

SHARP

KAPITOLA 6 PŘÍKAZY A INSTRUKCE JAZYKA BASIC POČÍTAČE MZ 800

Tato kapitola popisuje příkazy a instrukce překladače MZ-800 BASIC (1Z-016). Tyto příkazy a instrukce můžeme rozdělit do osmi skupin.

- . Základní příkazy
- . Základní instrukce
- . Instrukce rejstříku
- . Instrukce grafické
- . Instrukce pro ovládání hudebních signálů
- . Instrukce pro ovládání tisku
- . Instrukce strojového jazyka
- . Instrukce pro práci s chybami

Příkazy a instrukce jazyka BASIC pro režim MZ 700 jsou uvedeny v kapitole 9.

Při studiu příkazů a instrukcí je nutné používat tento návod spolu s originálem a to z toho důvodu, že z tiskových důvodů nebylo možno rozlišit přesně závorky hranaté, kulaté a ostré. V ostrých závorkách jsou uvedeny údaje, které musí specifikovat uživatel. Údaje v hranatých závorkách mohou být s instrukcí nebo příkazem použity, ale použity být nemusí. Pokud jsou použity složené závorky může být použit pouze jeden výraz. Při prvním použití závorek budeme dokumentovat a vysvětlíme všechna použití.

Na straně 6-3 je vyložen příkaz AUTO. Na tomto příkazu si ukážeme jak budeme vysvětlovat jednotlivé příkazy a instrukce. Pod pojmem format je ukázáno v jakém formátu je nutno zadávat příkazy a instrukce počítači. V překladu budou přeloženy obsahy závorek. Oále je uvedeno Abbreviated Format což je zkrácená forma příkazu nebo instrukce. Oále je uvedeno vysvětlení explanation a příklad example. Budeme příklady komentovat ve vysvětlení a pokud nebude nutné nebudeme je přepisovat.

6.1 PŘÍKAZY

AUTO

starting line number - číslo počátečního řádku
 increment - přírůstek

Příkaz AUTO generuje automaticky číslo řádku během zadávání programu. Pokud nejsou parametry zadány jsou automaticky 10. V příkladu 2 automaticky generovány čísla řádků s přírůstkem 5 od čísla 300. Příklad 3 generuje od čísla 100 řádky po 10 a příklad 4 generuje řádky s přírůstkem 20 od čísla řádku 10.

Poznámka : Vykonávání příkazu AUTO je ukončeno stisknutím tlačítek SHIFT a BREAK:

DELETE

starting line number - číslo počátečního řádku
 ending line number - číslo řádku, kde chceme ukončit
 line number - číslo řádku

Z programu vymaže řádky od řádku s číslem počátečního řádku až k řádku s číslem, kde chceme výmaz ukončit. Příklady ukazují vymazání řádků od 150 - 300, všechny řádky do řádku 100, všechny řádky od řádku 400 a řádek 150.

LIST

starting line number - číslo počátečního řádku
 ending line number - číslo řádku, kde chceme ukončit

Příkaz LIST vypíše na displej program mezi počátečním řádkem a řádkem, kde chceme ukončit. Vypisuje se z části paměti určené pro BASICký text. Přerušení výpisu je možné provést stisknutím tlačítka mezerníku. Opětovným stisknutím je možno ve výpisu pokračovat. Výpis můžeme ukončit stisknutím tlačítek SHIFT a BREAK.

Výpis programu může být proveden i na tiskárnu a to příkazem LIST/P. V příkladech je znázorněn výpis celého programu, všech řádek až po řádek 30, všech řádků programu od řádku 30, všechny řádky programu mezi řádky 30 a 50, a řádku 30.

SEARCH

text data - textová data

Příkaz SEARCH vyhledává v programu BASICkém řádky, které obsahují textový řetězec uvedený v příkazu a zobrazí všechny nalezené řádky na displej. Pokud je v textovém řetězci použit znak uvozovek ", použijte CHR\$(34). Pokud budeme chtít nalezené řádky vytisknout lomíme příkaz písmenem P. Výpis zastavíme stisknutím tlačítek SHIFT a BREAK. Výpis na displeji přerušíme stisknutím mezerníku a opětovným stisknutím ve výpisu pokračujeme. V příkladech hledáme řádky obsahující textový řetězec "ABC" a řádek obsahuje instrukci PRINT "A".

RENUM

new line number - nové číslo řádku
old line number - staré číslo řádku
increment - přírůstek

Příkaz RENUM přečísluje řádky programu v BASICU. Při provádění tohoto příkazu jsou přečíslovány řádky i jsou změněna čísla v instrukcích GOTO, GOSUB, ON GOTO a ON GOSUB. V příkladech jsou přečíslovány řádky od řádku 10 s přírůstkem 10, od řádku 100 s přírůstkem 10, změni řádek 50 na řádek 100 a od něj přečísluje řádky s přírůstkem 20. Poslední příklad znázorňuje i přečíslování instrukce GOTO.

Poznámka : Pokud jsou zadány nové i staré číslo řádku musí být nové číslo řádku větší než staré. Maximální velikost čísla řádku může být 65535. Pokud budeme chtít generovat řádek s vyšším číslem dojde k chybě.

NEW

Příkaz New vymaže z paměti celý program v BASICU a obsahy pamětí polí a proměnných. Pokud je část paměti pro BASIC omezena instrukcí LIMIT vymaže se pouze omezená část. Jako příklad si zadejte do počítače krátký program a po jeho zadání použijte instrukci NEW.

NEW DN

Tento příkaz se používá k rozšíření uživatelské paměti tím, že se vymaže s interpretru BASIC informace o ovládání přídatné tiskárny MZ 1P16. Pokud tuto tiskárnu nepoužíváme můžeme si takto rozšířit uživatelskou paměť. Použitím příkazu NEW DN však vymažeme uživatelskou paměť.

CLR

Příkaz CLR vynuluje všechny proměnné a zruší definici polí. Všechny číselné proměnné položí rovno nule, všechny textové řetězce rovny (" "), tedy nulovém řetězci. Pokud budeme chtít po tomto příkazu použít pole znaku musíme na ně znovu použít instrukci DIM. Příkaz CLR také zruší nadefinované funkce instrukcí DEF FN. Příkaz CLR nesmí být používán v podprogramech a mezi instrukcemi FOR a NEXT. Vpříkladě na řádku 40 příkaz CLR vynuluje proměnnou A a B\$.

CONT

Příkaz CONT slouží ke znovuodstartování programu, který byl odstartován a jeho provádění bylo přerušeno pomocí tlačítek SHIFT a BREAK a nebo instrukcí STOP. V následujícím přehledu uvádíme, kdy je možno příkaz CONT použít.

- Pokračování je možné . Program byl zastaven použitím tlačítek SHIFT a BREAK.
 . Program byl zastaven pomocí instrukce STOP

Pokračování možné není

- . Před použitím příkazu RUN
- . "READY" je zobrazeno na displeji po chybě v běhu programu
- . Operace s přehrávačem byla přerušena použitím tlačítek SHIFT a BREAK
- . Program byl zastaven během provádění instrukce MUSIC
- . Po ukončení programu instrukcí END

See also - znamená podívejte se také na příkaz nebo instrukci související s danou instrukcí, zde je míněna instrukce STOP

RUN

starting line number - číslo počátečního řádku

Příkaz RUN se používá k provádění programu zapsaného v BASICké textové paměti. Pokud je program odstartován bez zadání čísla počátečního řádku tak je program prováděn od prvního řádku programu a zároveň jsou v tomto případě vynulovány všechny proměnné a pole před provedením programu. V příkladech vidíme odstartování od začátku a od řádku 200.

6.2 ZÁKLADNÍ INSTRUKCE

CLS

Instrukce CLS vyčistí obrazovku, respektive obrazovku určenou příkazem CONSOLE.

CONSOLE

starting line - počáteční řádek

number of lines - počet řádků

Příkaz CONSOLE určuje část obrazovky, která je určena pro čištění instrukcí CLS nebo stisknutím tlačítek SHIFT a BREAK. Tento příkaz je neproveditelný po zadání příkazu PLOT ON. Zadejte počet řádků pokud budete chtít na obrazovce měnit prvky a zadejte počet řádků dostatečně veliký, aby změny nebyly obtížné. V příkladech vidíme použití příkazu CONSOLE pro celou obrazovku a pro její část.

