikazom MEM
- c/ skontrolujte tento obsah prikazom DUMP a porovnajte ho s výpisom urobeným prikazom MEM
- d/ hľadajte odlišnosti a porozmyšľajte o ich dôvode
- e/ zostavte a vykonajte program na naplnenie videostránky ľubovoľným znakom ASCII.

V doteraz uvedenom texte sme prebrali prvú a obtiažnejšiu časť informácií o počítači MAŤO

Druhá časť bude zameraná na komunikovanie s počítačom na vyšej úrovni, ktorá podstatne uľahčuje prácu užívateľovi.

8. PROGRAMOVANIE V JAZYKU BASIC - G

V tejto časti textu vás oboznámime s programovacím jazykom BASIC-G, ktorý je špeciálne orientovaný na grafické spracovanie informácií. Upozorňujeme, že nie je možné popisať všetky kombinácie prikazov a možnosti, ktoré tento jazyk dáva. Záleží od šikovnosti programátora, ako využie získané znalosti. Preštudujte si preto pozorne uvádzané príklady a porozmýšľajte nad ich zmyslom. Dobrou pomôckou je štúdium hotových jednoduchších programov, popisovaných v odbornej literatúre a v časopisoch.

Teraz pomocou príkazu operačného systému zavolajte vyšší programovací jazyk prikazom

BASIC EOL

Bezprostredne po tom mikropočítač podá správu:

BASIC - G/ V2.1

a v dialógovom riadku sa objavi "OK". Od tohto okamžiku môžete komunikovať s mikropočítačom na vyššej úrovni, pričom sa vám bude v ľavom rohu dialógového riadku objavovať aj znak ">".

Programovací jazyk BASIC-G umožňuje dva režimy práce. Priamy režim vyznačujúci sa tým, že prikazový riadok nezačína číslom - prikaz po odoslani sa ihneď vykoná a programovým režimom, kde sekvencia povelov a prikazov je napísaná na číslovaných riadkoch.

Najskôr si však musíme vysvetliť funkciu niektorých špeciálnych kláves, ktoré BASIC-G používa. Sú to tieto klávesy:

\$	označenie reťazcovej premennej
"	označenie začiatku a konca reťazca znakov ASCII
.	desatinná bodka
/	čiarka slúžiaca ako oddelovač
:	bodkočiarka slúžiaca na formátovanie výpisu
?	skrátený povel na vytlačenie hodnoty do dialógového riadku
()	zátvorky pre aritmetické výrazy
> = <	relačné operátory slúžiace na porovnávanie hodnôt
+ - * / ^	aritmetické operátory /sčítanie, odčít., násobenie, delenie, umocn./
:	dvojbodka slúžiaca ako oddelovač pre viacnásobné príkazy
SPACE	medzera
STOP	zastavenie programu

Pred uvedením prvého príkladu musíte sa oboznámiť s matematickým aparátom mikropočítača.

8.1 ČÍSELNÝ FORMÁT

Všetky čísla, ktorých absolútна hodnota je v rozsahu ± 1 až 999999 sú zobrazované v pevnej desatinnej čiarke bez exponentu. Všetky ostatné sú zobrazené v exponenciálnom tvare:

S M . MMMMM E S XX

S	predstavuje znamienko + alebo -, pričom + sa nezobrazuje
M	mantisa čísla
E	identifikátor exponentu - skratka EXPONENT
X	exponent daného čísla

Matematický aparát pracuje v číselnom rozsahu 1×10^{-38} až 1×10^0 E +38, v pevnej desatinnej čiarke na 6 platných desatinnych miest.

8.2 ARITMETICKÉ PREMENNÉ

Aritmetické premenné môžete označovať až dvoma znakmi, pričom prvý znak musí byť písmeno a druhý písmeno alebo číslica.

Napríklad: X, Y, A, A1, Z6, AB ER atď.

Aritmetické premenné slúžia na uchovanie hodnoty aritmetického, logického alebo relačného výrazu.

8.3 OPERÁTORY ARITMETICKÉHO VÝRAZU

Symbol	Výraz	Význam
+	A + B	sčítanie
-	A - A	odčítanie
^	A ^ B	umocnenie A na B
*	A * B	násobenie
/	A / B	delenie
znamienko +	+ A	kladné číslo A
znamienko -	- A	záporné číslo A

Jednotlivé operátory môžete v aritmetických výrazoch používať viackrát, prípadne ich kombinovať.

Napríklad: $2\phi - ((A + B) \cdot C)^D$ alebo
 $AA + D3 + (-X1/44)^2$ a pod.

8.4 OPERÁTORY V LOGICKÝCH VÝRAZOCH

Symbol	Výraz	Význam
OR	A OR B	logický súčet
AND	A AND B	logický súčin
NOT	NOT A	dvojkový komplement A

Logické operátory používame vtedy, keď chceme pri rozhodovaní podmieňovať výsledok operácie splnením daných vstupných podmienok.

Napríklad: 1ϕ if A AND B = 1 then STOP
 2ϕ let A = 1

8. 5 OPERÁTORY V RELAČNÝCH VÝRAZOCH

Symbol	Výraz	Význam
=	A = B	A je rovné B
<	A < B	A je menšie ako B
>	A > B	A je väčšie ako B
≤	A ≤ B	A je rovné alebo menšie ako B
≥	A ≥ B	A je rovné alebo väčšie ako B
≠	A ≠ B	A sa nerovná B

Pri logických a relačných operátoroch si treba uvedomiť, že výsledná hodnota /myslené ako vyhodnotenie výrazu mikropočítačom/ môže nadobúdať len dve hodnoty a to:

"1" pravdivý výraz

"0" nepravdivý výraz

Operátor NOT je z tohto tvrdenia výnimka, pretože výsledkom vyhodnotenia je dvojkový komplement čísla premennej.

8.6 MATEMATICKÉ FUNKCIE

FUNKCIA	Význam
SIN (X)	sínus čísla X
COS (X)	cosínus čísla X
TAN (X)	tangens čísla X
ATN (X)	arctangens čísla X
LOG (X)	prirodzený logaritmus čísla väčšieho ako 0
EXP (X)	x-tá mocnina základu prirodzeného logaritmu "e"
SQR (X)	druhá odmocnina čísla X väčšieho ako 0
ABS (X)	absolútна hodnota čísla X
INT (X)	celočíselná hodnota čísla X
RND (X)	generovanie náhodného čísla z intervalu 0 až 1
SGN (X)	algebraická hodnota čísla X

Pozor! Pri používaní goniometrických funkcií SIN (X), COS (X), TAN (X), ATN (X) si musíme uvedomiť v akých uhlových mierach je vyjadrená hodnota X. Použitím príkazov

RAD prepne režim RADIANY

DEG prepne režim STUPNE

Po spustení programu príkazom RUN ako aj po použití povelu CLEAR sa implicitne nastavuje režim chápania argumentu X v RADIÁNOCH.

Je samozrejmé, že namiesto čísla X sa môže uvádzať aj aritmetický výraz. Ešte nám zostáva vysvetliť, v akom poradí mikropočítač spracováva zadané výrazy.

Hierarchia je nasledovná:

1. exponenciálne výrazy
2. znamienka plus alebo minus
3. násobenie alebo delenie
4. sčítanie alebo odčítanie
5. relačné operátory
6. lôgické operátory NOT, AND, OR

Ak majú dva operátory rovnakú úroveň spracovania, postupuje mikropočítač smerom zľava doprava. Napríklad výraz

$2\phi - A + B * C ^ D$

bude mikropočítačom spracovaný v tomto poradí:

1. vypočíta $2\phi - A$
2. vypočíta $C ^ D$
3. potom vynásobí B výsledkom získaným v kroku 2
4. výsledok získaný v kroku 1 pripočíta k výsledku získanom v kroku 3

Ak ste si neistí hierarchiou spracovania výrazov, používajte zátvorky! A teraz pristúpime ku komunikácii s mikropočítačom v dialógovom režime.

9. KALKULÁTOROVÝ REŽIM

Kalkulátorový režim práce - ináč nazývaný tiež priamy režim - spočíva v zadani prikazu, výpočtu aritmetického, lôgického alebo relačného výrazu, ktorý počítač ihneď prevedie. Výsledok /pokiaľ je žiadany/ zobrazí v dialógovom riadku alebo v pracovnej časti obrazovky.

Mikropočítač pritom kontroluje váš príkaz a prípadné zistené chyby semantického alebo syntaktického charakteru vám oznamí v dialógovom riadku. Zoznam všetkých správ, ktoré môže odoslať je uvedený vo zvláštnej kapitole na konci výkladu.

Je to preto, aby ste ich mali vedla seba a tým zabránili možnému omylu.

Pripomeňme si, že mikropočítač po odoslani príkazového riadku BEZ ČÍSLA chápe tento ako príkaz pre KALKULÁTOROVÝ režim a ihneď ho výkoná.

Ak príkaz začína ČÍSLOM chápe tento ako programový riadok a nevýkoná nič. Na záver si precvičte niekoľko príkladov na kalkulatorový režim.

Zadanie z klávesnice	Výsledok
CONT + CLS	zmaže obrazovku
? 2 0 0 0 0 0 0 EOL	2 E + Ø6
PRINT "SKUSKA" EOL	SKUSKA - v pracovnej časti OK - v dialog. riadku
A 3 3 5 EOL	OK - v dialog. riadku
PRINT A EOL	335 - v prac. časti
? (2 2 + 1 0) / 2 EOL	16 - v dialog. riadku
PRINT 1 0 / 0 EOL	+++ DV BY ZERO +++

Posledný príklad ukazuje zadanie, u ktorého mikropočítač zistil a oznámil semantickú chybu. Ide o zakázané delenie nulou. Túto správu mikropočítač napiše do dialógového riadku s krátkym akustickým návestím.

Ak vás tieto riadky presvedčili, že ovládate postup, pokúste sa samostatne zadávať rozličné výpočty a realizovať ich, aby ste získali potrebnú rutinu a odhalili matematické možnosti mikropočítača.

Postupne zistíte, že v priamom režime sa len veľmi ťažko realizujú zložitejšie, viacnásobné príkazy. Je to preto, lebo musíte vykonať niekoľko príkazov za sebou, kým sa dopracujete k žiadanejmu výsledku. Túto nevýhodu odstraňuje programový režim práce mikropočítača.

10. PROGRAMOVÝ REŽIM

Mikropočítač vám dáva možnosť zoradiť jednotlivé kroky výpočtov za sebou určitým spôsobom. Takému usporiadaniu myšlienkového postupu predpísaným spôsobom hovoríme PROGRAM. Na program má mikropočítač vyhradenú určitú časť svojej pamäte. Ak chcete vedieť, akú má práve teraz voľnú kapacitu, odošlite tento prikaz:

PRERERA EOL

a mikropočítač vám v dialógovom riadku napiše jej veľkosť v byte.

Aby bola možnosť evidovať jednotlivé príkazy, budeme si postupne definovať nové pojmy.

Najmenšou programovou jednotkou je jeden programový riadok. Na jeho evidenciu v programe je potrebné, aby ste mu priradili číslo. To znamená, ak napišete do dialógového riadku príkaz, ktorý sa začína ČÍSLOM, mikropočítač ho zaeviduje do svojej pamäte ako programový riadok a na vytvorenie programu, jeho editovanie, prípadné spustenie má mikropočítač rezervované tieto príkazy:

Príkaz	Význam
NEW	Zruší starý program a inicializuje počítač na počiatočný stav.
L LIST n	Vypíše obsah programu uloženého v pamäti do pracovnej časti obrazovky. Pri zadani hodnoty n = číslo riadku, začína výpis od tohto riadku.
LLIST n	Výpis obsahu uloženého programu po jednotlivých riadkoch do dialógového riadku za účelom jeho prípadného editovania. Pri zadani hodnoty n = číslo riadku, začína výpis od tohto riadku.

AUTO n Prikaz umožňujúci automatické čislovanie riadkov. Ak n = číslo počiatočného riadku, bude čislovanie prebiehať od daného riadku a s prirastkom 10.

