

## Mikropočítač PMD 85-2

Mikropočítač PMD 85-1, ktorý vyrába TESLA, Spotrebná elektronika, k. p. Bratislava, prešiel vo VVZ podniku inovačným cyklom. Na trh sa začína dodávať PMD 85-2, ktorý má oproti predchádzajúcemu typu mnoho úprav. Má novú klávesnicu na báze vodivého elastomeru, čím sa zjemnil jej chod a zvýšila sa trvanlivosť. Celková konštrukcia mikropočítača pri správne vykonávanej obsluhu zaručuje vysokú prevádzkovú spoľahlivosť. Ďalšia úprava sa týka snímania údajov z magnetofonu. Signál je programovo analyzovaný, čo umožnilo vyriešiť fázovo nezávislé snímanie nahrávok; tzn. záznam dat je nezávislý na type magnetofonu. PMD 85-2 má nový monitor a novú verziu BASIC-G V2.0. Monitor je do značnej miery kompatibilný s pôvodnou verziou. Niektoré zo zmien monitoru: V prípade, že je založený modul ROM, po zapnutí PMD 85-2 sa vloží obsah modulu do RWM počítača automaticky, bez príkazu BASIC G, v opačnom prípade sa prihlási podsystém monitoru. Tabuľka znakov obsahuje všetky znaky podľa ISO-7, tzn. aj malú abecedu a rôzne ďalšie znaky ako štvorček, vlnovku a pod., ktoré nie sú na klávesnici, ale sú prístupné užívateľovi. Pridržaním klávesu sa bude jeho funkcia opakovať – autorepeat. Väčšina klávesov má dve funkcie. Niektoré ďalšie možnosti sú: zmazávanie údajov na obrazovke, výpis na obrazovku plným jasom, polojasom, blikaním a v inverzii, vypínanie akustickej signalizácie, nie je povinná medzera medzi kľúčovým slovom a parametrami, obsah pamäte sa vypisuje bez prestávok s možnosťou pozastavenia alebo ukončenia výpisu, niekoľkonásobné zrýchlenie výkonu grafických funkcií, číslo nahrávky môže byť od 00 až po 99, možnosť pripojenia ako terminál k počítaču (vtedy

sa menia funkcie riadiacich klávesov), možnosť zmeny prenosovej rýchlosti, umožnenie blokového prenosu údajov.

Zmeny v BASIC G V2.0 sú prevedené tak, že všetky programy, ktoré pracovali pod BASIC G V1.0 a nepoužívali služby monitora neuvedené v príručke k PMD 85-1, ani nepoužívali zvláštne vedomosti o interpreti BASICu, budú pracovať. Odstránili sa chyby a previedli sa drobné zlepšenia, doplnili sa nové možnosti už existujúcich príkazov a doplnili sa nové príkazy. Niektoré zo zmien sú: možnosť písania niekoľkých grafických príkazov na jeden riadok, premenné X a Y je možné používať ako bežné premenné aj keď sa používajú grafické príkazy, goniometrické funkcie sa počítajú presne na plných 6 až 7 desatinných miest, prepínanie režimu stupne alebo radiány, povel MONIT je nahradený stlačením kombinácie dvoch klávesov, výpis riadkov dlhších než 48 znakov sa vypíše do dvoch riadkov obrazovky, možnosť zmeniť dĺžku vypisovaného riadku, umožnenie nastavenia miesta na obrazovke pre výpis textu, pozastavenie vykonávania programu alebo výpisu stlačením klávesa SHIFT, po uvoľnení klávesu činnosť pokračuje ďalej. BASIC G V2.0 dovoľuje zapisovať konštanty v programe aj v šestnástkovej sústave. Obsahuje aj jednoduchú možnosť chránenia programov proti prezeraniu a kopírovaniu.

Inovácia výroby znižuje poruchovosť, predlžuje životnosť klávesnice, zvyšuje rýchlosť výkonu grafických príkazov, umožňuje komfortnejšiu obsluhu a poskytuje širšie možnosti uplatnenia počítača.

Ing. Tibor Guliš



*Ing. Anton KAJÁNEK*  
Solisko III-33  
059 84 Vyšné Hágy

# OSOBNÝ MIKROPOČÍTAČ PMD 85

NÁVOD NA POUŽITIE A OBSLUHU  
MIKROPOČÍTAČA PMD 85-2

**TESLA** BRATISLAVA, k.p. 



## ÚVOD.

Konštrukcia mikropočítača, ako i starostlivosť venovaná výrobe a vývoju, pri správne vykonávanej obsluhu, zaručujú jeho vysokú prevádzkovú spoľahlivosť.

Pri obsluhu mikropočítača sa riadte podľa pokynov uvedených v užívateľských príručkách.

Preto je vo Vašom záujme, aby ste si užívateľské príručky pred uvedením mikropočítača do činnosti pozorne preštudovali.



## ZMENY NA OSOBNOM POČÍTAČI PMD 85-2

### A. ÚVOD



Osobný počítač PMD 85-1 prešiel výrobnou inováciou, čo sa prejavilo hardverovými a softverovými úpravami. Z hardverových úprav má PMD 85-2 novú klávesnicu na báze vodivej gumy, čím sa zjemnil jej chod a zvýšila sa jej trvanlivosť. Ďalšia úprava sa týka snímania údajov z magnetofónu. Signál zo vstupného tvarovača je softverovo analyzovaný, čo umožnilo vyriešiť fázovo nezávislé snímanie nahrávok. Zo softverových úprav má PMD 85-2 nový monitor, ktorý je do značnej miery kompatibilný so starým (zmeny v monitore sú opísané v ďalšej časti textu) a novú verziu BASIC-G.