CURSOR

X coordinate - Xová souřadnice

Y coordinate - Yová souřadnice

Instrukce CURSOR slouží k nastavení kursoru na dané místo obrazovky. Obě zadané souřadnice musí být celá čísla a to pro Y souřadnici v rozsahu od 0 do 24 a u X souřadnice musí být podle nastavení rozměru obrazovky mezi 0 a 39 a nebo 0 a 79. Instrukce CURSOR se používá velmi často ve spojení s instrukcemi PRINT a INPUT. Například vidíme posun kursoru do místa B,10.

REM

Tato instrukce se během provádění programu neprovádí. Používá se k označování částí programu k jejich snadnějšímu pozdějšímu čtení. Programátor má po čase možnost rychleji se v programu orientovat.

LET

variable - proměnná

expression - výraz

Instrukce LET přiřazuje proměnné (textové nebo číselné) hodnotu určenou výrazem na druhé straně přiřazovacího znaménka. Jak je ukázáno na příkladech může být instrukce LET vypuštěna. V příkladech vidíme jak může být vypuštěna instrukce LET a jsou zde porovnány dva programy, potom je přiřazena proměnné N hodnota 32, Proměnné A je hodnota zvýšena o 5, a proměnné B\$ hodnota "SUNDAY". Hodnota 1 je přiřazena proměnné A bez instrukce LET. Poslední dva příklady ukazují nesprávné použití přiřazovací instrukce. V prvním z nich je textové proměnné přiřazovaný číselný výraz a ve druhém je místo proměnné za instrukcí LET je použita funkce logaritmu tj. nikoliv proměnná.

STOP

Pomocí instrukce STOP můžeme zastavit provádění programu a ručně dodat počítači informaci a nebo si můžete informace od počítače vyžádat. Pokračování je možné zadáním příkazu CONT jak již bylo vysvětleno. Po zastavení programu je na displeji zobrazeno číslo řádku, kde byl program zastaven. Na příkladě je uvedeno jak se program zastaví, když je možno pokračovat příkazem CONT. Žádné otevřené rejstříky nemohou být uzavřeny instrukcí STOP.

END

Instrukce END se používá pro zakončení programu. Po ukončení programu interpret BASIC umožní přímý přístup k počítači. Tato skutečnost je počítačem sdělena zobrazením READY na displeji. Provedením instrukce END se uzavřou všechny otevřené rejstříky. Po provedení instrukce END není možné použít příkaz CONT.

FOR - NEXT

control variable	- proměnná cyklu
initial value	- počáteční hodnota
final value	- konečná hodnota
increment	- přírůstek

Instrukce FOR - NEXT opakují instrukce mezi instrukcí FOR a instrukcí NEXT tak dlouho, jak je dáno zadáním instrukce. Lépe je to ukázáno na příkladě na str. 6-14 uprostřed.

- (1) Při běhu programu se nejprve přiřadí proměnné N počáteční hodnota tj. 0.
- (2) Řádky 30 až 50 se provedou a a hodnoty proměnných A a N se zobrazí na displeji.
- (3) Na řádce 60 se hodnota N zvětší o dvě a je porovnaná s konečnou hodnotou tedy 10. Pokud je menší než 10 počítá se vrátí na řádek 20 a celý postup se opakuje.

Po ukončení provádění se pokračuje v programu za řádkem 60.

Program tedy 6x skočil ze řádku 60 na řádek 20.

Pokud je v instrukci vynechán přírůstek je automaticky považován za roven 1. V našem příkladě bv se provedlo tedy 11 skoků, pokud by byl přírůstek vynechán.

Instrukcí FOR - NEXT, které v češtině nazýváme také instrukcí cyklu, může být v sobě vřazeno více, ale je nutné dávat pozor na to abychom neskákali z vnitřního cyklu do vnějšího. Pokud je na stejném místě ukončeno více cyklů je možno pro ukončení použít jedinou instrukci NEXT, je však nutné u ní uvést správné parametry. Nejlépe nám tyto skutečnosti dokumentuje příklad nastr. 6-15 nahoře. Inner loop - vnitřní skok, Outer loop - vnější skok

Na str. 6-15 uprostřed jsou uvedeny dva příklady špatného použití instrukce cyklu a to v prvním případě je dvakrát použita stejná proměnná cyklu a ve druhém se kříží cykly tj. není ukončen vnitřní cyklus před ukončením vnějšího.

Poznámka : V syntaxu jazyka BASIC není uložena informace kolik cyklů můžeme mít uloženo v sobě. Jsme tedy pouze omezeni velikostí paměti. Instrukce CLR a LIMIT nemohou být použity uvnitř cyklu.

LABEL

label name - jméno návěstí

Instrukce LABEL umožňuje označit řádek návěstím. Potom je možno

provádět skoky a podprogramy nejen na číslo řádku, ale i na návěští. Na příkladu vidíme označení řádku 100 návěstím "ABC" a podprogram adresovaný z řádku 20 bude tedy prováděn od řádku ABC (řádek 100).

GOTO

line number - číslo řádku
label - návěští

Instrukce GOTO nám umožňuje změnit chod programu tak, že nám umožňuje předepsat jiné vykonávání programu než řádek po řádku podle rostoucích čísel řádků. Po instrukci GOTO může následovat buď číslo řádku a nebo návěští, která určují na který řádek se přejde a bude se pokračovat v provádění programu. Pokud se řádek neprovádí je na něm instrukce REM nebo DATA začne se provádět řádek s číslem nejbližší vyšším. V příkladu vidíme převedení programu zpět na řádek 20. Instrukce GOTO nesmí použít přechod na řádek, který je mezi instrukcemi FOR a NEXT. Na formátu instrukce GOTO vidíme význam složených závorek. Je zde tedy možno použít pro určení řádku buď číslem a nebo návěstím. Tuto skutečnost tedy vystihují závorky složené.

ON GOTO

numeric expresion - číselný výraz
line number - číslo řádku
label - návěští

Instrukce ON - GOTO je instrukce, která je obdobou instrukce GOTO. Umožňuje tedy přechod na místo programu, kde se bude v provádění programu pokračovat, podle potřeb programátora. Tato instrukce umožňuje však volit podle hodnoty číselného výrazu místo programu, na které chceme přejít. Na příkladu

```
100 ON A GOTO 200,300,400,500
```

jsme číselný výraz nepoužili, pouze jsme zadali místo něj proměnnou A, která zde může nabývat čtyřech hodnot 1,2,3 a nebo 4. Podle hodnoty A budeme přecházet na řádky 200,300,400 a nebo 500

V případě zadání desetinného čísla jako číselného výrazu (v našich příkladech A) je před provedením instrukce desetinná část ignorována.

Poznámka : Pokud je hodnota číselného výrazu větší než počet čísel řádků v seznamu program probíhá dál od dalšího řádku programu. To samé se stane pokud je číselný výraz záporný a nebo je menší než 1.

GOSUB - RETURN

line number - číslo řádku

label - návěští

Instrukce GOSUB umožňuje převést hlavní program do podprogramu a to tak, že začátek podprogramu začíná buď specifikovaným číslem řádku a nebo návěstím. Ukončení podprogramu se provádí instrukcí RETURN. Po ukončení podprogramu přejde program zpět do hlavního programu na další instrukci nebo příkaz.

Podprogram je část programu, která v jednom programu může být použita vícekrát. Z podprogramu je možno přejít na další podprogram a tak dále. Omezení v počtu použitých podprogramů je pouze v paměti počítače. Všechny podprogramy musí být ukončeny instrukcí RETURN. V příkladě je řádek 10-60 hlavní program a řádek 100 až 110 podprogram.

ON GOSUB

numeric expression - číselný výraz

line number - číslo řádky

label - návěští

ON GOSUB je instrukce, která umožňuje vícenásobné použití podprogramů stejně jako v případě ON GOTO vícenásobné použití skoků. Příklad použití je znázorněn na programu pro rozvrh času. Je zde zajímavé vícenásobné použití podprogramu. Doporučujeme Vám podrobné prostudování programu. Zkratky MON,TUE,WED, THU,FRI,SAT jsou anglické zkratky názvů dnů v týdnu.