Čislovanie riadkov môže byť v rozsahu 0 až 32767, pričom prirastok môže byť lubovoľný. Doporučujeme vám používať čislovanie riadkov s desiatkovým prirastkom, aby ste dodatočne mohli vkladať ďalšie riadky. Zrušenie riadku je možné urobiť jednoducho - stačí vložiť jeho číslo a potvrdiť klávesou EOL.

Napríklad: **1 5 EOL** zruší riadok s poradovým číslom 15

Ak ste porozumeli predchádzajúcemu, môžeme pristúpiť k pisaniu krátkych programov.

Napríklad: Výpočet plochy kruhu podľa vzorca:

$$F = \text{PI} * r * r \quad \text{kde PI} = 3,1415$$

Na príklade si ukážeme funkciu automatického čislovania riadkov. V pracovnej časti obrazovky sa vám bude po každom odoslani vypisovať programové riadky aj s priradenými číslami.

Pričaz	Odpoveď mikropočítača
N E W EOL	OK_
CONT + CLS	zmaže obrazovku
L I S T EOL	OK_
A U T O EOL	10_
P R I N T " P L O C H A " EOL	20_
P I = 3 . 1 4 1 5 EOL	30_
D I S P " P O L O M E R E " EOL	40_
I N P U T R EOL	50_
F = P I * R * R EOL	60_
P R I N T R F EOL	70_
E N D EOL	80_

Všimnite si programové riadky 30 a 40. V riadku 30 je uvedený príkaz DISP, ktorý má podobnú funkciu ako príkaz PRINT, ale mikropočítač ho vypisuje do dialógového riadku. Jeho úlohou v našom prípade je upozorniť na to, že máte vložiť vstupnú hodnotu.

Teraz vykonaním príkazu **GCLEAR EOL** môžete zma-
zať obrazovku. Ak to skúsite, zistíte, že príkaz sa zaradil
ako programový riadok. To preto, že nie je zrušený príkaz
AUTO a nemôžeme prejsť do priameho režimu. Urobíme to vyma-
zaním dialógového riadku **CONT + CLR** a potom príkaz
GCLEAR zmaže obrazovku.

Vypísanie nášho programu si môžeme urobiť už známym
priazom **LIST EOL**

Na obrazovke uvidíme náš program:

```
10 PRINT "PLOCHA"  
20 PI = 3.1415  
30 DISP "POLOMER="  
40 INPUT R  
50 F = PI * R * R  
60 PRINT R, F  
70 END
```

Teraz si môžeme precvičiť editovanie programového riadku
číslo 10. Urobíme to priazom LLIST:

```
LLIST EOL 10 PRINT "PLOCHA"  
CONT + ENDL SPACE SPACE KRUHU" EOL OK
```

a v pracovnej časti obrazovky môžeme vidieť zeditovaný ria-
dek č. 10:

```
10 PRINT "PLOCHA KRUHU"
```

Alebo pomocou príkazu LLIST EOL dostaneme ten istý výsledok
do dialógového riadku. Ak je vám jasný spôsob vkladania
a editovania programových riadkov, vyskúšame si príkaz na
spustenie programu:

```
RUN EOL
```

Bezprostredne po tomto prikaze začne mikropočítač interpretovať program a vypíše text:

V pracovnej časti obrazovky
V dialógovom riadku

PLOCHA KRUHU
POLOMER =

Teraz mikropočítač čaká na vloženie číselného údaja hodnoty POLOMER. Napište číslu 1 a riadok ukončte EOL. Ak ste náhodou vložili iný znak ako číslu, mikropočítač ho neprijme, ale bude čakať na číselný údaj. Ak by ste chceli opustiť tento stav, stačí súčasne stlačiť klávesy **STOP** + **EOL** a mikropočítač vám v dialógovom riadku vypíše správu

+++ STOP IN LINE 40 +++

Pokračovanie v takto zastavenom programe môžeme dosiahnuť vykonaním prikazu

CONT **EOL** alebo **GOTO 40** **EOL**

Pripadne máte ešte možnosť pozastavenia bežiaceho programu klávesou **SHIFT** - pokiaľ túto klávesu držíte stlačenú, program sa nevykonáva, pri uvoľnení pokračuje ďalej.

Ak ste vložili číselnú hodnotu premennej $r = 1$, umožní li ste počítaču pokračovanie v interpretácii programu riadkom 50, kde má urobiť výpočet premennej F. V riadku 60 žiadate vytlačiť tento výsledok do pracovnej časti obrazovky. Z mikropočítača dostávate potom nasledovný výsledok:

V pracovnej časti obrazovky 1 3,1415
V dialógovom riadku OK

Program môžete odštartovať aj od určitého čísla riadku. Skúste vykonať nasledujúcu sekvenciu príkazov.

CONT + **CLS** **EOL**
GOTO 20 **EOL**

a porozmýšľajte nad chovaním mikropočítača. Pre úplnosť

priazov súvisiacich so zostavovaním programov treba ešte uviesť príkazy pre prácu s magnetofónom pod riadením interpretu BASIC-G, ktorý používa odlišné príkazy ako boli uvedené v časti 6. Sú to nasledovné príkazy:

SAVE n;"komentár"

Uloženie programu začínajúceho sa prvým riadkom na magnetofón, kde parameter n = poradové číslo programu v rozsahu 0 - 99. Komentár nie je povinný, ak ho však napišete musí začínať úvodzovkami - ináč nesmie obsahovať klúčové slovo jazyka BASIC-G.

LOAD n

Vyhľadanie a uloženie programu z pásky magnetofónu s poradovým číslom "n" do programovej pamäte BASICu.

CHECK n

Kontrola správnosti záznamu z pásky magnetofónu za účelom kontroly alebo vyhľadania konca záznamu pre možnosť ďalšieho zápisu. Tento príkaz nemení obsah programovej pamäte.

U príkazov LOAD a CHECK, pokiaľ dojde k prečítaniu hlavičky, ktorej číslo nezodpovedá požadovanému, vypisujú sa prečítané hlavičky programov do pracovnej časti obrazovky, takže zadánim čísla 99 /pokiaľ záznam s takýmto číslom na páske nemáte/ dojde k vypísaniu obsahu celej pásky.

Vráťme sa ešte k príkazu SAVE. Ak komentár použitý v príkaze začína znakom "*" /hviezdička - jeho meno potom môže byť len sedem znakov/, takto zaznamenaný program sa po nahrati príkazom LOAD n ihneď spustí a nie je ho možné zastaviť klávesou **STOP**. Ak program prerušíte klávesou **RST** a odštartujete BASIC odznova, program sa vymaže.

Napríklad: **SAVE 21"MEMEN"**

Doteraz by ste podľa prebraného výkladu mali ovládať príkazy pre prácu v priamom i programovom režime. Mali by ste hravo zvládnuť napísanie programu jednoduchšieho typu, jeho editovanie, nahranie na magnetofónovú pásku. Viete už používať príkazy na odštartovanie, prerušenie a zastavenie programu. Povieťe si ešte o príkaze, ktorý sa dá veľmi výhodne používať pri ladení programu. Ladenie programu je proces, pri ktorom odhalujeme skryté chyby. Za týmto účelom je potrebné mať možnosť programovo zastaviť interpretáciu programu na určitom riadku a znova pokračovať po preskúmaní napr. hodnôt premenných, vetvení a pod.

Vložte do nášho programu tento príkazový riadok:

55 STOP EOL

a odštartujte tento program už známym príkazom **RUN EOL**. Vidíme, že po zadaní hodnoty premennej POLOMER vypíše mikropočítač do dialógového riadku správu

+++ STOP AT LINE 55 +++

Preskúmajte si napr. obsah premennej F - príkaz na pokračovanie v programe už poznáte.

V ďalšom si osvojite ďalšie príkazy používané v samotnom programe. Ide o štandardné príkazy používajúce väčšina variant interpretov BASIC. Príkazy pre grafiku si ponecháme ako samostatnú časť.

Pričaz	Význam
BEEP	Akustické návestie
DATA	Označenie dát alebo reťazcových znakov
DEF FNC	Definovanie funkcií užívateľom
DEG	Prepne na mod STUPNE

DIM	Deklarovanie reťazcových a dátových polí
FOR-TO-NEXT (STEP)	Tvorenie programových slučiek
DISP	Zobrazenie výrazu do dialógového riadku
GCLEAR	Vymazanie obrazovky
GOSUB	Volanie podprogramu
RAD	Prepne na mod RADIANY
RETURN	Ukončenie podprogramu
GOTO	Skok na programový riadok
IF-THEN	Podmienené vykonanie príkazu THEN
INPUT	Zadanie vstupných údajov z klávesnice
LET	Priľaďovací príkaz
PAUSE	Časové zdržanie vykonávaného programu
POKE	Zápis dátového údaja do pamäte
APOKE	Zápis adresného údaja do pamäte
PEEK	Načítanie dátového údaja z pamäte
APEEK	Načítanie adresného údaja z pamäte
READ , DATA	Čítanie údajov do premenných z programu
RESTORE	Nastavenie ukazovateľa dátového bloku
STOP	Zastavenie interpretácie programu
ON ERR	Ošetrenie chýb v programe
ON	Programový prepínač skokov a volania podprogramov
ADR (X)	Vracia adresu, na ktorej je uložená premenná X
INK	Prog. prepínač inverzie tlače v príkaze PRINT
PRINT	Zobrazenie výrazov do pracovnej časti obrazovky
AT	Nastavuje miesto výpisu tlače v príkaze PRINT
TNKEY	Načítanie hodnoty KEY programových klávesov

TAB (I)	Tabulátor v príkaze PRINT
FRE (A)	Voľná pamäťová oblasť pre uloženie programu
SPC (I)	Vkladanie I medzier na výstupné zariadenie
BIT A,B	Určenie hodnoty bitu A na váhe B
USR (I)	Volanie strojového programu na adresu I
CODE X\$	Volanie strojového programu zapisaného ako reťazec ASCII v premennej X

Dosiaľ nebolo vysvetlené, aké možnosti má mikropočítač na uskladnenie premenných, prípadne reťazcov. Všeobecne sa tieto skupiny dát nazývajú "pole". S jednotlivými prvками pola sa dá pracovať ako s premennými. Všeobecne pole sa označuje dvoma znakmi. Prvý znak musí byť vždy písmeno, druhý znak môže byť písmeno alebo číslica.

Napríklad: A1 (15)	Znamená 15-ty prvk jednorozmerného pola A1
AA (10,10)	Znamená prvek z dvojrozmerného pola AA. Pozícia prvku je na súradničiach 10, 10
AB (10,10,10)	Znamená prvek z trojrozmerného pola AB. Pozícia prvku je na súradničiach 10, 10, 10

Zapamätajme si, že pre väčší počet prvkov používaného pola je potrebné, aby ste deklarovali dané pole - čiže, aby ste určili jeho veľkosť prikazom DIM.

Uvedené sa vzťahovalo výlučne na číselné polia. Ak chceťte používať pole, ktoré bude obsahovať reťazce ASCII znakov, rozšíri sa značka pola o znak "\$".

Reťazce sú znaky ASCII, ktoré sú písané medzi dvoma úvodzovkami. Ich dĺžka môže byť maximálne 255 znakov. Reťazcové premenné môžu byť označované jednoznakové /napr. A\$/ alebo dvojznakové /napr. AA\$, A2\$/ Reťazcové premenné môžu byť deklarované aj v poliach, pričom maximálna dimenzia je 3.