### B. OBSLUHA

Po zapnutí PMD 85-2 sa na rozdiel od predchádzajúceho typu neohlási vždy monitor, ale sa vloží do pamäti obsah založeného programového modulu. To znamená, že ak zapnete počítač so založeným modulom BASIC-G V2.0, po zapnutí sa ohlásí priamo interpreter BASICU. Ak nie je založený žiadny modul, ohlásí sa monitor správou:

++ OS READY ++




### C. OPIS KLÁVESNICE

Klávesnica sa od predchádzajúceho typu líši použitím nových tlačidiel. Tieto tlačidlá majú mäkký chod, preto sa pri stlačení ľubovoľného klávesu (okrem  a ) generuje krátke akustické návestie ("cvaknutie"). Ak pridržíme nejaký kláves dlhšie, tak sa bude jeho funkcia opakovať (tzv. autorepeat).













Tabuľka znakov obsahuje všetky znaky podľa ISO-7. Ak chceme vypísať znak, ktorého kód je mimo intervalu 32-127,








zobrazí sa znak □ ("malý štvorček").



Písmená sa píšú rovnako ako na predchádzajúcom type. Rozdiel je len v tom, že ak stlačíme kláves  a spolu s ním nejaké písmeno, zobrazí sa nám malé písmeno: napr. ak stlačíme naraz  a , napíšeme: malé p.



### C.1. RIADIACE KLÁVESY



Riadiace klávesy , , , , , , , , , , majú nezmenené funkcie. Kláves  nastaví kurzor za posledný nemedzerový znak v riadku. Kláves  má rovnakú funkciu, ale po výpise správy netreba stlačiť ešte niečo "naprázdno".

Väčšina riadiacich klávesov má aj druhú funkciu, ktorá sa dosahuje súčasným stlačením príslušného klávesu a klávesy  (SHIFT). Význam druhých funkcií jednotlivých riadiacich klávesov je takýto:



  - pri stlačení sa zobrazí znak, ktorý nasleduje v kódovej tabuľke ASCII za tým znakom, na ktorom je kurzor, pričom sa kurzor neposunie. Ak teda stlačíme   na medzere, objaví sa nám !, ak kláves stlačíme znovu, alebo ho pridržíme, zobrazujú sa postupne znaky "# \$ % atď. v poradí, ktoré je dané ich ASCII kódom. Tento kláves je určený na to, aby sa dali dosiahnuť aj tie znaky, ktoré nie sú na klávesnici PMD (napr. vlnovka).



  - stlačením tohoto klávesu prejde počítač do režimu terminálu. Tento režim sa používa vtedy, keď je počítač PMD 85-2 pripojený na počítače radu SMEP (bližšie v samostatnom odseku).



  - "Teplý štart systému". Týmto klávesom sa dostaneme z ľubovoľného stavu, keď sa žiada vstup znaku späť do monitora. Vypíše sa ++ OS READY ++.



  - odštartuje program, ktorý je v pamäti od adresy





0000. Je to teda ekvivalent príkazu JUMP 0000. Ak sme mali v pamäti BASIC-G (a ten sme opustili napr. stlačením   ), znovu sa nám naštartuje.

  - zmaže údaj na obrazovke.


  - zmení podklad na inverzný a vymaže údaje na obrazovke. Ak teda po zapnutí stlačíme tento kláves, celá obrazovka sa zafarbí na bielo a budú sa písať čierne písmená. Druhým stlačením sa dosiahne zase čierna obrazovka.

  - zasvieti sa ľavá (žltá) dióda led a tým sa vypne akustická signalizácia. Po opätovnom stlačení dióda zhasne a akustická signalizácia sa obnoví.





  - slúži na nastavenie spôsobu vypisovania na obrazovku. Pri stláčaní tohto klávesu sa cyklicky prepína vykresľovanie plným jasom, polojasom, blikanie plným jasom a blikanie polojasom.

## C.2. PROGRAMOVÉ KLÁVESY

V používaní programových klávesov (klúčov) sú tieto zmeny:

- pri zápise do klávesu (pomocou  ) sa ukladá len časť riadku vľavo od kurzora, po zápise sa ozve tón a obsah dialógového riadku sa nezmení. Je možné zapísať aj reťazce, ktoré sa končia medzerami.
- ak stlačíme klúč, ktorý má nulový obsah, nevykoná sa žiadna činnosť (nevypisuje sa chyba).

## D. PRÍKAZY MONITORA

V PMD 85-2 neexistuje príkaz BASIC-G, BASIC sa spúšťa stlačením   alebo   (ak už bol raz v pamäti). Ostatné príkazy ostali zachované, ale pri ich vykonávaní sú určité rozdiely.



V prvom rade vo všetkých príkazoch nie je povinná medzera medzi kľúčovým slovom a parametrami. Ostatné rozdiely uvedieme v nasledujúcich odsekoch.

PRÍKAZ SUB - Ak ne zadáme žiadne dáta, len adresu, nehlási sa chyba, ale príkaz zostane v dialógovom riadku.

PRÍKAZ MEM - Má pozmenený tvar výpisu. Na začiatku riadka nevynechá voľné miesto, ale priamo tam napíše príkaz SUB s príslušnou adresou.

PRÍKAZ DUMP - Vypisuje obsah pamäti bez prestávok a otázok continue?. Výstup je kedykoľvek možné pozastaviť pridržaním klávesu  alebo úplne zastaviť klávesom .

PRÍKAZ JUMP - Je bez zmien.

PRÍKAZ MGSV - Číslo nahrávky môže byť 00 až 99 (platí aj pre MGEND a MGLD).



PRÍKAZ MGEND - Vykonávanie tohto príkazu je možné hocikedy prerušiť stlačením klávesu  (aj v prípade, že nie je pripojený magnetofón).