IF - THEN - ELSE

relational expression	- relační výraz
logical expression	- logický výraz
statement	- instrukce
line number	- číslo řádky
label	- návěští

Instrukce IF-THEN-ELSE umožňuje rozvětvit program podle výsledku logického nebo relačního výrazu. Pokud je výraz pravdivý provede se instrukce za instrukcí THEN. Pokud za instrukcí THEN je číslo řádku nebo návěští pokračuje program od takto určeného řádku. Pokud je logický nebo relační výraz nepravdivý je provedena instrukce za instrukcí ELSE. Pokud je za instrukcí ELSE číslo řádku nebo návěští je program převeden na tento řádek. Pokud není v instrukci IF - THEN - ELSE instrukce ELSE uvedena program pokračuje na dalším řádku.

Vpříkladech jsou znázorněna různá použití. V prvním případě pokud je C menší než 1 je C přiřazena hodnota 3 a pokud je větší nebo rovno 1 je C přiřazena hodnota C-1.

V druhém případě pokud C nerovná se 0 je proveden skok na řádek 150 a jinak je program ukončen.

Ve třetím příkladu pokud je AŽ rovno "ABC" přiřazen řetězec "ABCDEF" a pokud ne tak se v programu pokračuje na dalším řádku.

Poznámka : Je nutno při použití relačních a logických výrazů mít na paměti, že porovnávání se děje tak, že čísla mohou mít pouze takové hodnoty, které dovoluje rozsah počítače a proto je nutno tato porovnávání provádět pozorně zvláště v oblasti hodně malých čísel.

Dále je nůtno mít na zřeteli reprezentaci čísel v počítači, kdy numerický výraz nemusí být vždy zcela totožný s přímo zadaným číslem i když se matematicky sobě rovnají. Viz příklad str. 6-21

IF - GOTO

relation expression	- relační výraz
logical expression	- logický výraz
line number	- číslo řádku
label	- návěstí

Instrukce IF - GOTO umožňuje provést instrukci skoku podle výsledku relačního nebo logického výrazu. Touto instrukcí můžeme tedy také rozvětvit program podle našich potřeb. Obdobně jako u předchozí instrukce je skok proveden pokud je logický nebo relační výraz pravdivý. Pokud pravdivý není pokračuje se na dalším řádku programu bez ohledu na příkazy a instrukce zapsané na stejném řádku jako IF - GOTO za touto instrukcí. Vpříkladě je uveden příklad skoku podmíněný tím, že vkládaná hodnota X musí být 999, aby se příkaz skoku provedl.

IF - GOSUB

relation expression	- relační výraz
logical expression	- logický výraz
line number	- číslo řádku
label	- návěstí

Instrukce IF - GOSUB umožňuje provádět podprogram určený číslem řádku a nebo návěstím na základě výsledku relačního nebo logického výrazu. Po provedení podprogramu (pokud je prováděn) se chod programu vrátí na první řádek, který následuje za řádkem s instrukcí podmíněného podprogramu. Podprogram je prováděn pouze tehdy, jestli relační výraz pravdivý a nebo pokud je logický výraz splněn. Pokud podprogram prováděn není (podmínky nejsou splněny) pokračuje se v programu dále. Použití je demonstrováno na příkladě. Všimněte si, že u této instrukce jsou prováděny při nesplnění podmínky instrukce následující instrukci IF - GOSUB na téže řádce.

PRINT `

palette code - kód barev
data - zobrazená data

Zkrácená forma zápisu instrukce PRINT je otazník. Slovně uvádíme tuto formu pouze v tomto případě.

Instrukce PRINT zobrazuje data na obrazovce v barvě jakou předepíšeme pomocí kódu barev. Pokud barvu pomocí kódu neuvedeme je barva textu řízena přijatou barvou předchozími instrukcemi. Pokud je pro rozdělení dvou různých dat použit středník jsou tato data zobrazena bez mezery, pokud je použita čárka je mezi daty provedena desetimezerová tabulace. Pokud nejsou zadána žádná data je vynechán jeden řádek.

Číselná data jsou zobrazována v rozsahu od 1×10^{-8} do 1×10^8 normálně s pohyblivou desetinnou tečkou a mimo tento rozsah jsou zobrazována v exponenciálním vyjádření.

První příklad ukazuje zobrazení textových dat ABC a číselných 123 bez mezery a v barvě korespondující s kódem 2. Ve druhém příkladě zobrazen stejný údaj, ale s použitím tabulátoru a barvy podle kódu 3.

Poznámka :

Instrukci PRINT můžeme také použít k ovládní obrazovky podle znaku, který zařadíme za PRINT do úvozovek. Znaky musí být zapsány v grafickém režimu tj. po stisknutí tlačítka GRAPH.

Po zadání znaku musíme stisknutím tlačítka ALPHA přejít zpět na normální režim. Význam použití tlačítek je následující

CLR	- vymaže celou obrazovku a umístí kurzor do základní polohy (levý horní roh)
HOME	- umístí kurzor do základní polohy
Šipka kurzoru vlevo	- posune kurzor o jednu polohu doleva
Šipka kurzoru vpravo	- posune kurzor o jednu polohu doprava
Šipka kurzoru nahoru	- posune kurzor o jeden řádek nahoru
Šipka kurzoru dolů	- posune kurzor o jeden řádek dolů.

V programu jsou tyto znaky zobrazeny za instrukcí PRINT světle na tmavém pozadí. CLR je nahrazeno písmenem C a HOME písmenem H.

PRINT USING

palette code	- kód barev
format string	- formátový řetězec
data	- zobrazená data

Instrukce PRINT USING zobrazuje data na obrazovku v zadaném formátu. Tato instrukce je zadávána ve stejném formátu jako instrukce PRINT s tím rozdílem, že zadáváme formát zobrazení formátovým řetězcem s použitím desetinné tečky. Nejlépe použití instrukce PRINT USING vyložíme na příkladech. Zde je použito příkladů více, proto použijeme stejného členění jako originál. aby jste snáze našli příslušný příklad. Neuvádíme jednotlivé znaky, protože jsou obtížné pro tisk.

(1) Formátové znaky pro číselné hodnoty

- (a) Symbol " hudebního" křížku je určen pro určení jedné číselné pozice. Pokud je zobrazovaných číslic méně než je určeno, je číslo umístěno co nejvíce vpravo a zbývající místa vlevo jsou nahrazeny mezerou.
- (b) Desetinná tečka určuje, kde bude desetinná tečka zobrazena a kolik míst bude zobrazeno před tečkou a kolik za tečkou. Pokud je před tečkou určeno více míst než příslušné číslo obsahuje jsou chybějící číslice nahrazeny mezerou a chybějící čísla za desetinnou tečkou nahrazena nulami. I zde pro určení počtu míst používáme "hudební" křížky. V příkladě vidíme určení formátu tak, že jsou zobrazována 3 místa před tečkou a dvě za tečkou a jak je zobrazeno celé číslo o dvou číslicích.
- (c) Symbol čárky zde má jiný význam než na jaký jsme zvyklí. Jak jste si již všimli používá se u počítače místo desetinné čárky desetinná tečka. Čárka se používá pro oddělení trojic číslic u delších čísel, aby byla snadněji čitelná. Tento způsob není u nás obvyklý a může vést k záměnám s desetinnou tečkou a proto jej používejte opatrně.

- (d) Znaménko používáme normálně tak, že záporné znaménko zobrazíme před číslem a kladné znaménko nezobrazujeme. Je možné však zobrazit obě znaménka před číslem tj. pokud je zobrazované číslo kladné je před ním zobrazeno znaménko +. Dále je možno umístit znaménko za číslo. Máme dvě možnosti. Můžeme před nebo za "hudební" křížky určující počet číslic umístit znaménko + a nebo znaménko -. Při použití znaménka + je znaménko vždy zobrazeno (i když je kladné) a při použití znaménka - se zobrazí pouze znaménko záporné a znaménko kladné je nahrazeno mezerou.
- (e) Umístění dvou hvězdiček před "hudební" křížky určující počet míst má za následek zobrazení těchto dvou hvězdiček před číslem a za nimi zobrazení dalších hvězdiček na pozice nulové před číslem. V příkladě jsme určili číslu 4 místa a před určení jsme umístili dvě hvězdičky. Při zobrazení tří místného čísla se zobrazí dvě hvězdičky a jedna hvězdička na nulovou první pozici.
- (f) Umístění dvou znaků pro libry před určení počet míst zobrazovaného čísla má za následek zobrazení znaku libry bezprostředně před první číslicí zleva.
- (g) Viz (f) pouze místo znaku libry je použit znak dolaru.
- (h) Pokud jsou za "hudebními" křížky umístěny čtyři exponenciální operátory určujeme formát exponenciálního vyjádření. První místo je v tomto případě určeno pro znaménko.
- (i) Rozšíření seznamu operandů.
Pokud je za instrukcí PRINT USING zapsáno více proměnných oddělených středníky nebo čárkami budou zobrazeny tyto proměnné ve formátu, který je určen formátovým řetězcem.