POZOR! Nezabudnite rozlišovať druh premenných - A, A1, F\$, R1\$. Pri práci s reťazcami používame ešte tieto funkcie:

Funkcia	Význam
ASC (X\$)	Dekadickej hodnota prvého znaku reťazca X\$
CHR\$ (A)	Prevod dekadickej hodnoty A na hodnotu ASCII
FRE (X\$)	Zistenie voľného miesta na uloženie reťazcov
LEFT\$ (X\$, I)	Načítanie zľava I-znakov z reťazca X\$
RIGHT\$ (X\$, I)	Načítanie sprava I-znakov z reťazca X\$
MID\$ (X\$, I, K)	Načítanie časti reťazca od znaku na pozícii I a dĺžky K
LEN (X\$)	Určenie dĺžky reťazca X\$
STR\$ (X)	Premena čísla X na reťazec
VAL (X\$)	Premena reťazca X\$ na číslo
HEX\$ (X)	Prevod čísla X na štvormiestny HEXA reťazec
ADR (X\$)	Vracia adresu pamäte, na ktorej je uložená premenná X\$

Uviedli sme prehľad príkazov a povelov jazyka BASIC-G. Pre prehľadnosť vám vysvetlime funkciu jednotlivých príkazov podrobne, s presným písaním syntaxu, ich činnosťou a príkladmi použitia. Vyskúšajte si na základe získaných poznatkov uvádzané príklady.

10.1 PREHĽAD RIADIACICH PRÍKAZOV

NEW	Pričaz nuluje všetky vložené príkazové riadky programu a premenné
LIST n	Pričaz robí výpis uloženého programu na obrazovku. Ak je dané číslo riadku "n", výpis bude začínať riadkom s uvedeným číslom

RUN n	Priaz odštartuje program. Predtým však dojde k nulovaniu premenných, polí a ukazovateľov dát. Ak je uvedené číslo riadku "n", štart programu začína riadkom s uvedeným číslom.
GOTO n	Priaz má podobnú funkciu ako priaz RUN, ale nedojde k nulovaniu premenných, polí a ukazovateľov dát.
CONT	Priaz zabezpečí pokračovanie v interpretovaní programu, ktorý bol zastavený priazom STOP v programe alebo stlačením tlačítka STOP na klávesnici.
LLIST n	Priaz používaný k editovaniu programového riadku v dialógovom riadku. Ak nie je uvedené číslo riadku "n", vypíše sa riadok s najnižším poradovým číslom.
SAVE n	Priaz uloží na magnetofónovú pásku program s poradovým číslom n
LOAD n	Priaz načíta z magnetofónu záznam s poradovým číslom n do operačnej pamäte počítača
CHECK n	Priaz urobí kontrolu záznamu s poradovým číslom n na kontrolnu sumu
AUTO n	Priaz pre automatické číslovanie programových riadkov s inkrementom 10, od riadku číslo n.
?	Krátky priaz na výpis výrazu do dialógového riadku. Po výpise sa dialógový riadok vymaže stlačením ľubovoľnej klávesy.

10.2 PREHĽAD ŠTANDARDNÝCH PRÍKAZOV

AT	Syntax: AT s,r r ... číslo riadku Ø - 26 s ... číslo stĺpca Ø - 47 Príkaz slúži na nastavenie miesta výpisu na obrazovke, kam sa vypíše nasledujúci text v príkaze PRINT. Ak sa príkaz AT vyskytuje ako posledný v príkaze PRINT, tento príkaz sa chápe ako neukončený - akoby bola na jeho konci bodkočiarka. PRÍKLAD: 1Ø FOR I=Ø TO 26 2Ø PRINT AT I,2*I; "*" 3Ø NEXT I
ADR	Syntax: ADR (n) Príkaz vráti adresu, na ktorej je definovaná číselná alebo reťazcová premenná fyzicky uložená v pamäti počítača. PRÍKLAD: 1Ø LET A=156 2Ø LET B\$="adresa" 3Ø X=ADR (A): Y=ADR (B\$) 4Ø PRINT X,Y
BEEP	Syntax: BEEP Príkaz na generovanie krátkeho akustického návestia PRÍKLAD: 1Ø BEEP
BIT	Syntax: BIT a,b a ... číslicová premenná v rozsahu Ø - 255 b ... váha bitu v rozsahu Ø - 7 Výsledkom funkcie je priradenie hodnoty definovaného byte "a" premennej na váhe "b". PRÍKLAD: 1Ø A = 129 2Ø DØ = BIT A,Ø 3Ø D7 = BIT A,7 4Ø PRINT DØ, D7

DATA

Syntax: DATA d1,d2, ...,dn

Označenie dát alebo reťazcových znakov pre ich čítanie prikazom READ. Jednotlivé dáta sú oddelené čiarkou. Prikaz DATA môže byť umiestnený v libovoľnej časti programu.

PRÍKLADE: 1 ϕ READ A, B\$

2 ϕ PRINT A

3 ϕ PRINT B\$

4 ϕ END

5 ϕ DATA 356

6 ϕ DATA "reťazec"

DEF

Syntax: DEF FNC a (X) = výraz

a ... písmenový znak užívateľskej premennej
X ... jednoznaková premenná, ktorá môže byť
aj výraz

Priaz sa používa k definovaniu funkcií užívateľom, aby nemusel stále sa opakujući výpočet programovať pri každom použití iných parametrov.

PRÍKLADE: 1 ϕ DEF FNC A (X) = EXP (X^2)

2 ϕ INPUT X

3 ϕ PRINT FNC A (X)

4 ϕ GOTO 2 ϕ

DIM

Syntax: DIM p /rozmer 1/, p /rozmer/,

p\$ /rozmer/

p ... označenie číselného poľa

p\$.. označenie reťazcového poľa

Priaz sa používa k deklarovaniu reťazcových a dátových polí.

PRÍKLADE: 1 ϕ DIM A(1 ϕ , 2 ϕ), K(2,3,5), L\$(8,2,2)

DSAVE

Syntax: DSAVE n, m, "komentár"

n ... poradové číslo záznamu

m ... premenná označujúca číselné alebo reťazcové pole

Príkaz sa používa pre zaznamenanie prvkov číselného alebo reťazcového poľa na magnetofónovú kazetu. V prípade, že príkaz obsahuje nedimenzované pole, hlási interpreter chybu.

PRÍKLAD: 1 \emptyset DIM A\$(1 \emptyset)

2 \emptyset ? "zapni magnetofón"

3 \emptyset DSAVE 12, A\$(\emptyset), "data A\$"

4 \emptyset ? "mg stop!"

DLOAD

Syntax: DLOAD n, m

n, m ... sú výrazy ako v príkaze DSAVE

Príkaz umožňuje napĺňanie číselného alebo reťazcového poľa, ktoré musí byť nadimenzované, z magnetofónu. Pri príkaze sa kontroluje dimenzovaný priestor poľa, ktoré je zaznamenané na páske a v prípade nesúhlasu hlási chybu.

PRÍKLAD: 1 \emptyset DIM A(1 \emptyset ,1 \emptyset)

2 \emptyset FOR I=1 TO 5

3 \emptyset DLOAD A, A(\emptyset , \emptyset)

4 \emptyset NEXT I

FOR TO NEXT STEP

Syntax: FOR a = n TO m STEP z

a ... riadiace premenná

n ... počiatočná hodnota cyklu

m ... konečná hodnota cyklu

z ... krok cyklu

Príkaz slúži na realizáciu cyklov. Každý cyklus musí byť ukončený príkazom NEXT. Na príklade je ukázané vnorenie dvoch cyklov.

PRÍKLAD: 1 \emptyset FOR A=1 TO 5

2 \emptyset FOR B=1 TO 3

3 \emptyset PRINT B;

4 \emptyset NEXT B

5 \emptyset PRINT

6 \emptyset NEXT A

DEG	Syntax: DEG Pričaz prepne zadávanie goniometrických funkcií v STUPŇOCH PRÍKLAD: 1Ø DEG
DISP	Syntax: DISP výraz Pričaz sa používa k zobrazeniu výrazu do dialógového riadku. PRÍKLAD: 1Ø DISP SIN (X), 123*2 2Ø DISP "vstup"
GCLEAR	Syntax: GCLEAR Pričaz vymaže celú obrazovku. PRÍKLAD: 1Ø PRINT "ABCDEF" 2Ø GCLEAR
GOSUB	Syntax: GOSUB výraz Pričaz slúži na volanie podprogramov, pričom "výraz" môže byť číslo riadku programu alebo výraz, ktorého hodnota je z intervalu Ø až 32767. Pričaz môže byť použitý až do hĺbky 6 volaní. Podprogram musí byť ukončený pričazom RETURN. PRÍKLAD: 1Ø GOSUB 1ØØ 2Ø GOSUB X*6
GOTO	Syntax: GOTO výraz Pričaz slúži na realizáciu skoku v programe s možnosťami ako v pričaze GOSUB. PRÍKLAD: 1Ø GOTO 3Ø 2Ø GOTO X/3
IF THEN	Syntax: IF výraz THEN k k ... číslo riadku alebo pričaz Pričaz pre podmienené vykonanie pričazu THEN na základe pravdivosti výrazu. Pokiaľ je výraz nepravdivý, pokračuje interpretovanie programu na nasledovnom riadku. Výraz môže byť aritmetický alebo reťazcový.

PRÍKLAD: 1 \emptyset IF X+Y= \emptyset THEN 15 \emptyset
2 \emptyset IF A>B OR C<D AND E=F THEN STOP
3 \emptyset IF A\$="ano" GOTO 33

INK

Syntax: INK (n)

n ... \emptyset = podklad čierny, písmo biele

n ... 4= podklad biely písmo čierne

Priaz slúži na ovládanie jasu podkladu
a písma výpisu.

PRÍKLAD: 1 \emptyset PRINT INK(4); "BASIC"; INK(\emptyset); "V2.1"

INPUT

Syntax: INPUT zoznam premenných

Priaz slúži na zadávanie dát z klávesnice.

Ukončenie zadávania dát je dané stlačením
EOL. Zoznam vstupných premenných, ktorým bu-
dú vstupujúce dáta priradené, sú oddelené
čiarkou.

PRÍKLAD: 1 \emptyset INPUT A\$

2 \emptyset INPUT A(1,1), C, A1(2,2)

INKEY

Syntax: INKEY

Priaz priradi hodnotu stlačeného programové-
ho klúča v rozsahu \emptyset - 11 podľa toho, o ktorý
klúč ide. V prípade, že nie je žiadny stlače-
ný, generuje sa hodnota 255.

PRÍKLAD: 1 \emptyset A= INKEY

2 \emptyset PRINT A

3 \emptyset GOTO 1 \emptyset

LET

Syntax: LET a = výraz

a ... označenie premennej, ktorej hodnota
"výraz" bude priradená

Priaz slúži na priradenie hodnoty výrazu
danej premennej. Reťazcové výrazy môžu byť
priradené reťazcovej premennej a aritmetické
výrazy k aritmetickej premennej. Priaz LET
nie je povinný.

PRÍKLAD: 1 ϕ LET B=C/2 + 127
2 ϕ LET A\$="premenná"
3 ϕ C=44 + B

NEXT

Syntax: NEXT premenná
Pričaz slúži na uzavorenie smyčky príkazu
FOR s danou riadiacou premennou.
PRÍKLAD: 1 ϕ FOR I=1 TO 1 ϕ
2 ϕ BEEP
3 ϕ NEXT T

ON

Syntax: ON výraz GOSUB zoznam príkazových
riadkov
ON výraz GOTO zoznam príkazových
riadkov
Na základe hodnoty "výraz" (1-255) sa pre-
vedie vykonanie príkazu na príkazový riadok,
ktorý sa nachádza s daným poradovým číslom
v zozname príkazových riadkov. Ak tieto po-
lohy zoznam neobsahuje, bude sa interpretovať
na nasledujúcom riadku. Pre výhodnenie vý-
razu sa berie len celočíselná hodnota.
PRÍKLAD:..
1 ϕ ON X-6 GOTO 5 ϕ , 3 ϕ , 8 ϕ ak X-6=1 GOTO 5 ϕ
ak X-6=2 GOTO 3 ϕ
ak X-6=3 GOTO 8 ϕ
ináč pokračuje

ON ERR

Syntax: ON ERR príkaz
Pričaz umožňuje potlačiť hlásenie chýb a o-
šetrenie chýb v užívateľskom programe. To
znamená, že v prípade chyby sa vykonávanie
programu nepreruší, ale pokračuje na urče-
nom mieste.