PRÍKAZ MGLD - Je naprogramovaný odlišným spôsobom tak, aby bolo možné čítať záznamy aj z menej kvalitných magnetofónov a aby bolo možné čítať záznamy, ktoré boli nahraté na inom type magnetofónu. Tieto úpravy by sa mali prejavovať vyššou spoľahlivosťou nahrávok programov a dát. Pri jeho vykonávaní sa vypisujú mená súborov, ktoré sa našli na páske aj do obrazovej časti obrazovky, teda ak chceme získať "obsah" pásky, stačí napísať napr. MGLD 99 (ak nemáme na páske záznam s takýmto číslom) a na obrazovke sa nám postupne vypíše obsah pásky. Vykonávanie príkazu MGLD je možné v ľubovoľnom okamihu prerušiť klávesom . Pri takomto prerušení sa vypíše správa ++ FILE ERROR ++.

PRÍKAZ JOB - Je bez zmien.



## E. MODUL TERMINÁL

Do tejto časti monitoru sa vstupuje stlačením klávesov   . Táto časť monitoru umožňuje použiť osobný počítač PMD 85-2 ako terminál počítača. Spojenie s počítačom sa uskutočňuje cez asynchrónne sériové prúdové slučky, pričom obidve linky sú pasívne. Rýchlosť prenosu je štandardne nastavená na 4800 Baudov. Možno ju meniť zmenou hodnoty premennej SPEED (0C07CH). Hodnota premennej SPEED sa vypočíta pomocou vzorca

$$\text{SPEED} = 128000/\text{BAUD},$$

kde BAUD je prenosová rýchlosť. Na pripojenie k počítaču slúži konektor pre V.24.

Komunikačný modul terminál má okrem štandardných funkcií terminálu aj tri nové funkcie, ktoré umožňujú blokový prenos údajov medzi nadradeným počítačom a PMD 85-2. Sú to príkazy INBLOCK, OUTBLOCK a JUMP.

INBLOCK - Vstup ( do PMD ) bloku údajov.

Príkaz má dĺžku 131 Byte. Prvý Byte má hodnotu 4, ďalšie dva Byte určujú adresu v pamäti (v PMD), od ktorej sa bude nasledujúcich 128 Byte ukladať.

OUTBLOCK - Výstup bloku údajov.




Príkaz má dĺžku 3 Byte. Prvý Byte má hodnotu 2, ďalšie dva určujú adresu v pamäti, od ktorej sa nasledujúcich 128 Byte vyšle na výstupnú linku. Po nich bude nasledovať Byte súčtu týchto (128) Byte.

JUMP - Skok na určenú adresu.

Príkaz má dĺžku 3 Byte. Prvý Byte má hodnotu 5, ďalšie dva určujú adresu, kam sa vykoná skok s uložením návratovej adresy na vrch zásobníka.

V režime terminálu sa menia kódy (a funkcie) riadiacich klávesov. Kódy riadiacich klávesov sú uvedené v nasledujúcej tabuľke. V prvom stĺpci je kód, v druhom príslušný kláves.



↑ pred názvom klávesu znamená, že treba zároveň držať kláves  
 . Dšipka označuje šipku s čiarkou pri hrote (dvojitá šipka)  
 alebo  .

Kód	Kláves	Kód	Kláves
1	↑ CLR	17	↑ šipka doprava
2		18	↑ DEL
3	END	19	↑ šipka doľava
4		20	
5		21	
6	↑ C-D	22	
7	↑ WRK	23	RCL
8	šipka doľava	24	šipka doprava
9	↑ Dšipka doprava	25	Dšipka doprava
10	↑ Dšipka doľava	26	Dšipka doľava
11	WRK	27	CLR
12	šipka šikmo	28	INS
13	EOL	29	
14		30	C-D
15	↑ INS (PTL)	31	↑ šipka šikmo
16	↑ END	127	DEL

Kódy, ktoré nie sú v tabuľke, nedajú sa z klávesnice dosiahnuť.

Modul terminál bol vyvinutý s prihliadnutím na počítače radu SMEP (SM-3-20, SM-4-20, SM 52/11). To však nevylučuje jeho použitie v spojení s inými počítačmi.

Po naštartovaní modulu terminál sa vysielajú do počítača dva znaky: CTRL Q a CR (kódy 17 a 13). Slúži to na prípadné odblokovanie prenosovej linky pri počítačoch SMEP. Ďalej sa PMD 85-2 používa ako terminál počítača. Pri rolovaní obrazovky terminál vysielá kódy CTRL S pred začiatkom rolovania a CTRL Q po skončení. Od počítača sa vyžaduje, aby po každom znaku CR (kód 13) posielal niekolkú (3-5) výplňových znakov (kód 0).

Softverové pripojenie na počítače radu SMEP pod operačným systémom DOS RV-V 2 je možné takto:

Predpokladajme, že hardverovo je PMD-85 pripojený ako TTX: (kde X je číslo terminálu). Ak je pripojený cez AMUX,



rýchlosť treba nastaviť takýmto riadiacim príkazom DOS RV:

SET/SPEED = TTX : 4800 : 4800

Ak je pripojený cez ASAD, je nutné nastaviť na ASAD-e prepínačmi rýchlosť 4800 BD.

Potom je nutné nastaviť tieto charakteristiky terminálu:

SET / C2111 = TTX:

SET / CRT = TTX:

SET / BUF = TTX:48.

Charakteristika C2111 zabezpečuje vysielanie výplňových znakov. V prípade nastavovania charakteristík v obraze systému na disku pomocou programu VMR sa miesto C2111 píše LA30S.

#### G. ZMENY V BASIC-G V2.0 OPROTI V1.0.

Táto príručka opisuje zmeny a doplnky v BASIC -G V2.0, ktorý sa dodáva v PMD 85-2. Treba ju chápať ako doplnok k príručke o BASICU-G. Od čitateľa sa predpokladá, že sa s jej obsahom už oboznámil.