(2) Formátové znaky pro řetězce

(a) Pokud je ve formátovém řetězci použit vykřičník je zobrazen pouze první znak z řetězce

(b) Zvláštní znaky a mezi nimi umístěných n mezer zobrazí $n + 2$ znaky řetězce. Pokud je řetězec delší je zobrazeno $n + 2$ znaky zleva. Pokud je řetězec kratší je umístěn vlevo a zbývající místa jsou doplněna mezerami.

(3) Použití řetězce spolu s určením délky proměnné umožňuje zobrazit řetězec dohromady s proměnnou určenou jejím názvem, který je umístěn zastředníkem. Nejlépe je tato vlastnost instrukce PRINT USING vidět na příkladu.

(4) Oddělené použití části instrukce USING

Obvykle používáme slova PRINT a USING dohromady. Je však možné za instrukcí PRINT napsat seznam zobrazovaných proměnných a po něm teprve použít USING a za formátovým řetězcem zapsat další seznam proměnných a ty se budou již zobrazovat podle určeného formátu. Opět je tato vlastnost instrukce PRINT USING vidět na příkladu.

INPUT

message - zpráva
variable - proměnná

Instrukce INPUT čte data vkládaná během běhu programu a ukládá je do paměti pod názvy proměnných.

Pokud je v programu instrukce INPUT program se na ní zastaví a zobrazí na displeji zprávu obsaženou v zadání instrukce INPUT a pokud není zpráva zadána zobrazí se otazník. Počítač čeká na zadání dat a po jejich zadání je nutno stisknout tlačítko CR.

Data jsou uložena pod určené jméno proměnné do paměti a program pokračuje dál ve svém běhu. Typ zadávaných dat musí být stejný s určenou proměnnou. Znakové konstanty mohou být zadávány bez úvozovek. Pouze pokud má znaková konstanta začínat nebo končit mezerou a nebo čárkou je nutné znakovou konstantu zadat s úvozovkami.

Na řádku 10 příkladu vidíme zadávání dvou proměnných jednou instrukcí INPUT. První proměnné A je přiřazen první datový údaj a druhé B\$ druhý. Na řádku 20 je vidět použití zprávy v instrukci.

GET

variable - proměnná

Instrukce GET sleduje není-li stisknuta některá klávesa na klávesnici a pokud tomu tak je, je hodnota této klávesy přiřazena určené proměnné. Proměnné je určena nulová hodnota pokud není klávesa stisknuta. Číselné proměnné zde nabývají hodnot jednoduchých celých čísel (0 - 9). U textových proměnných můžeme zadat jednoduchý znak. Pokud je určena číselná proměnná a je stisknuta " nečíselná " klávesa není přiřazena proměnné žádná hodnota.

V příkladě na řádce 10 vidíme zadání instrukce GET. Všimněte si instrukce podmíněného skoku zpět na stejný řádek pro případ nulové hodnoty A\$. Tento program na displeji zobrazí znak jehož klávesu stiskneme pokud je zobrazitelný.

DIM

variable - proměnná
subscript - popis rozměru

Instrukce DIM deklaruje pole jedno až čtyřrozměrné a rezervuje pro tato pole místo v paměti. Název pole může být složen ze dvou znaků s tím, že první musí být písmeno a popis rozměru pole nám určuje velikost pole v příslušném rozměru. Jeho hodnota

je dána velikostí volné paměti v počítači. Pro každé deklarované pole musí být použit jiný název. Provedení instrukce DIM vynuluje celé pole tj. Přidělí číselnému poli (všem jeho prvkům) hodnotu 0 a textovému poli hodnotu prázdného znaku. Z tohoto důvodu je nutné, aby instrukce DIM byla provedena před obsazováním pole konkrétními hodnotami.

Jestliže je deklarováno pole, které již byla v programu deklarováno a nebo jestliže nově deklarovaný rozměr je větší než stávající pole nastane chyba.

V příkladech vidíme deklaraci pole jednorozměrného se 100 prvky a dvojrozměrného s 10 x 10 prvky.

READ - DATA

variable - proměnná
constant - konstanta

Podobně jako instrukce INPUT a GET i instrukce READ slouží k zadání dat do počítače. Na rozdíl od obu dříve již popsaných instrukcí vstupu tato instrukce nepoužívá pro zadávání hodnot proměnných klávesnici. Hodnoty jednotlivým proměnným jsou přiřazeny podle seznamu hodnot, který je uveden za instrukcí DATA. Jména proměnných za instrukcí READ musí jedno po druhém korespondovat s hodnotami uvedenými za instrukcí DATA. Pokud jsou jako data uvedeny řetězce, nemusí být v uvozovkách, pouze v případě nulového řetězce a řetězce obsahujícího mezery je nutné řetězec do úvozovek uzavřít.

Na prvním příkladě jsou hodnoty za instrukcí DATA přiřazeny instrukcí READ proměnným A,B,C a D a zobrazeny na displeji. V druhém příkladě vidíme střídání typu proměnných a také zapsání datového řetězce bez uvozovek. Je nutné pouze dodržet správný typ proměnné v souvislosti s typem zadávaných dat.

Ve třetím příkladě je ukázáno přiřazování hodnot prvkům pole a jejich zobrazení na displeji.

Ve čtvrtém příkladě jsou hned dvě chyby. Za prvé na řádku 10 je instrukce READ a za ní číselná proměnná A, ale instrukce DATA má za sebou řetězec X. Za druhé jsou za sebou dvě instrukce READ bez instrukce DATA.

RESTORE

line number - číslo řádku
label - návěstí

Instrukce RESTORE je určena k určení, kterou instrukci DATA budeme přiřazovat k instrukci READ. Instrukci RESTORE používáme pokud je stejná instrukce DATA použita vícekrát v jednom programu. Pokud není za instrukcí RESTORE uvedeno číslo řádku nebo je nulové je instrukci READ přiřazena instrukce DATA na řádku s nejnižším číslem. Jinak instrukcí RESTORE určíme z jakého řádku (zadáme jeho číslo a nebo návěstí) vezmeme instrukci DATA s daty a přiřadíme ji instrukci READ. Nejlépe je použití instrukce RESTORE vidět na příkladě.

DEF FN

function name - jméno funkce
variable - proměnná
numeric expression - číselný výraz

Instrukce DEF FN nám umožňuje si nadefinovat vlastní funkci. Funkce se musí skládat s funkcí, které jsou obsaženy v interpretu jazyka BASIC. Jméno funkce musí být velká písmena. Na prvních dvou příkladech vidíme definování funkcí FNA(X) a FNE(V). Na třetím příkladě je znázorněna chyba. Jedním použitím instrukce DEF FN lze definovat pouze jednu funkci. Ve čtvrtém příkladě je použito při definování funkce funkce již námi definované což je také nepřipustné. Pátý příklad vypočítává kinatickou energii hmoty 5,5kg s počátečním zrychlením 3.5 m/s^2 , zrychlením dvojnásobným a trojnásobným.

Poznámka :

Všechny námi definované funkce jsou zrušeny příkazy CLR a NEW.

TRON

Příkaz TRON zobrazuje jak je program prováděn tím, že jsou na obrazovce (pokud je lomeno P na tiskárně) zobrazena čísla řádků v hranatých závorkách v pořadí jak jsou řádky interpretrem BASIC prováděny. Zadejte počítači příklad a před zadáním příkazu RUN zadejte příkaz TRON. Pod programem je znázorněn tisk jak bude prováděn.

TROFF

Příkaz TROFF ruší funkci zobrazování chodu programu zadanou příkazem TRON.