PRÍKLAD: 1 ϕ ON ERR GOTO 5 ϕ
2 ϕ DISP "zadaj X"
3 ϕ PRINT X, 1/(X*X-9)
4 ϕ GOTO 1 ϕ
5 ϕ DISP "chyba zadania"
6 ϕ ON ERR GOTO 5 ϕ : GOTO 2 ϕ

PAUSE

Syntax: PAUSE výraz

Pričaz slúži na pozastavenie vykonávania programu, kde "výraz" môže byť celá hodnota v rozsahu 0 - 255. Časový inkrement je 0.1 sek. a čakanie je možné prerušiť stlačením klávesy ~~SPACE~~, ~~BEEP~~

PRÍKLAD: 10 PAUSE 20

PRINT

Syntax: PRINT zoznam výrazov

Pričaz slúži na vytlačenie výsledkov alebo hodnôt do pracovnej časti obrazovky. Zoznam výrazov je zoznam číselných premenných prvkov poľa, reťazcové premenné, výrazy alebo reťazce ASCII, ktoré sú navzájom oddelené oddelovačom.

Riadok PRINTu je rozdelený na štyri štrnásťznakové zóny, čiarka (,) medzi výrazmi predpisuje prechod na nasledujúcu zónu. Ak oddeľovačom je bodkočiarka (;) ide o zhustenú tlač veľa sa.

PRÍKLAD: 10 X = 3

20 PRINT X, X*2, X^2, X^X

30 PRINT X; X*2; X^2; X^X

RAD

Syntax: RAD

Pričaz prepne zadávanie hodnôt goniometrických funkcií v RADIÁNOCH. Tento mod je nastavovaný implicitne po zapnutí počítača.

PRÍKLAD: 10 RAD

READ

Syntax: READ zoznam premenných

Pričaz slúži na priradenie hodnôt daným premenným alebo poliam definovaným v zozname a oddelenými čiarkou. Typy premenných musia korešpondovať s typmi uvedenými v prikaze DATA.

PRÍKLAD: 10 READ A,A\$

RESTORE

Syntax: RESTORE príkazový riadok

Príkaz slúži na nastavenie dátového bloku, z ktorého sa budú priraďovať hodnoty premenným v príkaze READ. Ak neuvedieme číslo riadku, začína sa čítať prvým dátovým blokom.

PRÍKLAD: 1Ø RESTORE 5Ø

2Ø READ A, B, C\$

5Ø DATA 155, 11, "STRING"

RETURN

Syntax: RETURN

Príkaz slúži pre výstup z podprogramu, ktorý je volaný z hlavného programu príkazom GOSUB.

PRÍKLAD: 1Ø PRINT "skúška"

2Ø GOSUB 4Ø

3Ø STOP

4Ø PRINT "abcdefgh"

5Ø RETURN

SPC

Syntax: SPC (výraz)

Príkaz slúži na vytlačenie počtu prázdných znakov /medzier/ v príkaze PRINT medzi výpisom jednotlivých položiek zoznamu a kde výraz je celočíselná hodnota.

PRÍKLAD: 1Ø PRINT A, SPC (22), B

STOP

Syntax: STOP

Príkaz sa používa k zastaveniu interpretovaného programu. Po jeho vykonaní sa v dialógovom riadku vypíše správa +++STOP AT LINE n +++, kde n=číslo riadku, na ktorom k zastaveniu došlo a systém čaká na príkazy z klávesnice. Pokračovanie takto zastaveného programu docílime príkazom CONT.

PRÍKLAD: 1Ø PRINT "abcdefgh"

2Ø STOP

3Ø GOTO 1Ø

TAB

Syntax: TAB (výraz)

Príkaz slúži na nastavenie začiatku tlače výrazu v riadku, kde výraz je celočíselná hodnota. Riadok má dĺžku 48 znakov.

PRÍKLAD: 10 PRINT TAB(5),A, TAB(20),B;D

10.3 PRÍKAZY PRE SYSTÉM 8080

Táto skupina príkazov je určená hlavne pre tých, ktorí majú hlbšie znalosti o systéme s mikroprocesorom MHB 8080 A a s ním súvisiacich podporných obvodoch. Pri nedostatočných znalostach sa vám ľahko môže stať, že sa vám podarí prepísat dôležitý údaj v pamäti počítača a dojde k "zrúteniu" systému.

Máte možnosť používať zadávanie adresy v hexadecimálnej sústave, ak číslo udávajúce adresu označíte apostrofom ('C1B4), alebo v desiatkovej sústave bez apostrofu (22456).

APEEK

Syntax: APEEK (a)

- Výsledkom funkcie je prečítaná hodnota dvoch po sebe nasledujúcich byte v pamäti, chápana ako 16 - bitové číslo so znamienkom počinajúc adresou (a) uvedenou v príkaze.

PRÍKLAD: 10 A = APEEK ("8000")

20 B = APEEK (22540)

30 PRINT A, B

APOKE

Syntax: APOKE a,d

Výsledkom funkcie je zápis 16-bitovej konštanty na dve po sebe nasledujúce pamäťové miesta počinajúc adresou uvedenou v príkaze.

PRÍKLAD: 10 APOKE "0010, "3F3F

20 APOKE 49028, 65000

CODE

Syntax: CODE a\$,b\$

Príkaz vykoná interpretáciu programu napísaného v strojovom kóde mikroprocesoru 8080, ktorý je uložený ako reťazec ASCII znakov v reťazcových premenných definovaných v príkaze a oddelenými čiarkou. Dĺžka reťazca je

obmedzená na 40 znakov, pričom posledný znak musí byť inštrukcia RET (C9).

PRÍKLAD: 10 A\$ = "3E413234C1CD0085C9"
20 CODE A\$, A\$

PEEK

Syntax: PEEK (a)

Príkaz načíta osembitovú hodnotu z pamäťového miesta udaného v adrese, pričom adresa môže byť aj výraz.

PRÍKLAD: 10 A = PEEK (64577-32445)
20 PRINT A

POKE

Syntax: POKE a,d1,d2, ...dn

Príkaz zapíše osembitové hodnoty "d" do po sebe nasledujúcich pamäťových miest od adresy "a".

PRÍKLAD: 10 POKE "7000, 10,20,255
20 POKE 49028, 255

USR

Syntax: USR (v)

Príkaz vykoná program v strojovom kóde začínajúci na pamäťovom mieste danom hodnotou "v", ktorá môže byť v rozsahu 0 - 32767.

Parameter z tejto executivy je možné preniesť do programu BASIC prostredníctvom definovanej premennej. Doporučuje sa používanie pamäťovej oblasti 7000 - 7E00 H.

PRÍKLAD: 10 A = USR (20000)

10.4 PRÍKAZY PRE VSTUPNO - VÝSTUPNÉ ZARIADENIA

Táto skupina príkazov pre vás nemá význam, nakoľko je určená pre spoluprácu s doskou interfejsov (napr. z PMD 85) pripojenou na aplikačný systémový konektor. Pre úplnosť popisu možnosti jazyka BASIC-G ich však uvedieme. V prípade možnosti a záujmu doporučujeme príručku pre počítač PMD,85 s názvom "OUTPUT - ENTER".

OUTPUT	Syntax: OUTPUT kr; zoznam výrazov Pričaz zabezpečí výstup hodnoty výrazov na príslušný register "r" adresovaného kanálu "k".
CONTROL	Syntax: CONTROL k,a; výraz Pričaz zabezpečí zapísanie hodnoty "výraz" do registru s adresou "a" na kanáli číslo "k".
ENTER	Syntax: ENTER kr; zoznam premenných Pričaz načíta a priradí premenným definovaným v pričaze hodnoty registru "r" na kanáli číslo "k".
STATUS	Syntax: STATUS k,a Pričaz načíta hodnotu registru "a" definovaného kanálu "k"
IN	Syntax: IN x Pričaz prečíta hodnotu zo vstupného zariadenia na adrese "x".
OUT	Syntax: OUT x,n Pričaz zapíše hodnotu "n" na výstupné zariadenie s adresou "x".

10.5 PRÍKLADY NA PRECVIČENIE ZÍSKANÝCH ZNALOSTÍ

V tejto časti si uvedieme niekoľko príkladov programov, ktoré si preštudujte a vyskúšajte ich činnosť. Sú vybraté také, aby ste si mohli odskúšať všetky pričazy a funkcie jazyka BASIC-G okrem pričazov pre prácu s grafikou, ktoré si ponecháme ako samostatnú časť. Za každým príkladom uvedieme niekoľko úloh na modifikáciu programu, aby ste sa pocvičili v používaní jednotlivých pričazov.

Skúste si sami neskôr zostaviť program podľa vašich predstáv s využitím pričazov, ktoré dobre ovládate. A ešte rada.

Ak vám program nebude pracovať podľa vašich predstáv, nehľajte chybu v počítači, aj keby tomu všetko nasvedčovalo. Spravidla je chyba vo vašom programe - logická alebo syntaktická.

Priklad č.1

```
10 REM "sportka"  
20 FOR I=1 TO 6  
30 PRINT INT(48 * RND(1)+1);  
40 NEXT I
```

Doplňte program tak, aby nedochádzalo ku generovaniu rovnakých čísel.

Priklad č.2

```
10 REM "tabuľka funkcia"  
20 T=8: DEF FNC A(X)=EXP(X/T)  
30 FOR X=0 TO 100 STEP 2  
40 PRINT "X= ";X;"A(X)= ";FNC A(X)  
50 NEXT X
```

Všimnite si písanie viac príkazov na jeden programový riadok. Oddeľovačom je dvojbodka. Napište program pre definovanie goniometrickej funkcie.

Priklad č.3

```
10 REM "súčet čísel zadaných ako dát"  
20 FOR I=1 TO 10  
30 READ N  
40 S=S + N  
50 NEXT I  
60 PRINT "SÚČET JE "; S  
70 DATA 2,5,8,9,3,44,2,3,99,63,57
```

Doplňte program tak, aby sa na obrazovke vypisovali aj

prečítané dát a potom ho upravte, aby sa dát zadávali z klávesnice.

Priklad č.4

```
10 REM "vety"
20 A$= "MA": B$= "TA": C$= " "
30 M$= A$ + A$ + C$ + B$
40 N$= B$ + B$ + C$ + A$ + B$
50 P$= RIGHT$(A$,2)
60 PRINT M$,N$,P$
```

Vytvorte pomocou ďalších prikazov z daných reťazcov vety, ktoré dávajú zmysel, prípadne si doplňte program o ďalšie reťazce.

Priklad č.5

```
10 REM "obloha"
20 FOR N=1 TO 100
30 Y=INT(RND(5)*46+1)
40 X=INT(RND(2)*23+1)
50 PRINT AT Y,X;"*"
60 NEXT N
```

Všimnite si poradie udania súradníc v príkaze PRINT AT. Upravte program tak, aby sa generovalo náhodné rozmiestnenie hviezdičiek a obrazovka rolovala. Skúste potom program doplniť riaditeľným objektom, ktorým sa budete využívať hviezdičkám a máte hotovú jednoduchú hru.

Priklad č. 6

```
10 REM "kalendár pre roky 1753 - 2199"
20 CLS
30 DIM K(12),D(12)
40 FOR N=1 TO 12
50 READ K(N),D(N)
```

```

60 NEXT N
70 FOR N=1 TO 7
80 READ D$(N)
90 NEXT N
100 DISP "ZADAJ DATUM - DD, MM, RRRR"
110 INPUT D,M,R
120 C=INT(R/100)*100: XR=R-C: XM=INT(XR/4)
130 IF R/4 > INT(R/4) THEN 150
140 K(1)=0: K(2)=0
150 A=XR+XM+K(M)+J(C/100-16)+D: X=(A-7*INT(A/7))+1
160 PRINT D;M;R;" = "; D$(X)
170 BEEP: GOTO 100
180 DATA 1,4,4,2,4,0,0,6,2,4,5,0,0,0,3,0,6,0,1,0,4,0,6,0
190 DATA "SOBOTA", "NEDELA", "PONDELOK", "UTOROK"
200 DATA "STREDA", "STVRTOK", "PIATOK"

```

Viete vysledovať z programu matematické vzťahy, podľa ktorých výpočet prebieha?