BASIC-G V2.0 je navrhnutý tak, že všetky programy, ktoré pracovali pod BASIC-G V1.0 a nepoužívali služby monitora ne- uvedené v príručke, ani nepoužívali zvláštne vedomosti o moni- tore a interpretri BASICU, budú pracovať aj pod V2.0 programy, ktoré využívajú príkazy PEEK, POKE, CODE, USR, je nutné v nie- ktorých prípadoch pozmeniť. Tieto zmeny môžu byť niekedy mini- málne-zmena adresy, ale v niektorých prípadoch zložité - ak program používal špeciálne podprogramy monitora alebo BASICU.

Zmeny v BASICU-G V2.0 sú takéhoto druhu:

- 1) Opravy chýb a drobné zlepšenia verzie 1.0
- 2) Doplnenie nových možností už existujúcich príkazov
- 3) Doplnenie nových príkazov.

Okrem týchto zmien sa v niektorých prípadoch zmenilo aj "správanie" prekladača BASICU (napr. pri vykonávaní povelu LIST). Tieto zmeny sú len dôsledkom inej interpretácie nie-



ktorých povelov v monitore PMD 85-2.

## G.1. ODSTRÁNENIE CHÝB A DROBNÉ ZLEPŠENIA

Verzia V2.0 má opravené a doplnené tieto vlastnosti:

- Za grafickými príkazmi (PLOT, MOVE, AXES) môžu v jednom riadku nasledovať ďalšie príkazy, ktoré sú oddelené dvojbodkou.
- Premenné X0, X1, X2, X3 a Y0, Y1, Y2, Y3, ktoré sa používali vo V1.0 na podporu grafických príkazov, sa vo V2.0 na tento cieľ nevyužívajú. Preto je možné používať ich ako bežné premenné aj vtedy, keď program používa grafické príkazy.
- Funkcia INKEY spracúva aj kľúč číslo 11.
- Návestia príkazov môžu byť len od 0 do 32767.
- Goniometrické funkcie (sin, cos, tang), ktoré sa vplyvom chyby vo V1.0 počítali s presnosťou na 3 desatinné miesta sa vo V2.0 počítajú presne na plných 6 až 7 desatinných miest.
- V príkazoch ON výraz GOTO, ON výraz GOSUB môže byť hodnota výrazu v intervale -32768 až 32767.
- Povel MONIT bol odstránený, je nahradený stlačením kombinácie klávesov ↑ a RCL.

## G.2. ROZŠÍRENIE MOŽNOSTÍ EXISTUJÚCICH PRÍKAZOV

### G.2.1. PRÍKAZ POKE

Príkaz POKE môže mať takýto tvar:

POKE ADR, H1, H2, H3, ....., HN,

kde význam tohto príkazu je taký, že hodnoty H1, H2, H3 až HN sa uložia do pamäti od adresy ADR.

napr. príkaz:

POKE '7000, 1, 2, 3, 4

je ekvivalentný príkazom:

POKE '7000, 1: POKE '7001, 2: POKE '7002, 3: POKE '7003, 4



Pozn.: Apostrof (') pred číslami znamená zápis v šestnástkovej sústave.

### G.2.2. PRÍKAZ PLOT

V príkaze PLOT je možné zadávať súradnice viacerých bodov, ktoré sú oddelené bodkočiarkami:

PLOT X1, Y1; X2, Y2; .....; XN, YN

Takýto príkaz nakreslí lomenú čiaru, ktorá sa začína v bode X1, Y1 a končí sa v bode XN, YN. Takto je možné jedným príkazom nahradiť niekoľko za sebou idúcich príkazov PLOT, napr. Príkaz:

PLOT 1, 1;1, 2;2,2; 2,1; 1,1

nakreslí štvorec a príkaz:

PLOT 1,1,1;1,2,1;2,2,1;2,1,1

nakreslí štyri body vo vrcholoch štvorca.

### G.2.3. POVEL LIST

Pri vykonávaní povelu LIST sa riadky, ktoré majú viac než 48 znakov vypisujú do dvoch riadkov obrazovky. Pritom začiatok druhého riadku je odsunutý tak, aby bolo "vysunuté" číslo riadku.

V prípade, že chceme zmeniť dĺžku vypisovaného riadku (na inú než 48), môžeme to urobiť príkazom:

POKE '2E, DL,

kde DL je dĺžka riadku.

Napr. príkazom:

POKE '2E, 80

dosiahneme, že výpis programu pri LIST bude rovnaký ako vo V1,0, lebo sme nastavili dĺžku riadku na 80 znakov.



Naopak po príkaze:

```
POKE '2E, 32
```

bude výpis široký len 32 znakov.

POZOR !!! - Príkazmi uvedenými vyššie sa mení nielen šírka výpisu programu, ale aj šírka výpisov počas vykonávania programu (príkazom PRINT). Pretože vo V1.0 je bežne nastavená táto šírka na 80 a vo V2.0 na 48, niektoré programy sa môžu správať inak pri výpisoch.

#### G.2.4. PRÍKAZ PRINT

Príkaz PRINT je rozšírený o funkcie AT a INK. Tieto funkcie umožňujú lepšie riadiť výpis textov na obrazovku.

##### G.2.4.1. FUNKCIA AT

Funkcia AT umožňuje nastavenie miesta na obrazovke, kam sa vypíše nasledujúci text v príkaze PRINT. Funkcia má dva parametre: riadok (0 až 26) a stĺpec (0 až 47). Parametre sú oddelené čiarkou. Ak sa funkcia AT vyskytne ako posledná v príkaze PRINT, tento príkaz sa chápe ako "neukončený" (t.j. akoby bola na jeho konci bodkočiarka).

Napríklad takýto program:

```
10 FOR I=0 TO 26  
20 PRINT AT I, 2*I, "*"   
30 NEXT I
```

nakreslí na obrazovke uhlopriečku (približnú) z hviezdíček.

##### G.2.4.2. FUNKCIA INK

Funkcia INK umožňuje ovládať jas a blikanie vypisovaného textu.