DEF KEY

key number - číslo klávesy
character string- znakový řetězec

Počítač je vybaven pěti funkčními tlačítky která jsou umístěna v levém horním rohu počítače a jsou označeny F1 až F5. Tato tlačítka se používají tak, že si znakovým řetězcem určíme dopředu co bude každé tlačítko za řetězec zobrazovat. Funkční klávesy můžeme používat normálně mají čísla 1 až 5 a po stisknutí tlačítka SHIFT mají čísla 6 až 10. Znakový řetězec, který tlačítku přiřazujeme může mít maximálně 15 znaků. Pokud zadáme příslušnému funkčnímu tlačítku nový znakový řetězec je starý zapomenut. K zadávání znakového řetězce funkčním tlačítkům používáme funkci DEF KEY, kde zadáme číslo tlačítka (klávesy) a znakový řetězec.

Na příkladech můžeme vidět praktické přiřazení slova SHARP klávese F1 a příkazu RUN ve formě, která umožňuje automaticke odstartování programu. Pokud zadáme počítači CHR\$(13) má to stejný význam jako stisknutí klávesy CR.

KEY LIST

Příkaz KEY LIST umožňuje vypsát na displej seznam znakových retezců, které jsou přiřazeny jednotlivým funkčním tlačítkům. V příklade jsou vypsány znakové řetězce přiřazené funkčním tlačítkům po nahrání interpretru BASIC.

INIT

number of bytes	- počet bytů
monitoring code	- monitorovací kód
initialization code	- inicializační kód
end code	- koncový kód

Příkaz INIT definuje inicializační nastavení a režimy pro extér-ní zařízení.

Formát 1

V tomto formátu příkaz INIT inicializuje přídatnou desku RAM určenou pro ukládání souborů (MZ - 1R18) a vymezuje zde místo pro paměť specifikovanou počtem bytů, zbývajícím místu v paměti rezervuje pro buffer tiskárny. Zadávaný počet bytů musí být v rozsahu od \$0010 do \$FFFF. Když je zadávaný počet bytů vynechán předpokládá se běžné nastavení pro RAM paměť určenou pro zadávání souborů. Během provádění příkazu se objeví zpráva "OK?(Y/N)". Stisknutím Y nastavíte přídatnou RAM paměť jak je ukázáno na příkladě. Stisknutím N se na displeji zobrazí "Break" a vrátíme se zpět do režimu příkazu. Přídatná paměť nebo buffer pro tiskárnu mohou být nedostupné pokud je vyhrazené místo v paměti příliš malé.

Instrukce v příkladě inicializuje přídatnou paměť tak, aby bylo co největší místo v paměti pro ukládání souborů. Na obrázku vpravo je znázorněno rozložení paměti po vykonání instrukce INIT"RAM:\$CFFF".

(Formát 2)

V tomto formátu příkaz INIT určuje tiskárnu a režim, ve kterém bude užit buffer tiskárny.

M indikuje režim bufferu tiskárny

M0: Přímý režim (Buffer je inicializován)

M1: Spool režim (Buffer je inicializován)

M2: Přímý režim (Pokud je spool režim aktivní, tento režim nastane jakmile je cokoli z bufferu vytištěno)

M1 a M2 lze použít pouze v případě, že máme zasunutou desku rozšíření souborové paměti.

Zde upozorňujeme na to, že se k počítači dají připojit starší typy tiskáren Sharp, které se však nedodávají. Proto pasáže týkající se těchto tiskáren vynecháváme.

S určuje typ tiskárny

S0: MZ-1P16 - možno dokoupit

S1: MZ80P5 - starší typ tiskárny Sharp, který se nedodává

S2: Tiskárna, která převádí tištěná data do ASCII kódu.

S3: Průchozí kód

Následující kódy jsou převedeny jak je popsáno dále během provádění instrukce PRINT/P pokud je instrukcí INIT specifikován režim S0 nebo S1.

Na str. 6 - 3B jsou čtyři převody pod sebou.

or - nebo, is converted to - je převedeno na, and - a

CR kód musí být specifikován pokud je další kód jako 00H je použit jako CR kód.

(Formát 3)

Na str. 6-39 až 6-41 je popsáno nastavení vestavěného interfacu RS - 232C. Pokud by jste chtěli tento interface použít spojte se předem s technicko- poradenským střediskem Sharp, Praha 5, Zbo-rovská 35 - při Kancelářských strojích k.ú.o.

(Formát 4)

V tomto formátu příkaz INIT nastavuje displej, určuje rozložení bodů na obrazovce a počet barev jak je znázorněno v tabulce na str. 6-41

Mode - režim, Resolution - rozložení, Characters per line - bodů na řádek, Colours - barvy, Foreground and background colours - barvy popředí a pozadí.

Poznámka :

Pokud jako displej používáme televizor nemůžeme použít režim 3 a 4. K použití režimu 2 a 4 je nutné přidání přidavné grafické paměti MZ 1R25.

V příloze A je více informací o ovládání displeje.

BYE

Příkaz BYE převádí počítač zpět z používání interpretu BASIC do programu Monitor. V kapitole 8 je Monitor popsán.

BOOT

Příkaz BOOT převádí počítač do stavu takového, v jakém byl hned po zapnutí.

WAIT

numeric data - číselná data

Instrukce WAIT pozdrží provádění programu o čas, který je zadán číselným datem. Čas se uvádí v milisekundách tj. například WAIT 1000 pozdrží vykonávání programu o 1 vteřinu.

6.3 INSTRUKCE OVLÁDÁNÍ SOUBORU

DIR

Příkaz DIR zobrazuje jména souborů uložených v přídatné paměti pro ukládání souborů. Pokud je zadán ve formě DIR/P vytiskneme tato jména přímo na tiskárnu. Tento příkaz je použitelný pouze máme-li zasunutou přídatnou desku MZ 1R1B.

Každý název souboru může být z následujících tří typů

BTX : programové soubory v BASICu

BSD : datové a programové soubory v BASICu zapsané sekvenčně v ASCII kódu

OBJ : soubory ve strojovém jazyku

Příklad DIR RAM Zobrazí adresář souborů uložených na přídatné paměti RAM pro ukládání souborů.

RUN

device name - jméno periferie

filename - název souboru

Zruší program, který je uložen v BASICké programové paměti a vymaže programovou pracovní část paměti, nahraje program specifikovaný jménem souboru z periferie specifikované jménem periferie. Potom je nahráný program odstartován od začátku.

Jméno periferie může být vynecháno pokud periferie není připojena a nebo pokud je určena instrukcí DEFAULT. Pokud jsou vynechány všechny parametry není zrušen program v paměti.

Pokud je nahrán soubor sekvenčně v BSD musíme přidat jako parametr písmeno A.

Pokud máme nahrané programy ve strojovém jazyce, můžeme je nahrát a odstartovat zadáním písmene R, hned po zapnutí stroje jako IPL. Není nutné nahrávat jazyk. Typy souborů, které jsou nahrány mohou být BTX, BSD, OBJ.

V příkladě prvním je nahrán soubor typu BTX, který se jmenuje "PROG" z přehrávače a jeho odstartování. Druhý příklad nahrává soubor typu BSD z kazety a odstartuje jej. Soubor je jmenuje "DATA".

LOAD

device name - jméno periferie
 filename - jméno souboru

Příkaz LOAD nahraje určený program do pamětí z vnější paměťové periferie.

Jméno souboru nahrané na vnější paměťové periférii musí být stejné jako jméno souboru, který chceme nahrát. Název periferie musí být buď CMT - přehrávač a nebo RAM - přídatná paměť pro ukládání souborů. Tento parametr může být vynechán pokud je periferie určena pomocí instrukce DEFAULT. Přidáním písmena A nahráváme program, který je nahrán sekvenčně typu BSD. Pomocí tohoto režimu můžeme nahrávat pouze soubory textové v BASICu a nebo soubory ve strojovém jazyce.

Poznámka :

Pokud nahráváme část programu ve strojovém jazyku, která bude využívána v programu v BASICu musíme instrukcí LIMIT dopředu určit pro tuto část programu místo v paměti. Příkaz LOAD může být použit během programu k nahrání souboru ve strojovém kódu.

Při nahrávání programu dodržujeme následující postup

1. Zadáme z klávesnice LOAD"CMT: jméno souboru" CR
2. Na displeji se objeví PLAY a šipka, stiskneme tedy na přehrávači tlačítko PLAY
3. Na displeji se objeví nápis Found " jméno souboru", který nás informuje o tom, že se hledá na kazetě jméno zadaného souboru.
4. Na displeji se zobrazí nápis Loading " jméno souboru", který nás informuje o tom, že se soubor převádí z kazety do vnitřní paměti počítače.
5. Na displeji se objeví nápis Ready, který nás informuje o tom, že je soubor nahrán.