Priklad č. 7

```

10 REM "tabulka"
20 PRINT "CISLO"; TAB(5); "NAZOV"; TAB(25); "CENA"; TAB(33);
      "POCET"
30 DISP "ZADAJ POCET POLOZIEK"
40 INPUT N
50 FOR I =1 TO N
60 DISP "NAZOV, CENA, POCET", " "
70 INPUT A$, C, P
80 PRINT I; TAB(5); A$; TAB(25); C; TAB(33); P
90 NEXT I

```

Rozširte program o vyhľadávanie položky, ktorej názov je zadaný z klávesnice s vypisovaním prislúchajúcich údajov.

Priklad č. 8

```

10 REM "domcek"
20 GCLEAR

```

```

30 FOR A=1 TO 6
40 W$=CHR$(32)
50 FOR B=1 TO 11
60 READ C: IF C=> THEN 80
70 GOTO 90
80 C=32
90 W$=W$ + CHR$(C): NEXT B
100 PRINT W$
110 NEXT A
120 FOR A=1 TO 20
130 PRINT CHR$(149);:PRINT "PRIKLAD NA PRACU S RATAZCAMI"
140 NEXT A
150 DATA 0,134,135,128,0,0,0,0,0,0,0,0 11
160 DATA 134,0,0,135,0,0,0,0,142,0,0 10
170 DATA 138,131,131,137,0,0,0,146,140,146,0,0 11
180 DATA 128,139,0,129,0,0,0,145,140,145,0,0 11
190 DATA 128,0,0,139,0,0,0,0,132,0,0,0,0 11
200 DATA 128,0,0,139,0,144,0,129,141,141,141,141 11

```

Všimnite si prevod čísla na reťazec. Nakreslite tento obrázok jednoduchším spôsobom za použitia grafických znakov na klávesnici a prikazom PRINT. Vidíte, že pomocou grafických znakov môžete kresliť pomerne zložité obrázky, nakoľko znaky na seba navádzajú.

Doplňte do programu pohybujúcu sa figúrku: CHR\$(144)

Priklad č. 9

```

10 REM "melodia"
20 A$="210070CDA688C9"
30 AD=28672
40 FOR I=0 TO 32
50 READ D
60 POKE AD+I,D
70 NEXT I
80 CODE A$
90 DATA 20,40,20,35,20,30,25,32,20,20,20,18,10,25,20,25
100 DATA 20,30,15,20,25,25,25,30,25,35,25,40,25,45,25,30
110 DATA 255

```

Všimnite si, že dátá tvoria dvojice usporiadane tak, že prvé číslo dvojice udáva dĺžku tónu a druhé výšku tónu. Súbor dát je ukončený znakom 255 (FF) a uložený od adresy 7000 H (28672), odkiaľ si program volaný na adrese 88A6 H vyberá údaje pre generovanie tónu. Napište si vlastnú melódiu.

Priklad č. 10

Nakoniec si uvedme priklad jednoduchej hry slúžiacej na ilustráciu možnosti pseudografiky a animácie. Najskôr si však prikazom monitoru SUB uložte program v strojovom kóde:

SUB 7000 CD 03 8C C9

potom si zavolajte BASIC a začnite písať nasledujúci program:

```
1 REM "pozierac"
2 DISP "
      ← VLAVO → VPRAVO"
5 DIM Z(50,50)
10 GCLEAR: P=0: O=100: T=200
20 FOR M=1 TO 0
30 X=INT(RND(3)*20+12)
40 Y=INT(RND(7)*12+2): Z(Y,X)=1
50 PRINT AT Y,X; "*"
60 NEXT M
70 X=22: Y=24
80 PRINT AT Y,X; "@"
90 A=USR (7000) (8C03)
100 IF A=8 THEN GOSUB 220
110 IF A=24 THEN GOSUB 200
120 T=T-1: IF T=0 THEN 300
130 DISP T
140 PRINT AT Y,X; " "; Y=Y-1
150 IF Z(Y,X)=1 THEN BEEP: Z(Y,X)=0: P=P+1
160 IF Y<>0 THEN 80
170 GOTO 70
200 PRINT AT Y,X; " "
210 X=X+1: RETURN
220 PRINT AT Y,X; " "
230 X=X-1: RETURN
300 PRINT AT 20,12;"S C O R E : "; P
```

```
310 DISP "OPAKOVAT? ANO/NIE"  
320 INPUT A$  
330 IF A$="ANO" THEN RUN  
340 PRINT AT 20,12;" A H O J ! "  
350 END
```

Preštudujte si program a pokúste sa ho upraviť tak, aby kontroloval vyzbierané všetkých objektov a ukončil sa vyhodnotením času.

11. G R A F I C K Ý R E Ž I M

Doteraz ste pracovali výlučne s alfanumerickými a grafickými znakmi, ktoré ste mali k dispozícii na klávesniči počítača. Keď príkaz PRINT vám umožňuje kresliť po obrazovke, táto práca je nepohodlná a má množstvo obmedzení. Preto v ďalšej státi bude vysvetlená práca so špeciálnymi grafickými príkazmi, ktoré pracujú rýchlo a s veľkým efektom.

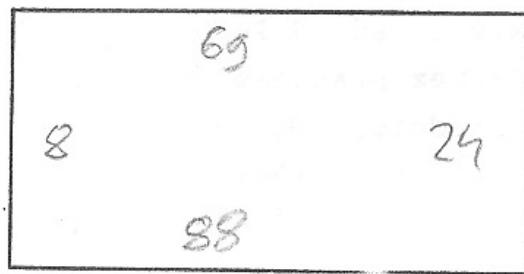
Grafické zobrazovanie motívov alebo funkcií je možné dvojma spôsobmi:

11.1 BODOVÝ REŽIM

Umožňuje zobrazovanie bod po bode (bit po bite). V tomto prípade je obrazovka počítača orientovaná podľa obr. 9.

Obr. č. 9 Bodove orientovaný prístup

∅-riadok ∅.bod



255 bod

242 riadok, 255 bod

Nemusíte mať obavu, že jednotlivé body budete musieť spájať vlastným programom. Mikropočítač má rýchly lineárny interpolátor, ktorý dokáže spojiť dve libovoľné body, pritom si môžete zvoliť svoju mierku zobrazenia. Pripomeňme si, že aj tieto príkazy môžeme používať v príamom i programovom mode.

Teraz sa zameriame na príkazy pre bodovo orientovanú grafiku. Najkôr si popíšme podrobne všetky príkazy:

SCALE	<p>» Syntax: SCALE Xmin, Xmax, Ymin, Ymax x ... hraničné hodnoty zobrazovania v osi X y ... hraničné hodnoty zobrazovania v osi Y Príkaz slúži na nastavenie transformácie medzi počtom zobrazovaných bodov a hodnotami zobrazovaných premenných. Je to vlastne nastavenie mierky zobrazovania. PRÍKLAD: 1Ø SCALE Ø, 255, Ø 242 transformácia je 1 : 1</p>
AXES	<p>Syntax: AXES x, y x, y ... súradnice priesečníku osi - môže byť aj výraz Príkaz vykreslí osi X a Y v priesečníku na daných súradničiach v závislosti na nastavenie mierky príkazom SCALE PRÍKLAD: 1Ø SCALE -1, 1, -1, 1 2Ø AXES Ø, Ø 3Ø AXES -1, -1 4Ø AXES 1, 1</p>
MOVE	<p>Syntax: MOVE x, y x, y ... súradnice X, Y - môže byť aj výraz Príkaz presunie "kresliace pero" na zadané súradnice. Od tohto miesta sa začne kreslenie najbližším príkazom PLOT. PRÍKLAD: 1Ø SCALE Ø, 1ØØ, Ø, 1ØØ 2Ø AXES 5Ø, 5Ø 3Ø MOVE 1Ø, 1Ø 4Ø PLOT 6Ø, 6Ø</p>

PLOT

Syntax: PLOT x1,y1,p; x2,y2,p;xn,yn,p
x, y ... súradnice nasledujúceho bodu - môže byť aj výraz

Ak hodnota p = 1 je udaná v príkaze, nevytvorí sa spojnice bodov, ale vykreslí sa len bod na daných súradniciach. Ak p nie je v príkaze uvedené, vytvorí sa spojnice bodov. Príkaz spojí predchádzajúci bod s bodom na súradniciach zadaných v príkaze.

PRÍKLADE: 10 SCALE Ø,1Ø,Ø,1Ø

2Ø AXES 5, 5

3Ø MOVE 2,2

4Ø PLOT 2,8;8,8;8,2;2,2

5Ø PLOT 1,7,1;7,

PEN

Syntax: PEN výraz

Príkazom môžeme riadiť jas kreslenia u grafických príkazov podobne ako pri príkaze INK v spojení s príkazom PRINT. Hodnota výraz sa vypočíta ako súčet hodnôt požadovanej činnosti:

4 ... kreslenie inverzné (čierne na bielom)

8 ... kreslenie negovaním (kde sa kreslí mení sa obsah obrazovky)

16 .. kreslenie mazaním (kde sa kreslí, tam sa maže obrazovka)

Ak nie je v súčte 8 ani 16, vykresluje sa nastavovaním t.j., kde sa kreslí, tam sa stavuje obrazovka. Ak nie je v súčte 4, kreslí sa biele na čiernom, teda "nastaviť" bod znamená urobiť ho svietiacim a "zmazať" bod znamená urobiť ho čiernym. Toto si treba uvedomiť v súvislosti s inverzným kreslením, kde je to vlastne naopak.

Vykreslovanie negovaním znamená zmenenie svietiaceho bodu na zhasnutý a naopak pri inverznom mode.

Pri zapnutí počítača sa implicitne nastavuje
PEN 8 - teda kreslenie biele na čiernom pod-
klade.

PRÍKLAD: 1Ø SCALE Ø,255,Ø,242
2Ø FOR I=2 TO 23Ø
3Ø PEN I-INT(I/4)*4
4Ø PLOT I,I,1;234-I,1
5Ø NEXT I

LABEL

Syntax: LABEL x,y; zoznam výrazov
x, y ... sú celočíselné hodnoty, kolokrát
sa majú rozmery v osi X a Y zväčšiť
od základnej veľkosti kreslenia v ma-
tici bodov 5 x 7

Priaz slúži podobne ako priaz PRINT na
vykreslovanie znakov s tým, že si môžeme
ľubovoľne zväčšiť znak, ktorý máme na klá-
vesnici pristupný. Samozrejme, že v prípade
väčšieho násobku rozmeru sa zobrazuje len
časť viditeľná vzhľadom k nastaveniu pri-
zom SCALE. Ak priaz používate v programe
viackrát s rovnakým zväčšením, môžete pou-
žiť skrátený zápis LABEL * ;výraz.

PRÍKLAD: SCALE Ø,1ØØ,Ø,1ØØ EOL
MOVE 1Ø, 5Ø
LABEL 3,5, "MATO"
LABEL * ;"-1"

FILL

Syntax: FILL x, y; m
x, y ... počet bodov v smerech X a Y
m bitová maska

Priaz umožňuje vyplňovať jednoduché obraz-
ce danou "m". Čalej priaz FILL umožňuje
vykreslovanie rovných čiar rôznych typov
v smere X a Y alebo previesť inverziu nad
požadovaným miestom.

PRÍKLAD: 1Ø SCALE Ø,1ØØ,Ø,1Ø

```
20 MOVE Ø,Ø
30 FILL 255,242;1      inverzia obrazovky
40 AXES Ø,Ø
50 FOR I=1 TO 10
60 MOVE I,Ø
70 FILL 1,5;1          nakreslí dielik na osi X
```

v ďalšej státi uvádzané príklady vám dajú predstavu o použití grafických povelov v spolupráci so štandardnými príkazmi.

11.2 PRÍKLADY BODOVE ORIENTOVANÉHO PRÍSTUPU

Priklad č. 11

```
10 REM "stromcek"
20 GCLEAR
30 SCALE Ø,10,Ø,10
40 READ X,Y
50 MOVE X,Y
60 FOR I=1 TO 11
70 READ X,Y
80 PLOT X,Y
90 NEXT I
100 DATA 1,2,9,2,7,4,8,4,6,6,7,6,5,8,3,6,4,6,2,4,3,4,1,2
```

Rozširte program tak, aby mal stromček kmeň.