Funkcia má jeden parameter, ktorý sa píše do okrúhlych zátvoriek. Hodnotu parametra dostaneme ako súčet niektorých z týchto hodnôt.



- 1 - výpis bude polojasom
- 2 - výpis bude blikat
- 4 - výpis bude inverzný (na bielom podklade čierne písmená)

Na začiatku práce je nastavenie INK (0), teda píše sa plným jasom, neblikajúco na čiernom podklade biele písmená.

Napríklad príkaz:

```
PRINT AT 10,5; INK (3); "BASIC"; INK (4); "V2.0"; INK (0)
```

vypíše do 10. riadku a 5. stĺpca polojasom blikajúce (1+2) slovo "BASIC" a plným jasom inverzne text "V2.0".

Funkcia INK nastavuje rovnaké charakteristiky ako prvé tri voľby v príkaze PEN.

#### G.2.4.3. RIADENIE ŠÍRKY RIADKU

Bežne je šírka riadku nastavená na 48 znakov, to znamená, že ak vypisujeme za sebou písmená napr. takýmto programom:

```
10 FOR I=1 TO 50
20 PRINT CHR$(32+I);
30 NEXT I
```

dostaneme takýto výpis:

```
! "#$ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ ABCDEFGHIJKLMNOP
QR
```

Teda po 48. znaku výpis pokračuje na novom riadku.

Ak chceme zmeniť šírku riadku, postupujeme rovnako ako pri šírke výpisu programu (pozri 2.3.), teda program

```
2 POKE 2E,32
10 FOR I=1 TO 50
20 PRINT CHR$(32+I);
30 NEXT I
```

vyprodukuje takýto výstup:

```
! "#$ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ ABCDEFGHIJKLMNOPQR
```



### G.2.5. PRÍKAZY GOTO A GOSUB

V príkazoch GOTO a GOSUB môže na mieste čísla riadku vystupovať aj výraz, ktorého hodnota bude z intervalu 0 až 32767. Nový príkaz GOTO, resp. GOSUB má teda tvar

#### GOTO VÝRAZ

kde na výraz sa kladú tieto obmedzenia:

- buď je to celé číslo,
- alebo je to aritmetický výraz, ktorý sa nezačína cifrou.

Teda nemôžeme napísať GOTO 10\*X, môžeme však napísať GOTO X\*10 alebo GOTO (10\*X).

Výhodu tohto rozšírenia oceníme vtedy, keď máme program, ktorý pozostáva z viacerých v podstate samostatných úsekov (podprogramov). Napríklad nech súčasťou nášho programu sú podprogramy, ktoré hýbu nejaký objekt po obrazovke dopredu a dozadu. V tomto prípade stačí, ak na začiatok programu napíšeme priradenia podobné týmto:

VPRED = 100 : VZAD = 200

Potom už počas písania (a ladenia) celého programu si nemusíme pamätať čísla riadkov, na ktorých sa začínajú jednotlivé podprogramy, stačia nám ich názvy, teda môžeme písať

GOSUB VPRED

alebo

ON SGN (X) + 1 GOSUB VPRED, VZAD

Okrem toho, že sme odbremenení od pamätania si čísiel riadkov, môžeme ľahko premiestniť jednotlivé podprogramy aj na iné čísla riadkov, stačí zmeniť len začiatkové priradenie príkazy. Treba si všimnúť, že "názvy" podprogramov sa musia líšiť v prvých dvoch písmenách (lebo sú to vlastne obyčajné názvy premenných) a mená premenných nesmú obsahovať kľúčové slová BASICU.



Rozšírenie príkazov GOTO a GOSUB je možné použiť aj inak, napr. tak, ako to vidíme v nasledujúcom programe:

```
10 FOR I=1 TO 10
20 GOTO (I-INT(I/2) * 2) * 10+30
30 PRINT I
40 NEXT I
```

Tento program vypíše všetky párne čísla od 1 do 10. Takýto spôsob použitia výrazov v príkazoch GOTO/GOSUB však neodporúčame, lebo program sa môže stať veľmi neprehľadný a ťažko modifikovateľný.

#### G.2.6. FUNKCIA VAL

Funkcia VAL je rozšírená tak, že jej argument je znakový reťazec, ktorý obsahuje ľubovoľný výraz BASICU. Funkcia VAL vráti vypočítanú hodnotu tohto výrazu.

Napríklad nasledujúci program si vyžiada funkciu premennej X a vytlačí tabuľku jej hodnôt od 1 do 10 s krokom 1:

```
10 DISP "ZADAJ FUNKCIU PREMENNEJ X" : INPUT F$
15 GCLEAR:PRINT "X", F$
20 FOR X=1 TO 10
30 PRINT X, VAL (F$)
40 NEXT X
```

Ak tento program spustíme a zadáme mu reťazec  $X * X$ , dostaneme tabuľku hodnôt funkcie  $X * X$ :

X	$X * X$
1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81
10	100



Ak zadáme reťazec  $2 * X+1$ , dostaneme tabuľku funkcie  $2 * X+1$ :

X	$2 * X+1$
1	3
2	5
3	7
4	9
5	11
6	13
7	15
8	17
9	19
10	21

#### G.2.7. PRÍKAZ PAUSE

Príkaz PAUSE nemusí mať žiadny parameter. Potom sa vykonáva rovnako ako PAUSE  $\emptyset$ .

#### G.2.8. POVELY SAVE A LOAD - MOŽNOSŤ CHRÁNENIA PROGRAMOV

BASIC-G V2.0 obsahuje jednoduchú možnosť chránenia programov proti prezeraniu a kopírovaniu. Program, ktorý chceme mať chránený, treba nahráť na pásku tak, aby prvý znak jeho mena bol " \* " (hviezdička). To dosiahneme takýmto povelom:

SAVE CC" \* MENO",

kde CC je číslo nahrávky a MENO je max. sedem-znakové meno programu. Všímajte si, že medzi číslom a hviezdíčkou nesmie byť čiarka.