SAVE

device name - jméno periferie
filename - jméno souboru

Jméno periferie může být pouze CMT (přehrávač) a nebo RAM
(přídatná paměť pro ukládání souborů)

Příkaz SAVE přiřadí programu v BASICu jméno souboru a nahraje ho na vnější paměťovou periférii.

Název vnější paměťové periferie může být vynechán pokud je periferie určena pomocí instrukce DEFAULT a nebo pokud periferie chybí. Například není připojena rozšiřovací deska paměti.

Pokud je do příkazu SAVE přidán parametr A, je nahráván program v ASCII kódu jako program typu BSD. Příkaz SAVE tedy umožňuje nahrávat programy typu BTX a BSD.

V příkladech je uveden příkaz SAVE pro nahrání na kazetu programu PROG typu BTX a programu DEMO typu BSD.

Poznámka:

Příkaz SAVE nahrává pouze programy v BASICu. Po nahrání programu si запиšte stav počítadla na přehrávači před a po nahrání, aby jste program mohli snadno vyhledat a aby jste věděli kam můžete nahrát další program.

Postup při nahrávání je následující

1. Zapišeme přes klávesnici SAVE "CMT: jméno souboru" CR
2. Na displeji se objeví šipka a nápis RECORD.PLAY, který nás upozorňuje na to, abychom stiskli tlačítko RECORD na přehrávači.
3. Na displeji se objeví nápis Writting " jméno souboru ", který nás upozorňuje na to, že z vnitřní paměti počítače je nahráván na kazetu program s přiřazeným jménem souboru.
4. Na displeji se objeví název READY, který nás upozorňuje na to, že je nahrávání ukončeno.

VERIFY

filename - jméno souboru

Příkaz VERIFY porovnává zda-li je program nahráný na kasetě stejný s programem ve vnitřní paměti počítače. Pokud tomu tak je, tak je na displeji zobrazeno READY, pokud ne na displeji je zobrazeno "CMT:Illegal data error". Pokud není nahrán program na kazetu dobře je nutno nahrávání zopakovat. Nelze použít příkaz VERIFY na programy nahrané v ASCII a je použitelný pouze pro přehrávač. V příkladě je porovnán program "NAME" s programem ve vnitřní paměti.

Postup při porovnávání je následující

1. Zadejte pomocí klávesnice VERIFY " jméno souboru"
2. Na displeji se objeví šipka a nápis PLAY, který nás upozorňuje na to, abychom stiskli klávesu PLAY na přehrávači.
3. Na displeji se zobrazí Found " jméno souboru", který nás upozorňuje na to, že je na kasetě vyhledáván soubor se stejným jménem.
4. Na displeji se objeví Verifying "jméno souboru" a tím jsme upozorněni na to, že se porovnává soubor z kazety se souborem ve vnitřní paměti.
5. Na displeji se objeví nápis : A/ Ready - oba soubory jsou totožné
nebo
B/ CMT:Illegal data error
Ready
- soubory nejsou stejné je nutno soubor nahrát znovu na kazetu.

DELETE

filename - jméno souboru

Příkaz DELETE vymaže soubor specifikovaný jménem souboru. Tento příkaz je použitelný pouze pro přidavnou paměť MZ 1R18 RAM.

RENAME

old filename - staré jméno souboru
new filename - nové jméno souboru

Příkaz RENAME přejmenuje daný soubor specifikovaný jménem souboru na jméno, které je zadané jako nové. Nové jméno souboru nesmí být totožné se jménem souboru, který je na přídavné RAM paměti již nahrán jinak je signalizována chyba. RAM může být vynecháno pokud je specifikováno instrukcí DEFAULT. Příkaz RENAME je možno použít pouze na přídavnou paměť MZ 1R18.

V příkladě je uvedeno přejmenování souboru OLOPROG. na NEWPROG.

CHAIN

device name - jméno periferie
filename - jméno souboru

Jméno periferie musí být pouze CMT nebo RAM.

Instrukce CHAIN převádí provádění programu na jiný program souboru. CHAIN instrukce může otevřít soubor. Provádění instrukce CHAIN má stejný efekt jako provedení příkazu RUN s tím rozdílem, že CHAIN ponechává proměnné a pole tak, jak byly. Jméno periferie může být vynecháno, pokud je určeno instrukcí DEFAULT.

V příkladě vidíme krátký program, kde je na řádku 30 provedena instrukce CHAIN, tedy je nahrán z kazety program PROG a odstartován s tím, že z předchozího programu jsou ponechány hodnoty proměnných A a B.

MERGE

device name - jméno periferie
filename - jméno souboru
Jméno periferie musí být CMT nebo RAM

Příkaz MERGE sloučí program určený jménem programu s programem

uloženým ve vnitřní paměti. Jménem programu se rozumí jméno souboru. Jméno periferie může být vynecháno pokud je určeno instrukcí DEFAULT.

Řádky programu ve vnitřní paměti, které mají stejná čísla jako řádky programu nahrávaného, jsou přepsány řádky programu nahrávaného. Pokud budeme chtít sloučit Příkazem MERGE programy typu BSD použijeme v příkazu parametr A.

V příkladu vidíme sloučení dvou programů pomocí příkazu MERGE"CMT:PROG" a výsledný program, který si můžeme prohlédnout pomocí příkazu LIST.

WOPEN ~~#~~

logical number - logické číslo

device name - jméno periferie

filename - jméno souboru

logické číslo musí být celé číslo od 1 do 127

jméno periferie může být CMT, RAM nebo RSn.

Tato instrukce otevírá BSD soubor pro výstup. Doplní jej logickým číslem a jménem souboru.

Jméno periferie může být vynecháno pokud je určeno instrukcí DEFAULT. Určení periferie RSn určuje výstup přes interface RS 232C

V příkladě na řádku 10 je vidět otevření souboru "DATA" a přiřazení logického čísla 1 s tím, že bude nahrán na kazetu, v druhém případě vyslán přes RS 232C. Celý program ukazuje zapsání čísel 1 až 99 do určeného souboru.

PRINT ~~#~~

logical number - logické číslo

data - data

Tato instrukce запиše data do souboru, který je otevřen instrukcí WOPEN ~~#~~ a to sekvenčně.

logické číslo musí být stejné jako číslo, kterým byl soubor otevřen. Data mohou být numerická i alfanumerická. V příkladě je vidět zapsání číselných dat 1,2,3 do souboru DATA. Soubor má logické číslo a je otevřen pro výstup.

ROPEN

logical number - logické číslo
 device name - jméno periferie
 filename - jméno souboru
 logické číslo musí být celé číslo od 1 do 127
 jméno periferie musí být CMT, RAM, RSn

Instrukce ROPEN# otevírá soubor pro vstup. Označí soubor logickým číslem z periferie určené jménem a soubor je určen jménem souboru. Pokud je instrukcí DEFAULT určena periferie je možné její jméno vynechat.

Na příkladu můžeme vidět otevření souboru DATA na kazetě a otevření pro čtení z interfacu RS 232C. Dále je vidět přečtení souboru z příkladu pro instrukci WOPEN#.

INPUT#

logical number - logické číslo
 variable - proměnná

Instrukce INPUT# čte data ze souboru otevřeného pro vstup instrukcí ROPEN# a přiřazuje je určeným proměnným. Proměnné mohou být prvky pole. Logické číslo musí být stejné jako při prvním otevření souboru pro vstup. Pokud jsou použity instrukce READ - DATA je nutné dát pozor na dodržení typu proměnných. Konec souboru může testován pomocí funkce EOF# pokud je určený soubor na přídatné RAM paměti.

V příkladu jsou čtena numerická data ze souboru otevřeného logickým číslem 2 a uložena jako proměnné A,B,C.

EOF

logical number - logické číslo

Funkce EOF# se používá k nalezení konce souboru. Tato funkce signalizuje konec souboru, pokud jsou všechna data přečtena. Hodnota -1 (pravda) je nabyta pokud je soubor ukončen. Tato funkce nelze použít s přehrávačem.