Priklad č. 12

```
10 REM "histogram percentualnych hodnot"
20 GCLEAR
30 SCALE Ø,10,Ø,250
40 FOR I=0 TO 9
50 READ A
60 MOVE I,Ø
70 FILL 15,2*A;1
80 NEXT I
90 AXES Ø,Ø
100 DATA 23,13,10,53,65,85,33,40,20,5
```

Upravte program tak, aby sa nad každým stĺpcom objavilo jeho poradové číslo.

Priklad č. 13

```
10 REM "napis"
20 SCALE 0,10,0,10
30 GCLEAR;
40 MOVE 0,4      ,
50 FOR I=1 TO 6
60 X$="MATO~"
70 LABEL 1,2; X$
80 BEEP
90 NEXT I
100 PAUSE (5); GOTO 20
```

Zmeňte program na výpis iného textu, v inej mierke a rozličnej velkosti písma.

Priklad č. 14

```
10 REM "raster"
20 GCLEAR
30 SCALE 0,10,0,10
40 FOR I=0 TO 10
50 AXES I,I
60 NEXT I
```

Zvolte SCALE tak, aby ste mali rozmerovo rovnaké štvorce. Doplňte súradnice rohov príkazom LABEL.

Priklad č. 15

```
10 REM "popis osi X"
20 GCLEAR
30 SCALE -1,11,-1,11
40 AXES 0,0
50 FOR I=1 TO 10
60 MOVE I, 0
70 FILL 1,4;1
80 MOVE I,-.4
90 LABEL 1,1;I
```

100 NEXT I

Popište aj os Y a doplňte program o vykreslenie zvolenej funkcie.

Priklad č. 16

```
10 REM "funkcia SIN (x)"
20 GCLEAR
30 PI=22/7
40 SCALE 0,2*PI,-1,1
50 AXES 0,0
60 MOVE 0,0
70 FOR X=0 TO 2*PI + .3 STEP PI/20
80 PLOT X,SIN(X)
90 NEXT X
```

Upravte program pre väčší počet sinusoviek tak, aby sinusovka bola plošne vyšrafovovaná.

Priklad č. 17

```
10 REM "interpolator"
20 GCLEAR: MOVE 0,0
30 SCALE 0,100,0,100
40 FOR I =1 TO 50
50 X=INT(RND(2)*100): Y=INT(RND(5)*100)
60 PLOT X,Y
70 NEXT I
```

Pokúste sa napiшаť program, ktorý by kreslil náhodné rozmiestňované štvorce, z ktorých budú niektoré náhodne vyplnené.

Priklad č. 18

```
10 REM "kruznica"
20 GCLEAR
30 SCALE -2,2,-1.6,1.6
40 DEG
50 FOR I=1 TO 360 STEP 4
60 MOVE 0,0
70 PLOT COS(I), SIN(I)
```

80 NEXT I

Napište program pre kreslenie elipsy alebo štvorca podobným spôsobom. Skúste tento program upraviť bez použitia príkazu DEG (uhlové údaje budú v radiánoch) a posúdte odlišnosti. Vyskúšajte si pritom funkciu PEN.

Priklad č. 19

```
10 REM "pohyb"  
20 GCLEAR  
30 SCALE 1,100,1,100  
40 FOR I=1 TO 100  
50 MOVE I,50  
60 GOSUB 100  
70 NEXT I: GOTO 20  
100 PLOT I,30  
110 MOVE I,50  
120 PLOT I,30  
130 RETURN
```

Upravte podprogram použitím príkazu PEN a porovnajte odlišnosti. Vidíte, že bodový prístup sa nevyznačuje veľkou rýchlosťou.

Na záver vám predkladáme program dokumentujúci grafické možnosti počítača pri kreslení komplikovaných objektov. Program beží asi 20 minút, preto ak si chcete uspokojiť zvedavosť, zmeneňte hodnotu v príkaze STEP=2.

Priklad č. 20

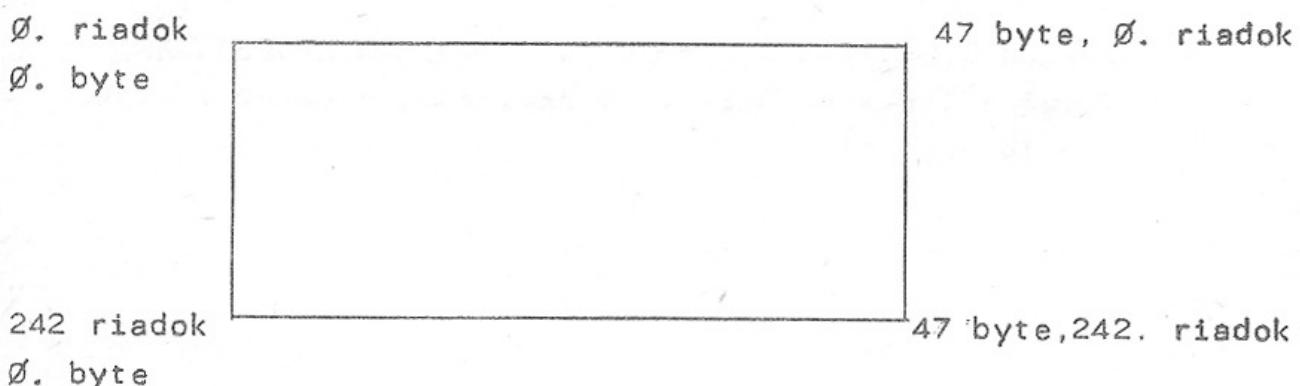
```
5 REM "bublina"  
10 SCALE -20,30,-20,30  
20 FOR Y=-20 TO 20 STEP .5  
30 FOR X=-40 TO 30 STEP .5  
40 R=SQR (X^2 + Y^2)  
50 IF R=0 THEN 80  
60 PLOT X+1*((Y+17)/2), Y+20*(SIN(R)/R)  
70 GOTO 90  
80 PLOT (Y+17)/2, Y+20  
90 NEXT X
```

```
100 MOVE -20, -20  
110 NEXT Y  
120 END
```

11.3 BYTE ORIENTOVANÝ PRÍSTUP

Pre niektoré aplikácie - hry s pohybujúcimi sa objektami, o ktoré máte zrejme najväčší záujem - je bodove orientovaný prístup pomalý. Preto má BASIC-G ďalšie dva príkazy, ktoré pracujú nad byte orientovaným priestorom podľa obr. č. 10.

Obr. č. 10 Byte orientovaný priestor



Najskôr si podrobne vysvetlime význam jednotlivých prikazov s uvedením presného syntaxu:

BMOVE Syntax: BMOVE x,y
x ... súradnica na osi X v intervale Ø - 47
y ... súradnica na osi Y v intervale Ø - 242
Príkaz slúži na nastavenie miesta vykreslovania podobne ako príkaz MOVE, ale stým rozdielom, že SCALE je pevne určené, ako je uvedené na obr.
č. 1Ø. Pozícia BMOVE Ø,Ø sa nachádza v Ľavom hornom rohu obrazovky.

PRÍKLAD: BMOVE 1Ø,15Ø nastavi polohu 1Ø byte na osi X a polohu 15Ø riadok v osi Y

BPLOT

Syntax: BPLOT x\$,n

x\$... reťazcová premenná, ktorej byte sa budú rozkreslovať

n argument udávajúci počet byte premennej, ktoré sa rozkreslia v riadku vedľa seba.

Priaz rozkreslí v smere zhora nadol prvky premennej zadanej v priaze. V jednom riadku bude vedľa seba rozkreslených byte v počte danom parametrom "n". Ďalšie prvky premennej budú rozkreslované pod predchádzajúcim riadkom.

PRÍKLAD: BMOVE T\$,1

BMOVE T\$,2

Ako príklad vykreslenia obrázku si uvedieme nasledovné. Potrebujeme vykresliť trojuholník so základňou 6 bodov a vysokou 3 body podľa obr. 11.

Obr. č. 11 Definovanie byte pre rozkreslenie objektu

						váha:	1	2	4	8	16	32	dekadicky
		*	*				Ø	Ø	1	1	Ø	Ø	12
	*	*	*	*			Ø	1	1	1	1	Ø	30
*	*	*	*	*	*		1	1	1	1	1	1	63

Pripomeňme si, že dva bity najvyššej hodnoty sú bezvýznamné. Definovaním binárneho tvaru a prepočítaním do dekadickej hodnoty si môžeme definovať reťazec

$$T\$ = \text{CHR}$(12) + \text{CHR}$(30) + \text{CHR}$(63)$$

Rozkreslenie takto definovaného reťazca prevedie priaz BPLOT, ktorý ako parameter za reťazcom pribera číslo, zodpovedajúce počtu byte z reťazca, ktoré chceme vykresliť vedľa seba do jedného riadku obrazovky. Ďalšia n-tica byte bude zobrazená

v nasledujúcich riadkoch. Ako príklad na pochopenie nám posluží na reťazec T\$:

Prikaz	Obrazovka
- BPLOT T\$,1	*** *** *** ***
- BPLOT T\$,2	*** *** ***
- BPLOT T\$.3	*** *** ***

Ďalšie zvyšovanie argumentu už nemá zmysel, nakoľko rozkreslovanie skončí pri treťom znaku.

Zmazať motív môžete opäťovným prekreslením na to isté miesto, alebo definovaním parametru v príkaze PEN, ktorý je popísaný v stati 11.1.

Zvláštnym problémom je pohybujúci sa motív. Najjednoduchší spôsob je zmazanie pôvodného a nakreslenie nového motívu na inom mieste. Možné to je, ale výsledok vás neuspokoji, nakoľko pohybujúci sa motív bude blikať. V tomto prípade je možné použiť filozofiu vzájomného prekrývania sa motívov. Zoberme si náš reťazec ako príklad pohybu smerom dolu:

Pôvodná poloha	nová poloha	prekrývajúci reťazec
***	***	= 12
***	***	= 18
***	***	= 33
***	***	= 63
T\$	žiadna poloha	T1\$

kde $T\$ = \text{CHR}$(12) + \text{CHR}$(18) + \text{CHR}$(33) + \text{CHR}$(63)$ a potom vykonajte nasledovnú sekvenciu.

```
BMOVE 20,20
BPLOT T$,1
BPLOT T1$,1
BMOVE 20,21
```

BPLOT T\$,1

, atď.

vidíme, že motív sa pohybuje. Skúste si zostaviť program, ktorý bude pohybovať motívom v danom rozsahu.

Na záver si vyskúšajte a podrobne preštudujte uvádzané príklady byteového prístupu na obrazovku.

Priklad č. 21

```
10 REM "rozkreslenie retazca"
20 FOR I=1 TO 24
30 A$=A$ + CHR$(21) + CHRS(42)
40 NEXT I: GCLEAR
50 BMOVE 10,10
60 BPLOT A$, 1
70 BPLOT A$, 2
80 BPLOT A$, 3
90 BPLOT A$, 4
100 BPLOT A$, 8
110 BPLOT A$, 12
120 BPLOT A$, 24
```

Z príkladu je úplne zrejmé, vplyv argumentu v príkaze BPLOT na rozkreslenie premennej. Vyskúšajte si príklad, keď argument je nepárne číslo.

Priklad č. 22

```
10 REM "obrazok"
20 GCLEAR
30 FOR I=1 TO 32: READ A: A$=A$ +CHR$(A): NEXT I
40 BMOVE 20, 100
50 BPLOT A$, 2
60 DATA 32,1,48,3,24,3,54,3,8,7,16,6,16,7,40,7
70 DATA 40,13,8,13,8,13,8,12,8,14,48,23,0,2,48,7
```

Vytvorte program, ktorý bude pracovať s viacerými reťazcami, ktoré v príkaze BPLOT budú rozkreslované v rôznych vzájomných kombináciách a ich časťach.