Keď sa povelom LOAD prečíta program, ktorého meno sa začína hviezdíčkou, ihneď sa spustí, akoby bol ihneď po skončení zadáný povel RUN. Program nie je možné zastaviť klávesom STOP. Ak prerušíme program pomocou klávesu RST, t.j. odštartujeme BASIC odznova, program sa vymaže.


Z uvedeného vyplýva, že program, ktorého meno sa začína hviezdíčkou, je chránený proti bežným spôsobom špionáže. Netvrdíme, pravda, že je to ochrana úplne bezpečná.



### G.2.9. POVEL AUTO

Povel AUTO môže mať len jeden parameter. V tom prípade sa za druhý dosadí 1Ø (táto možnosť bola aj v predchádzajúcom type, ale sa hlásila chyba, aj keď sa povel správne vykonal).

### G.2.10. POZASTAVENIE PROGRAMU

Kláves  (SHIFT) slúži na pozastavenie vykonávania programu alebo výpisu programu. Ak počas behu programu alebo vypisovania programu (povel LIST) tento kláves stlačíme, vykonávanie programu (alebo výpis) sa nám zastaví. Ak kláves pustíme, činnosť pokračuje.

## G.3. NOVÉ PRÍKAZY A FUNKCIE

### G.3.1. ŠESTNÁSTKOVÉ KONŠTANTY

BASIC-G V2.Ø dovoľuje zapisovať v texte programu konštanty v šestnástkovej sústave. Zapisujú sa tak, že pred vlastné číslo sa zapíše znak ' (apostrof). Rozsah konštánt je od 'Ø do 'FFFF. Šestnástkové konštanty nám dobre poslúžia ako parametre príkazov POKE a APOKE, príp. funkcií PEEK, APEEK a USR.

Pozn.: V tejto príručke zapisujeme šestnástkové konštanty spôsobom vysvetlením vyššie.

### G.3.2. NOVÉ FUNKCIE

#### G.3.2.1. FUNKCIA HEXØ

Funkcia HEXØ prevedie celú časť svojho číselného argumentu do tvaru reťazca v šestnástkovej sústave. Výsledkom funkcie je vždy reťazec dĺžky 4, t.j. ak je číslo menšie než '1ØØØ, je zľava doplnený nulami, ak je číslo väčšie než 'FFFF, je hlásená chyba, funkcia HEXØ je v istom zmysle analógická funkcii STRØ.



Napr. príkaz

```
PRINT HEX$ (243); " "; HEX$ ('C03A)
```

Vypíše

```
00F3 C03A
```

V niektorých prípadoch je vhodná funkcia opačná k HEX\$, t.j. taká, že prevedie zadaný reťazec, ktorý obsahuje číslo v šestnástkovej sústave do tvaru čísla. V tomto prípade nám posluží funkcia VAL tak, ako vidíme v nasledujúcom programe v riadku 30.

```
10 DISP "ZADAJ CISLO V SESTNASTKOVEJ SUSTAVE"  
20 INPUT A$  
30 C=VAL (" " + A$)  
40 PRINT "V DESIATKOVEJ SUSTAVE JE TO "; C  
50 GOTO 10
```

#### G.3.2.2. FUNKCIA APEEK

Výsledkom funkcie APEEK (A) je hodnota dvoch Byte v pamäti PMD 85, ktoré sa chápu ako 16-Bitové číslo so znamienkom. Funkcia APEEK takto nahrádza použitie funkcie PEEK na dve za sebou nasledujúce adresy.

#### G.3.2.3. FUNKCIA ADR

Výsledkom funkcie ADR je adresa jej argumentu.  
Presnejšie:

- ADR (A), kde A je číselná premenná, vráti adresu, na ktorej je uložená hodnota (4 Byte) premennej A.
- ADR (A\$), kde A\$ je reťazcová premenná, vráti adresu do pamäti, kde sa nachádza 4 Byte informácia o hodnote A\$. Prvý Byte je dĺžka (0 až 255), druhý Byte sa nepoužíva a tretí a štvrtý Byte udávajú adresu reťazca A\$ v pamäti.

Napríklad nasledujúca časť programu:



```

100 DLZKA=PEEK (ADR(A$))
110 A=APEEK (ADR(A$)+2)
120 FOR I=1 TO DLZKA
130 PRINT CHR$(PEEK (A+I-1))
140 NEXT I

```

Vytlačí obsah premennej A\$.

Pomocou funkcie ADR sa ľahšie programuje styk medzi programom v BASICU a programami v strojovom kóde, lebo BASIC program môže odovzdať adresu svojej premennej ako parameter.

Iná možnosť použitia funkcie ADR je v prípade, keď potrebujeme vyhradiť nejaký kus pamäti tak, aby sme v ňom mohli pracovať pomocou PEEK a POKE a pritom nám nevyhovuje oblasť nad 700.

Napríklad nech potrebujeme 4000 Byte oblasť. Tú si vyhradíme takto:

```

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

```

Po vykonaní týchto príkazov máme v premennej AD adresu, kde sa začína "NAŠICH" 4000 Byte, ktoré môžeme využívať.

### G.3.3. NOVÉ PŘÍKAZY

#### G.3.3.1. PŘÍKAZ PEN

Grafický príkaz PEN nastaví charakteristiky pre grafické príkazy (PLOT, AXES, LABEL a FILL). Príkazom PEN môžeme riadiť jednak jas a blikanie čiar rovnako ako funkciou INK v príkaze PRINT, jednak spôsob vykresľovania čiar, ktorý sa vo V1.0 riadil pomocou príkazov POKE -15878, .....