Funkce EOF# se používá ve spojení s instrukcí IF za instrukcí INPUT. V příkladu vidíme čtení dat ze souboru DATA a jejich zobrazování na displeji dokud není nalezen konec souboru.

CLOSE

logical number - logické číslo

Instrukce CLOSE uzavírá otevřený soubor pod určeným logickým číslem. Logické číslo přiřazené souboru se jeho uzavřením vyruší. Operace CLOSE na souboru otevřeném pro výstup vyrovná výstupní buffer. Operace CLOSE bez logického čísla uzavře všechny otevřené soubory.

Příklady ukazují uzavření souboru s logickým číslem 1 a uzavření všech otevřených souborů.

KILL

logical number - logické číslo

Příkaz KILL# zastaví zápis dat do otevřeného souboru pod logickým číslem. Pokud není za příkazem KILL# uvedeno logické číslo jsou zastaveny všechny zápisy a zároveň jsou uzavřeny všechny otevřené soubory a zrušena všechna logická čísla.

V příkladu je zastaven zápis do souboru otevřeného pod logickým číslem 3 a toto logické číslo je v souboru zrušeno.

DEFAULT

device name - jméno periferie

Instrukce DEFAULT definuje jména periférií, které budou použity pokud je parametr (device name) vynechán ve vstupních a výstupních instrukcích.

Jména periférií jsou následující

CMT	Přehrávač
RAM	Přídavná paměť pro zápis souborů
LPT	Tiskárna
RS	Interfáce RS 232C

V příkladu je instrukcí OEFAULT určeno, že následující chybějící jména periférií jsou CMT.

6.4 INSTRUKCE PRO OVLÁDÁNÍ GRAFIKY**COLOR**

palette code - kód barev
mode - režim

Instrukce COLOR určuje barevným kódem a režimem užití barev při instrukcích PRINT, PRINT USING a grafickými instrukcemi SET, RESET, LINE, BLINE, BOX, CIRCLE, PAINT, PATTERN a SYMBOL. Režim určuje typ logické operace převedené na barvy. Když je hodnota režimu určena 0 je zabarvená část přebarvena novou barvou. Pokud je hodnota 1 stará i nová barva jsou logicky očervená. Parametr režim nelze použít pro instrukce RESET a BLINE. V příkladu vidíme kreslení s barevným kódem 3 a překrývacím režimem 0.

PAL

palette code - kód palety
 color code - kód barvy

PAL instrukce řadí k sobě (srovnává) kód barvy a kód palety. Obojí kód palety i kód barvy mohou nabýt hodnot od 0 do 15. V barevném režimu jiném než 16-ti barevném, uživatel může nastavit 2 nebo 4 kódy palety najednou a vybrat z 16 barev. V 16-ti barevném režimu může uživatel nastavit blok palety pomocí příkazu INIT a vybrat čtyři palety pro blok opět výběr ze 16 barev. Hodnoty palety a barev jsou znázorněny níže.

(1) 2-barevný režim

kód palety	kód barvy
0	0 černá
1	15 světle bílá

(2) 4-barevný režim

kód palety	kód barvy
0	0 černá
1	1 modrá
2	2 červená
3	15 světle bílá

V této tabulce je ukázána relace mezi paletou a kódy barev jak je nastaveno po nahrání BASICu. Můžete se rozhodnout pro čtyři barvy z 16 možností.

(3) 16-barevný režim

Zbývající hodnoty kódy palety jsou identické s tímto barevným režimem. V tomto režimu jsou barvy pevně dány pro každý paletový blok. n - číslo paletového bloku. Podívejte se na instrukci INIT.

n	kód barvy	barva	n	kód barvy	barva
0	0	černá	2	8	šedá
	1	modrá		9	světle modrá
	2	červená		10	světle červená
	3			11	
1	4	zelená	3	12	světle zelená
	5				
	6	žlutá		14	světle žlutá
	7	bílá		15	světle bílá

Poznámka :

Když je paletový blok měněn pomocí instrukce INIT v 16-ti barevném režimu je nastavení kódu palety inicializováno.

SET

color specification - určení barvy

X - coordinate - X souřadnice

Y - coordinate - Y souřadnice

Instrukce SET určuje bod obrazovky X a Y souřadnicí a barvou. X a Y souřadnice mohou být číselné výrazy (číselné konstanty, proměnné ap.) Jejich hodnoty musí být v rozsahu

- 16384 = X souřadnice = 16383

- 16384 = Y souřadnice = 16383

virtual area - možná plocha

display area - plocha displeje

Přestože můžeme určit X a Y souřadnici po celé možné ploše, Basic umožňuje zobrazení pouze ve vyšrafované ploše. Kód palety určuje barvu bodu, který je kreslen. Režim musí být 0 nebo 1. Pokud je určena 0 bod je zobrazen v barvě určené kódem palety. Pokud je specifikována 1 bod je zobrazen v barvě určené kroužkem v běžné barvě s kódem palety určeným instrukcí SET. Pokud specifikace barvy v instrukci chybí je barva bodu určena posledním

určením instrukcí COLOR.

V příkladě je uvedeno zobrazení bodu o souřadnicích (100,50) v barvě přidružené ke kódu palety 3, překrývací režim 0.

RESET

color specification	- určení barvy
X - coordinate	- souřadnice X
Y - coordinate	- souřadnice Y
palette code	- kód palety
superimpose mode	- překrývací režim

Instrukce RESET mění barvu bodu na displeji určeným X a Y souřadnicí. X a Y souřadnice mají stejný rozsah a stejný význam jako u instrukce SET. Kód palety určuje barvu bodu, která bude zrušena, režim musí být 0 nebo 1.

LINE

color specification	- určení barvy
X - coordinate	- souřadnice X
Y - coordinate	- souřadnice Y
palette code	- kód palety
superimpose mode	- překrývací režim

Instrukce LINE nakreslí čáru mezi určenými body v určené barvě. X a Y souřadnice mají stejný význam a rozsah jako u instrukce SET. Parametr určení barvy má také stejný význam jako u instrukce SET. Pokud jsou souřadnice bodu mimo zobrazovanou plochu je čára přerušena na kraji zobrazované plochy. Oba uvedené příklady doporučujeme prakticky zkusit na počítači. Prvý příklad je spojení pouze dvou bodů v barvě určené kódem palety 2 a překrývacím režimem 0. Ve druhém příkladě je vytvořen zajímavý obrazec čarami ze středu obrazovky do rohu.

BLINE

color specification - určení barvy
 X- coordinate - souřadnice X
 Y- coordinate - souřadnice Y
 palette code - kód palety
 superimpose mode - překrývací režim

Instrukce BLINE mění barvu čáry spojující dané body na obrazovce. Význam a rozsah souřadnic je stejný jako u instrukce SET. Změna barvy je stejná jako u instrukce RESET. Pokud jsou souřadnice mimo zobrazovanou část je změněna barva pouze části zobrazované.

BOX

color specification - určení barvy
 X- coordinate - souřadnice X
 Y- coordinate - souřadnice Y
 palette code - kód palety
 superimpose mode - překrývací režim

Instrukce BOX používá dva páry souřadnic umístěných jako rohy čtverce. X a Y souřadnice mají stejný význam a rozsah jako u instrukce SET. Pokud je parametr určení barvy vynechán (stejně jako u instrukce LINE a SET) je barva dána poslední instrukcí COLOR. Pokud je určen kód palety je čtverec zabarven, pokud je tento parametr vynechán jsou kresleny pouze obrysy čtverce.

CIRCLE

color specification - určení barvy
 X- coordinate - souřadnice X
 Y- coordinate - souřadnice Y
 radius - poloměr
 aspect - hledisko rozlišující mezi X a Y poloměrem

Instrukce CIRCLE umožňuje nakreslení kruhu a nebo elipsy popřípadě

oblouk. Určení barvy je stejné jako u instrukce SET. Pokud určení barvy chybí je barva určena poslední použitou instrukcí COLOR. X a Y souřadnice jsou souřadnice středu kruhu a radius je jeho poloměr. Rozsah jejich hodnot je

$-16384 \leq \text{souřadnice} \leq 16383$

$0 \leq \text{poloměr} \leq 16383$

Plocha, ve které může být kruh zobrazen je určena příkazem INIT. Hledisko určující vztah mezi X a Y poloměrem určuje jejich poměr. Pokud je menší než 1 stává se poloměr X poloměrem a pokud je větší než 1 stává se poloměr Y poloměrem. Mezní hodnota je 1. Start a konec zadány v úhlech (v radiánech) určují, kde se kreslení elipsy začne a kde skončí. Pokud tento parametr chybí je nakreslena elipsa . Pokud je přidán parametr 0 je nakreslen oblouk i s čarami do středu, pokud parametr 0 chybí je nakreslen pouze oblouk. V příkladu je nakreslena elipsa a po stisknutí jakéhokoliv tlačítka tak i oblouk a kruh.