Priklad č. 23

```
10 REM "pohyb mazaním"
20 GCLEAR
30 FOR I=1 TO 48: READ A: A$=A$+CHR$(A): NEXT I
40 X=20: Y=100
50 BMOVE X,Y: BPLOT A$,3
60 BMOVE X,Y: BPLOT A$,3
70 Y=Y-2: IF Y=0 THEN 100
80 BMOVE X,Y: BPLOT A$,3
90 GOTO 60
100 BMOVE X,Y: BPLOT A$,3
110 BMOVE X,Y: BPLOT A$,3
120 Y=Y+2: IF Y=220 THEN 50
130 BMOVE X,Y: BPLOT A$,3
140 GOTO 110
150 DATA 0,63,0,48,63,3,56,63,7,57,63,7
160 DATA 12,12,12,48,51,3,54,63,27,53,51,43
170 DATA 13,12,44,57,63,30,25,63,38,49,0,35
180 DATA 34,63,17,0,33,0,0,33,0,48,33,3
```

Napište program pre pohyb objektu zloženého z viac byte (aspoň 10) v smere osi X a Y a porovnajte rýchlosť s objektom menším.

Priklad č. 24

```
10 REM "pohyb prekrývaním"
20 FOR I=1 TO 8: READ A: A$=A$ + CHR$(A): NEXT I
30 FOR I=1 TO 9: READ B: B$=B$ + CHR$(B): NEXT I
40 X=20: Y=20: GCLEAR
50 BMOVE X,Y: BPLOT A$,1
60 BMOVE X,Y: BPLOT B$,1
70 Y=Y + 1: IF Y=200 THEN 40
80 GOTO 60
90 REM "zakladny motiv"
100 DATA 12,30,18,30,12,12,18,18
110 REM "prekryvajuci motiv"
120 DATA 12,18,12,12,18,0,30,0,18
```

Porozmýšľajte, ako je možné urobiť plynulý pohyb motívu v smere osi X. Prezradíme Vám, že pre zacovanie dokonalej plynulosti pohybu nevystačíte s jedným prekrývajúcim reťazcom. Treba si uvedomiť, že v smere osi X máme k dispozícii minimálny krok 1 byte, kým v smere osi Y je to 1 bit.

No a nakoniec uvedieme jeden špeciálny program, ktorý je určený pre počítač PMD-85 !!! Pomocou tohto programu - ak ho nahráte do počítača PMD-85 a spusťte - môžete načítať program určený pre PMD-85. Ak potom stlačíte klávesu "W", načítaný program môžete nahrať s protokolom mikropočítača MATO-1. Signál je generovaný na konektore K4 (GPIO B - pin č.14). Takto môžete používať programy určené pre PMD-85 na Vašom mikropočítači. Pozor - nie všetky programy Vám budú pracovať správne, nakoľko väčšina hier nepoužíva štandardný podprogram klávesnice (SCAN).

FILENAME: CONVERT

ADR	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	3E	1C	CD	00	85	CD	44	02	21	00	DA	CD	8D	02	22	52
0010	22	2E	2E	2E	43	49	54	41	4E	49	45	20	46	49	4C	45
0020	20	5A	4F	20	53	54	41	4E	44	41	52	44	4E	45	48	4F
0030	20	56	53	54	55	50	55	00	21	00	DE	CD	8D	02	22	57
0040	22	2E	2E	2E	5A	41	5A	4E	41	4D	20	46	49	4C	45	20
0050	4E	41	20	56	59	53	54	55	50	45	20	47	50	49	4F	2D
0060	42	20	50	42	30	00	21	06	E1	CD	8D	02	28	4B	34	2F
0070	50	49	4E	20	43	2E	31	34	29	20	53	20	50	52	4F	54
0080	4F	4B	4F	4C	4F	4D	20	4D	41	54	4F	2D	31	00	21	00
0090	E5	CD	8D	02	45	4F	4C	2E	2E	4E	41	56	52	41	54	
00A0	20	44	4F	20	4D	4F	4E	49	54	4F	52	55	20	50	4D	44
00B0	2D	38	35	00	CD	A3	88	CD	A1	84	FE	52	CA	D1	00	FE
00C0	57	CA	89	01	FE	89	CA	40	8C	FE	0D	CA	40	8C	C3	B4
00D0	00	21	10	FA	CD	8D	02	4D	47	46	20	4F	4E	20	21	20
00E0	20	20	20	20	20	20	20	20	20	20	20	20	20	00	CD	A3
00F0	88	CD	E2	8D	21	10	FA	22	3E	CO	3A	B2	C1	CD	25	81
0100	3E	2F	CD	00	85	3A	B3	C1	CD	00	85	3E	20	CD	00	85

0110	21	B8	C1	0E	08	7E	CD	00	85	23	0D	C2	15	01	2A	B6
0120	C1	EB	21	A0	02	CD	C2	8D	C2	4C	01	21	10	FA	CD	8D
0130	02	52	45	41	44	49	4E	47	20	4F	4B	20	21	20	20	20
0140	20	20	20	20	20	20	20	20	00	C3	B4	00	21	10	FA	CD
0150	8D	02	46	49	4C	45	20	45	52	52	4F	52	20	21	20	20
0160	20	20	20	20	20	20	20	20	00	C3	B4	00	21	10	FA	
0170	CD	8D	02	4D	47	46	20	4F	4E	20	2D	20	54	48	45	4E
0180	20	54	59	50	45	20	21	00	C9	3E	80	D3	4F	CD	6D	01
0190	CD	A3	88	CD	A1	'84	CD	F5	01	21	10	FA	CD	8D	02	4D
01A0	47	46	20	4F	46	46	20	21	20	20	20	20	20	20	20	20
01B0	20	20	20	20	20	00	C3	B4	00	3E	01	D3	4D	CD	38	02
01C0	AF	D3	4D	CD	3F	02	C9	3E	01	D3	4D	CD	38	02	CD	3F
01D0	02	AF	D3	4D	CD	3F	02	37	C9	06	FF	CD	ED	01	3E	55
01E0	CD	22	02	C9	06	80	CD	ED	01	CD	C7	01	C9	CD	B9	01
01F0	05	C2	ED	01	C9	06	FF	CD	ED	01	CD	D9	01	21	B2	C1
0200	11	0D	00	CD	10	02	CD	E4	01	2A	B6	C1	EB	21	A0	02
0210	13	06	00	7E	CD	22	02	7E	80	47	23	1B	7A	B3	C2	13
0220	02	78	C5	4F	CD	C7	01	06	08	79	1F	4F	DC	C7	01	D4
0230	B9	01	05	C2	29	02	C1	C9	3E	22	3D	C2	3A	02	C9	3E
0240	44	C3	3A	02	21	00	D0	CD	8D	02	43	4F	4E	56	45	52
0250	54	20	50	4D	44	20	38	35	20	2D	2D	3E	20	4D	41	54
0260	4F	2D	31	20	20	20	20	20	20	20	20	20	20	20	20	20
0270	3C	43	3E	31	39	38	39	2F	55	52	00	21	40	Q2	22	3E
0280	C0	0E	30	3E	3D	CD	00	85	0D	C2	83	02	C9	22	3E	C0
0290	E1	7E	23	E5	FE	0D	C8	FE	00	C8	CD	00	85	C3	90	02

pin č.14.....výstup signálu
pin č.1.....signálová zem

Pohľad na konektor K4 (GPIO B) počítača PMD-85

Výstupná úroveň signálu je TTL, čo je pre väčšinu magnetofonov príliš veľa. Preto je potrebné zníženie úrovne odporovým deličom 1:20.

12. MODIFIKOVANIE SYSTÉMU

Mikropočítač si všetky dôležité pamäťové miesta odkladá do svojho zápisníka umiestneného na adresách podľa obr. č. 8, ktorý sa nachádza v pamäti RWM, takže ich môžeme prepisať a tým zmeniť chovanie systému. Niektoré dôležité miesta si popíšeme:

adresa	význam
84E6 H	Náčítanie hodnoty bodu v príkaze PLOT
84F3 H	Náčítanie hodnoty byte v príkaze BPLOT
C03E H	Poloha kurzoru pre výpis v príkaze PRINT
ØØ2E H	Veľkosť dĺžky výpisu riadku v príkaze LIST (=48)
C1B4 H	Štartovacia adresa práve nahraného programu z magn.
C1B6 H	Dĺžka práve nahraného programu z magn.
C13Ø H	Ukazovateľ tabuľky tónov pre príkaz BEEP
F4 H	Adresa výstupného portu akustického meniča - bit D1

13. CHYBOVÉ HLÁSENIA

V priamom aj v nepriamom režime vám mikropočítač po zistení chyby túto oznamí v dialógovom riadku - v programovom režime naviac oznamí aj číslo riadku, v ktorom chybu zistil. Tieto hlásenia sú uvádzané v anglických skratkách a obsahujú maximálne 11 znakov:

SYNTAX ERR	Chybne zadaný príkaz
FNC. PARAM	Nesprávny parameter funkcie
SUBSCR.RNG	Výraz v príkaze alebo funkcia je mimo definovaného rozsahu napríklad v príkaze DIM
ONLY IN PG	Príkaz možno použiť len v programovom mode INP, DEF, INPUT
OVERFLOW	Prekročený maximálny rozsah výpočtu
DV BY ZERO	Vo výraze došlo k deleniu nulou
TYPE CONV	Chybný prevod, napr. medzi číselnou a retázcovou premennou

CAN'T CONT	Chybný povel CONT na pokračovanie v prorušenom programe. Pred týmto povelom bol použitý iný, ktorý spôsobil zmenu v programe.
NO FOR STM	Chybne realizovaný cyklus v prikaze FOR TO NEXT
ARR ALLOG	Chybná dimenzia poľa, alebo viackrát dimenzované pole
DATA EXHAU	Chýbajúce dátá pre čítanie povelom READ
PG TOO BIG	Preplnenie pamäťového priestoru. Program treba skompresovať vynechaním medzier a použitím viacnásobných prikazov v riadku
STRING LONG	Dĺžka reťazca je viac ako 255 znakov
NO STR, SPC	Nie je už žiadne voľné miesto pre reťazec
NUMB. NONEX	Programový riadok s uvedeným číslom neexistuje
STR. ALGRT	Reťazcový výraz je v prikaze príliš dlhý alebo zložitý. Treba ho rozdeliť do niekoľko jednoduchých prikazov.
STOP	Zastavený program
RETURN ERR	V sekvencii programu bol nájdený prikaz RETURN bez volania GOSUB
INPUT ERR	Chyba pri vkladaní dát v prikaze INPUT, napr. nečíselný znak v číslicovej premennej
FILLE BOUND	Prekročený maximálny počet záznamov magnetofónu 99
FILLE ERROR	Vznikla chyba pri prenose dát, napr. viednym čítaním
FIELD LOST	Prekročený žiadaný počet vstupných dát napr. pri INPUT
FILLE SMAL	Dátový súbor čítaný z pásky nekorešponduje s dimensiou danej premennej v programe

14. ZOZNAM PRÍKAZOV A FUNKCIÍ BASIC-G

Nasledujúci zoznam všetkých prikazov vyššieho programovacieho jazyka BASIC-G vám poslúži na rýchlu orientáciu v začiatkoch, keď ešte ste si nezapamätali celý súbor alebo správny syntax.