Príkaz PEN má tvar

PEN výraz



kde hodnota výrazu je súčtom niektorých z nasledujúcich hodnôt:

- 1 - vykreslenie polojasom
- 2 - vykreslenie blikaním
- 4 - kreslenie inverzne /čierne na bielom/
- 8 - vykreslenie negovaním /kade sa kreslí, tade sa mení obsah obrazovky/
- 16 - vykreslenie mazaním /kade sa kreslí, tade sa maže obsah obrazovky/.

Ak nie je v súčte ani 8, ani 16, vykresľuje sa nastavovaním, t.j. kade sa kreslí, tade sa nastavuje obrazovka. Ak je v súčte 8 aj 16, berie sa len 16.

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. Ak je v súčte 4, vymieňa sa funkcia podkladu a čiar, podklad bude biely (príp. polojasom, či blikajúci podľa nastavenia 1 alebo 2) a kreslí sa čiernou. Teda nastaviť znamená urobiť čiernym a zmazať znamená urobiť bielym.

Vykresľovanie nastavovaním znamená, že tie body na obrazovke, kadiaľ kreslíme (napr. čiaru), sa nastavujú bez ohľadu na to, aký mali stav pred kreslením. Pri vykresľovaní mazaním sa príslušné body zotrú. Vykresľovanie negovaním znamená, že vykresľovaným bodom sa mení stav zo svietiaceho na nesvietiaci a naopak. Ak teda v tomto režime nakreslíme dvakrát tú istú čiarku (aj tým istým smerom), zotrie sa.

Pri zapnutí počítača alebo stlačení RST sa nastavuje implicitne PEN 8, t.j. vykresľovanie negovaním plným jasom bez blikania, biele na čiernom podklade.

Napríklad takýto program:

```
10 SCALE 0,255,0,242
20 FOR I=2 TO 230
30 PEN I - INT (I/4) * 4
50 PLOT I,I,1;I, 234-I,1
60 NEXT I
```



Nakreslí dve uhlopriečky obrazovky, pričom sa na každej čiare cyklicky striedajú body s plným jasom, polojasom, blikajúce plným jasom a blikajúce polojasom.

Pozn.: - Nastavenie blikania, polojasu a inverznej obrazovky (čierne na bielom) je spoločné s funkciou INK.

#### G.3.3.2. PRÍKAZY DEG A RAD

Príkazy DEG a RAD slúžia na prepínanie režimu výpočtu goniometrických funkcií SIN, COS a TANG. Prepnutie do režimu "STUPNE" dosiahneme príkazom

##### DEG,

to znamená, že funkcie SIN, COS, TANG sa počítajú tak, že ich argumenty sa chápu v stupňoch.

Prepnutie do režimu "RADIANY" dosiahneme príkazom

##### RAD.

Po vykonaní tohto príkazu sa argumenty vyššie uvedených funkcií chápu v radiánoch.

Po spustení programu povelom RUN, ako aj po použití povelu CLEAR, sa nastavuje režim "RADIANY".

Využitie prepínača DEG si ukážeme na nasledujúcom programe:

```
10 SCALE -1,1,-1,1: GCLEAR:DEG:I=0
20 PLOT ,      COS (I), SIN (I):I=I+7:GOTO 20
```

Tento program kreslí polomery kružnice, ktorých vzájomný uhol je 7 stupňov, keď necháme program dlhšie bežať (po 7 "obehoch"), úsečky sa začnú postupne premazávať, až sa (po 14 "obehoch") všetky zotrujú. Keby sme tento istý program napísali s použitím radiánov



```

10 SCALE -1,1,-1,1: GCLEAR:RAD
20 K=7 * 3.1415927/180:I=0
30 PLOT 0,0,1: COS (I), SIN (I): I=I+K:GOTO 30

```

vplyvom zaokrúhľovacích chýb sa nám pri výpočte nebudú čiary presne zotierať.

### G.3.2.3. PRÍKAZ ON ERR

Tento príkaz umožňuje potlačiť hlásenie chýb a ošetrovanie chýb užívateľským programom v BASICU. To znamená, že v prípade výskytu chyby sa vykonávanie programu nepreruší, ale pokračuje na určenom mieste.

Tvar príkazu

#### ON ERR PRÍKAZ

Vykonávanie tohto príkazu spočíva v tom, že sa nastaví režim nezastavovania na chybe a zvyšok riadku sa ignoruje. Od vykonania tohto príkazu platí, že v prípade výskytu chyby sa najprv vyprázdni zásobník návratových adries (return) a zásobník rozrobených cyklov FOR. Potom sa pokračuje vo vykonávaní programu na príkaze, ktorý nasleduje za ON ERR. Pritom sa však zruší stav nezastavovania na chybe, t.j. ak vznikne ešte chyba, program sa zastaví.

Rozoberme si teraz takýto jednoduchý program:

```

5 ON ERR GOTO 50
10 DISP "ZADAJ X"
20 INPUT X
30 PRINT X,1/(X * X-9)
40 GOTO 10
50 REM === SPRACOVANIE CHYBY ===
60 DISP "CHYBNÝ VSTUP, ZADAJ ZNOVA"
70 ON ERR GOTO 50
80 GOTO 20

```



V riadku 5 sa nachádza príkaz ON ERR. Pri prechode týmto príkazom sa nastaví stav nezastavovania na chybe, zvyšok riadku sa ignoruje a pokračuje sa v riadku 10, kde sa vypíše pomocný text a v riadku 20 sa číta číslo X, v riadku 30 sa vyhodnotí výraz  $1/(X * X - 9)$  a vypíše sa hodnota X a hodnota vypočítaného výrazu. Po výpise sa pokračuje znova riadkom 10.

Chyba môže nastať pri zadávaní riadku 20 alebo pri vyhodnocovaní výrazu v riadku 30. Pri výskyte chyby sa začne vykonávať zvyšok riadku 5, kde je GOTO 50. V riadku 60 je ohlásenie chyby, riadok 70 nastavuje znova stav nezastavovania na chybe a riadkom 80 sa končí spracovanie chyby.

V predchádzajúcej ukážke bola situácia jednoduchá, pretože ľubovoľná chyba mohla vzniknúť len nesprávnym zadáním hodnoty premennej X. Spracovanie chyby sa mohlo obmedziť na tento jednoduchý prípad. V zložitejších situáciách je užitočné vedieť nielen to, že chyba vznikla, ale aj druh chyby a miesto jej výskytu. Tieto informácie je možné získať takto:

- číslo riadku, kde vznikla chyba je uložené na adrese '5E5D ako dva Byte. Na jeho zistenie teda môžeme použiť napríklad takýto príkaz:

```
RIAD=APEEK ('5E5D)
```

- kód chyby je uložený v jednom Byte na adrese '26, môžeme ho prečítať napr. takto:

```
KODCH=PEEK ('26)
```

V nasledujúcej tabuľke sú uvedené chyby, ktoré prislúchajú k jednotlivým kódom:

KÓD CHYBY	PRÍSLUŠNÁ CHYBOVÁ HLÁŠKA
1	SUBSCR.RNG
2	ARR.ALLOC.
3	FNC.PARAM.



4	ONLY IN PG
5	NO FOR STM
6	DATA EXHAU
7	PG TOO BIG
8	OVERFLOW
9	SYNTAX ERR
10	RETURN ERR
11	NUMB. NONEX
12	DV BY ZERO
13	CAN'T CONT
14	STRNG LONG
15	NO STR. SPC
16	STR.ALGRTH
17	TYPE CONV.
18	FILE SMALL
19	INPUT ERR
20	FIELD LOST
21	FILE BOUND
22	STOP
23	FILE ERROR

Tieto nové vedomosti môžeme použiť na zlepšenie predchádzajúceho programu tak, že vypisujeme rôzne chybové správy podľa toho, aká chyba nastala:

```

5 ON ERR GOTO 50
10 DISP "ZADAJ X"
20 INPUT X
30 PRINT X,1/(X * X-9)
40 GOTO 10
50 REM === SPRACOVANIE CHYBY ===
55 ER=PEEK (^26)
57 IF ER=8 THEN DISP "PRETECENIE, ZADAJ ZNOVA" : GOTO 70
58 IF ER=12 THEN DISP "DELENIE 0, ZADAJ ZNOVA" : GOTO 70
59 IF ER=19 OR ER=20 THEN DISP "CHYBNY VSTUP, ZADAJ ZNOVA":
    GOTO 70
60 DISP "CHYBA"; ER; "V RIADKU " ; APEEK (^5E5D); "ZADAJ
    ZNOVA"
70 ON ERR GOTO 50
80 GOTO 20

```



### DÔLEŽITÉ UPOZORNENIE

U novších verzií počítačov PMD 85-2 je výber sériového výstupu zabezpečený automaticky, v dôsledku čoho prepínač MG - V 24 nie je osadený.

Zapojenie vývodov konektorov K 2 až K 5 je vždy v súlade s návodom, bez ohľadu na skutočné označenie na teliesku konektora.



## NÁVOD NA POUŽITIE PMD 85 - 2A

Osobný mikropočítač PMD 85-2A je upravenou verziou typu PMD 85-2.

Operačná pamäť PMD 85-2A je vytvorená z dynamických pamätí s rozsahom 64 Kbitov, čo umožňuje plné obsadenie pamäťového priestoru mikropočítača.

Využitie plnej kapacity operačnej pamäti umožnila úprava programového vybavenia - interpretra jazyka BASIC-G, ktorého verzia BASIC-G/ V 2.A zvyšuje použiteľný rozsah programov na 19 KB/ pri nepoužití príkazu ROM na 22 KB / a zväčšuje priestor pre ukladanie znakových reťazcov na 11 KB.

Spôsob práce s PMD 85-2A:

1. Po zapnutí zdroja pracuje PMD 85-2A ako PMD 85-2, kapacita pamäti pre programy a data 19 KB.
2. Zvýšenie kapacity pre programy a data na 22 KB pri nepoužití príkazu ROM sa prevedie príkazom jazyka BASIC

APOKE '3D, '7D 0 0

3. Rozšírenie priestoru pre znakové reťazce na 11 KB je možné odštartovaním krátkeho programu v jazyku BASIC.

10 A\$ = "21FF8F46702B7CA7FA037F3E82D3F709"

20 CODE A\$

30 APOKE '41, 'BD00

40 CLEAR

4. Po odštartovaní programu v jazyku BASIC je možné pre tvorbu programov v hexadecimálnom kóde využiť aj oblasť 9000 - BFFFFH.

Pre PMD 85-2A sa mení mapa operačnej pamäti, ktorá je uvedená na str. 5 "Návodu na použitie osobného mikropočítača radu PMD", podľa priloženého obrázku.

## UPOZORNENIE

V osobnom mikropočítači PMD 85-2A v dôsledku aplikácie dynamických pamätí 64 KB je zvýšený odber zo zdroja + 5V o cca 0,3 A, čím nie všetky zdroje EA 1605 zabezpečujú správnu činnosť počítača.



ADRESA	OBSAH	ROZSAH	
0000 H	BASIC G.V.2.A	9. k	16 k
2400 H			
4000 H	PROGRAMY + DATA	19 k	16 k
7000 H			
7F00 H			
8000 H	PROGRAMOVÉ KLÚČE	3,8 k	
	ZÁSOBNÍK + DIAL. RIADOK + SPRÁVY	0,2 k	
9000 H	MONITOR	4 k	16 k
	REŤAZCE	11 k	
BD00 H			
	ZÁPISNÍK PROGRAMU BASIC	1 k	
C000 H	VIDEO 288 × 256 BITOV	ZÁPISNÍK O.S.	0,5 k
		UŽÍVATEĽOM PRÍSTUPNÁ OBLASŤ	15,5 k
			16 k

Ø BYTE

48 BYTE