14.1 EDITOVACIE PRÍKAZY:

LIST	AUTO	SAVE n
LIST n	AUTO n	LOAD n
LLIST	NEW	ROM n
LLIST n		CHECK n

14.2 RIADIACE PRÍKAZY:

RUN	RUN n	GOTO n
CONT		

14.3 ŠTANDARDNÉ FUNKCIE:

SIN (X)	INP (I)	INKEY
COS (X)	RND (X)	PEEK (I)
TAN (X)	SGN (X)	APEEK (I)
ANT (X)	FRE (X)	USR (I)
LOG (X)	SPC (I)	SQR (X)
EXP (I)	TAB (I)	BIT A, B
ABS (X)	STATUS A,B	ADR (A)
INK (I)	PEN (I)	

14.4 RETAZCOVÉ FUNKCIE:

ASC (X\$)	RIGHT\$ (X\$,I)	STR\$ (X)
FREE (X\$)	MID\$ (X\$,I)	VAL (X\$)
LEFT\$ (X\$,I)	LEN (X\$)	HEX\$ (I)

14.5 ŠTANDARDNÉ PRÍKAZY:

BEEP	PRINT	AT Y,X
DEF FNC	GOTO	PAUSE (X)
DIM	IF ~ THEN	REM
DSAVE	LET	CLEAR
DLOAD	ON	GCLEAR
FOR TO NEXT	ON ERR	NULL (X)
STEP	DTSP	READ
GOSUB	RETURN	DATA
DEG	RAD	

14.6 GRAFICKÉ PRÍKAZY:

SCALE	PLOT	LABEL
MOVE	BMOVE	FILL
AXES	BPLOT	

14.7 VSTUPNO-VÝSTUPNÉ PRÍKAZY:

OUTPUT	CONTROL	LIST #
ENTER	STATUS A,B	

14.8 PRÍKAZY PRE SYSTÉM 8080:

CODE	USR	ROM
POKE	APOKE	WAIT
PEEK	APEEK	INP
OUT		

15. ÚDRŽBA MIKROPOČÍTAČA

Mikropočítač je riešený tak, aby nevyžadoval žiadnu zvláštnu údržbu zo strany užívateľa. Napriek tomu je dobre ho po ukončení práce zakryť proti vnikaniu prachu. Ak by ste chceli po čase vyčistiť skrinku a klávesnicu mikropočítača, doporučujeme na čistenie jemnú mäkkú látku n a v l h č e n ú v letenej vode s malou príasadou saponátového čistiaceho prostriedku napr. JAR.

Doporučujeme vám riadiť sa pri práci s počítačom nasledujúcimi zásadami:

- neponechávajte dlhšiu dobu mikropočítač na slnku alebo v blízkosti tepelného zdroja
- chráňte mikropočítač pred vniknutím tekutiny do systému
- nepripájajte na aplikačný konektor systémovej zbernice iné, než sú doporučené zariadenia, hlavne nie také, ktoré by využívali napájanie z mikropočítača a nepoznáte ich spotrebú
- chráňte mikropočítač pred nárazmi a pádmi
- opravy všetkého druhu zverte radšej značkovej opravovni obchodno-servisnej organizácii

16. TECHNICKÉ ÚDAJE

Napájacie napätie	220 V / 50 Hz
Celkový príkon mikropočítača	max. 18 VA
Kapacita operačnej pamäte	48 kB
Kapacita pevnej pamäte ROM	16 kB
Rozmery mikropočítača	300 x 192 x 67 mm
Váha mikropočítača	cca 1600 g
Max. odber zo zdroja na aplikáčnom konektore	100 mA pri 5 V=

17. ZÁVER

Po preštudovaní tejto príručky ste sa oboznámili s obsluhou a prácou mikropočítač MAŤO viac pod vyšším programovačím jazykom BASIC-G, menej však s programovaním na úrovni strojového kódu mikroprocesoru 8080. Nie je možné popísat v krátkej príručke všetky možnosti vášho mikropočítača. Ak máte hlbší záujem o preniknutie do tajov výpočtovej techniky, doporučujeme vám obrátiť sa na niektoré organizácie, ktoré uskutočňujú výuku mikropočítačov. Z tých najrozšírenejších a najaktívnejších uvedieme aspoň niekoľko:

1. Stanica mladých technikov pri okresnom (krajskom) dome pionierov a mládeže
2. Klub výpočtovej techniky pri okresnom (krajskom) výbere Zväzarmu
3. Krajský klub vedecko-technickej činnosti mládeže pri KV SZM

Zvládnutie problematiku mikropočítačov je možné aj štúdiom odbornej literatúry, i keď v tomto smere je v ČSSR situácia bohužiaľ zlá. Nevychádza u nás žiadny špecializovaný časopis a tiež budete pri získavaní informácií odkázani na zahraničné časopisy, z ktorých vám uvedieme aspoň tie, ktoré vychádzajú v pribuznom slovenskom jazyku:

1. SDĚLOVACÍ TECHNIKA, AMATERSKÉ RÁDIO, ELEKTRONIKA,
VEDA A TECHNIKA MLÁDEŽE, ABC

ČSSR

Vo svete vychádza veľké množstvo časopisov špecializujúcich sa na počítače, programovanie a aplikácie. Nové vznikajú, staré zanikajú a tak poskytnúť ucelený prehľad je nemožné. Doporučujeme vám, aby ste sa obrátili na krajskú vedeckú knižnicu, kde sú k dispozícii aspoň tie najznámejšie ako napr. BYTE, MC, HOME COMPUTER a pod.

Situácia u nás v oblasti odbornej literatúry je podobná situácii v časopisoch. Väčšina kníh, ktoré vyšli v obore počítačovej techniky je encyklopedického alebo prísne teoretického charakteru.

Napriek tomu uvádzame zoznam literatúry, ktorá vám v začiatkoch pomôžе:

E. KOLLERT:	Výpočetní technika
Z. SOBOTKA:	Návrh mikropočítačov (otázky a odpovede)
Z. SOBOTKA:	Architektúra a programovanie (otázky a odpovede)
J. STARY:	Mikropočítač a jeho programování
J. SLIPKA:	Navrhování mikroprocesorových systémů
I. KOČIŠ, SUTKO:	Mikroprocesory a mikropočítače
Kolektív:	Slovník výpočetní techniky /anglicko-český/

Z anglickej literatúry uvádzame len niekoľko titulov, ktoré subjektívne pokladáme za najlepšie a môžeme udať prameň dosťupnosťi:

Osborne/McGraw-Hill: 8080A/8085 assembly language programming
(ŠVK B Bystrica číslo A 29291)

Prentice-Hall, inc: Microcomputer experimentation with
the INTEL sdk-85
(ŠVK B. Bystrica číslo B 7777)

Hayden · Microcomputers system
(ŠVK B. Bystrica číslo B 9826)

Predpokladáme, že v začiatkoch vás bude zaujímať hlavne oblasť hier, ktoré na vašom počítači môžete hrať. Uvedieme aspoň tie najzaujímavejšie, ktoré sú medzi užívateľmi počítačov PMD-85 rozšírené a môžete ich ľahko získať v niektornej zo spomínaných organizácií. Je pochopiteľné, že uvádzaný prehľad dostupných programov nie je zdaleka úplný a neustále sa rozširuje.

Programy v strojovom kóde:

- | | |
|-----------------|-------------------|
| 1. MANTIC MINER | 7. JET PAC |
| 2. KURIATKO | 8. AUTO |
| 3. ŽABY | 9. PAMPÚCH |
| 4. INVADERS | 10. RESCUER WILLY |
| 5. TEHLY | 11. SPACE RAIDERS |
| 6. ŠACH | 12. CATEGORIC |

Programy v jazyku BASIC-G:

- | | |
|----------------|-------------------------|
| 1. PAVÚKY | 3. PRISTÁTIE NA MESIACI |
| 2. LODIČKY | 9. HÁDANIE ČÍSEL |
| 3. MÍNOVÉ POLE | 10. VOTRELCI |
| 4. OTHÉLO | 11. RADAR |
| 5. KING-KONG | 12. ZÁPALKY |
| 6. WURMI | 13. HARRIER ATACK |
| 7. SÚBOJ | 14. FORMULA |

No a nakoniec sme si nechali informácie, aké sú vytvorené programové prostriedky (nazývané tiež systémové vybavenie), pomocou ktorých môžete písat vlastné programy v strojovom kóde mikroprocesora 8080 v jazyku symbolických adres (JSA), ako aj iné užitočné programy:

- | | |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. MRS | Rezidentný operačný systém obsahujúci textový editor, prekladač z JSA, spojovací a odladovací program, modul binárnej knižnice a spätný prekladač do JSA |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------|

- | | | |
|------------|---|--------------------------------------------------------------------------------------------------------------------------------|
| 2. COPY + | } | sú rôzne univerzálné kopírovacie programy |
| COPY-2 | | |
| MRCOPY + | | |
| 3. TEXTPRO | | Textový procesor pre prácu s textovými
súbormi orientovaný na obrazovku a so sério-
vým interfejsom výstupu na tlačiareň |
| 4. TESTRAM | | Testovací program operačnej pamäte RWM |
| 5. GRAFEDI | | Grafický editor umožňujúci kreslenie grafic-
kých motívov a prácu s nimi |
| 6. UNIPROG | | Program, ktorý pomocou špeciálneho prípravku
umožňuje programovanie pamäti typu PROM
a EPROM všetkých typov |

Tým sme vyčerpali všetky informácie, ktoré sme pokladali za nutné pre prvé oboznámenie sa s počítačom MAŤO. Sme pre-
svedčení, že vám pomôžu zvládnuť prácu s počítačom a umožnia
vám prežiť mnoho pekných chvíľ zábavy, ale aj aktívneho odpo-
činku. K tomu vám prajeme mnoho úspechov.

R1

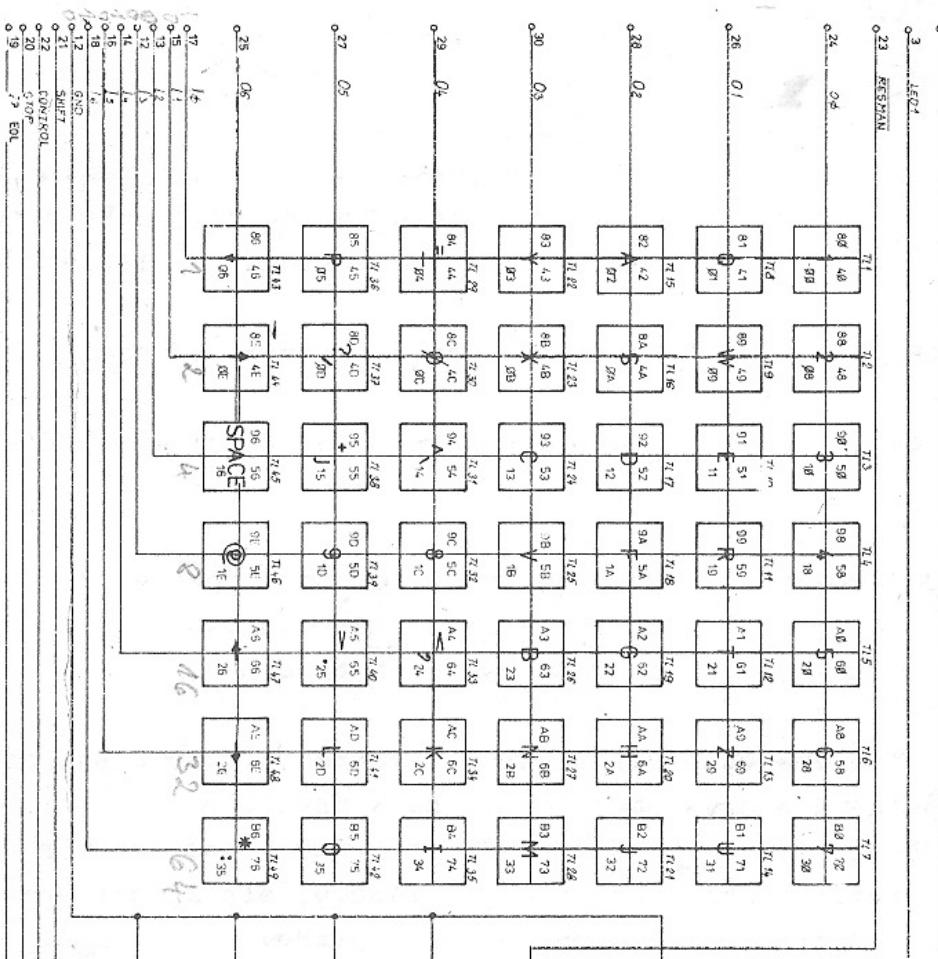
REF20

3

JFC4

o 23

RESMAN



NORM	NORM
+	+
CONT	SHIFT
	NORM

11. ELEKTRICKÁ SCHÉMA KLÁVESNICE