PAINT

palette code	- kód palety
X coordinate	- souřadnice X
Y coordinate	- souřadnice Y
boundary colour	- barva hranice

Instrukce PAINT vyplňuje plochu na obrazovce barvou určenou kódem palety. Pokud kód palety chybí je použita barva určená poslední instrukcí COLOUR.

Barvu hranice je možno vybrat ze 16 barev. Rozsah použití souřadnic je dán nastavením v souhlasu s instrukcí INIT. Pokud není celá plocha uzavřena hranicí je zabarvení plochy zabarveno i mimo plochu. Zabarvení je potlačeno pokud leží bod určený souřadnicemi na hranici nebo v ploše, která byla již zabarvena určenou barvou. Všechny obrázky jsou kresleny složeně z bodů, proto pokud kreslíme čáry a křivky na malé ploše může dojít k vytvoření malých uzavřených smyček. Pokud se to stane je nutné souřadnice bodů určených v této malé ploše částečně zrušit, aby mohlo dojít k zabarvení plochy.

v příkladě si prakticky vyzkoušejte zabarvení dané plochy kruhu a jeho hranice.

PATTERN

color specification	- určení barvy
numeric data	- číselná data
text data	- textová data
palette code	- kód palety
superimpose mode	- překrývací režim

Instrukce PATTERN definuje grafické modely v určené barvě. Význam určení barva je stejný jako u instrukce SET. Pokud není barva určena je použita barva z posledního určení instrukcí COLOUR.

Model, který má být nakreslen je určen použitím číselných a textových dat. Číselná data (± 1 až ± 24) představují počet seřazených 8-bitových bodů vzorové řady a textová data představují individuální bodové modelové řady.

Kreslení modelu je řízeno kladným ukazatelem. Počet bodů modelové řady je určován číselnými daty je zobrazován zespoda nahoru jestliže jsou číselná data kladná a zeshora dolů pokud jsou záporná. Po určení počtu bodů modelové řady jsou kresleny bodové modelové linky 8-bitové (1 znak) zprava od běžného sloupce a jsou zobrazeny. Textová data musí být určena použitím ASCII kódem, který koresponduje s číselnou reprezentací bodových modelových řad.

Binary number - Číslo ve dvojkové soustavě

V příkladu je určeno nastavení ukazatele.

Na obrázku u příkladu dvě je obrázek, jeho popis je dále přeložen.

Binary	- čísla v dvojkové soustavě
Hexadecimal or Character representation	- představitelé znaků a hexadecimalních znaků
position pointer before execution	- poloha ukazatele před provedením

POSITION

X- coordinate - souřadnice X
 Y- coordinate - souřadnice Y

Instrukce POSITION nastaví ukazatel na daný bod obrazovky. Poloha bodu ukazatele na obrazovce je dána následující instrukcí PATTERN, která je zobrazena. Rozsah souřadnic je stejný jako u instrukce SET.

SYMBOL

colour specification - určení barvy
 X- coordinate - souřadnice X
 Y- coordinate - souřadnice Y
 text data - textová data
 horizontal magnification- horizontální zvětšení
 vertical magnification - vertikální zvětšení
 angle code - kód úhlu

Instrukce SYMBOL nakreslí daný grafický model (znak) v dané velikosti a daném úhlu. Při této instrukci je souřadnicemi zadán bod levého dolního rohu obrázku viz obr. na str.6 -66 s natočeným písmenem A o 90 stupňů. Potom je jedna strana obdélníku horizontálního zvětšení a druhá vertikální zvětšení a natočení určeného znaku je dáno kódem úhlu (viz tabulka), kde se natáčení písmena děje ve směru otáčení hodinových ručiček. Pokud je vynechán parametr určení barvy je barva určena poslední použitou instrukcí COLOUR. Rozsah použití souřadnic je stejný jako u instrukce SET. Horizontální a vertikální zvětšení může být celé číslo od 1 do 255. Natáčení znaku se děje od horního levého rohu.

POINT

X- coordinate - souřadnice X
 Y- coordinate - souřadnice Y

Funkce POINT vrací bodu barvu, která mu byla na obrazovce určena. X a Y souřadnice mají rozsah podle nastavení instrukcí INIT.

6.5 INSTRUKCE HUDEBNÍ

MUSIC

note of melody - nota melodie

Instrukce MUSIC produkuje prostřednictvím reproduktoru počítače hudbu nebo zvukové efekty určené znakovým řetězcem proměných. Tři různé melodie mohou být hrány najednou. V určení melodie za slovem MUSIC je pořadí not a individuální části jsou od sebe odděleny středníkem a jedna melodie od druhé je oddělena čárkou. Každá nota je určena určením oktávy, jménem noty, délkou trvání.

(i) Určení oktávy

Základní oktáva melodie je určena formátem "On", kde n je číslo od 0 do 6. Kladné znaménko (+) určuje, že noty budou hrány o jednu oktávu výš než je základní oktáva a znaménko minus (-) určuje, že noty budou hrány o jednu oktávu níž než je základní oktáva. Pokud nejsou noty označeny znaménka jsou hrány v základní oktávě. Znaménka (+) popřípadě (-) nelze použít pokud bychom chtěli oktávu mimo rozsah 6 oktáv. Například je nepřípustné při základní oktávě 06 chtít noty se znaménkem (+).

(ii) Určení noty

Výška noty je určena písmenem podle obrázku na str. 6 - 68 písmeno R určuje mezeru.

(iii) Určení délky trvání noty

Délka trvání určuje délku trvání jednotlivé noty. Délka trvání noty je určena číslicí od 0, která má význam 1/32, do 9, která má význam celé noty. Pokud je určeno písmeno R je číslicí určena délka trvání mezery. Na str. 6-69 nahoře je obrázek určující význam jednotlivých číslic délky.

V příkladě je uvedena melodie O Zuzano, složená Stephenem Fosterem.

Následující tabulka určuje další možnosti při určení melodie

	n= celé číslo, rozsah	Popis
On	0 až 6	Nastavení oktávy. Frekvence A je 440 Hz při $n = 2$
Nn	0 až 83	Určuje noty přímo číslem a ne číslem oktávy a jménem noty. Hodnoty čísel, korespondující s výškou not jsou zobrazeny v tabulce na str. 6 - 70. Čísla od 0 do 8 nejsou použita.
Tn	1 až 7	Nastavuje tempo stejně jako instrukce TEMPO
Vn	0 až 15	Nastavuje intenzitu zvuku. Při nastavení $n=15$ je maximum a při nastavení $n=0$ je intenzita zvuku nulová.
Sn	0 až 7	Nastavuje zvukové modely podle tabulky na str. 6 - 70.
Mn	1 až 255 $l = cca 10m/s$	Vždy používáme S parametr k určení stupně, na kterém bude zvukový model změněn. Maximální stupeň je 1 a potom je snížení. Zvukové modely tedy mají pomalejší přehrání s rostoucím n .
Ln	0 až 9	Nastavuje délku noty

Na straně 6 - 70 nahoře je tabulka korespondujících čísel s notami.

Na téže straně uprostřed je tabulka zvukových modelů (v podstatě se jedná o zvukové vlny). X rozměr má zde čas a Y rozměr zde má intenzita zvuku.

Instrukce MUSIC převádí data do bufferu a potom do zvukového generátoru hned jak je instrukce počítačem provedena. Během provádění této instrukce je možno změnit obraz na displeji. Provádění instrukce MUSIC je možno zastavit stisknutím kláves SHIFT a BREAK. Provádění hudebních instrukcí lze také zastavit následujícími instrukcemi.

MUSIC STOP - zastaví zvukový generátor

MUSIC WAIT - přeruší provádění programu až do úplného naplnění bufferu.

MUSIC INIT - Inicializuje hudební nastavení na "02V15L5T4S8M255".

TEMPO

numeric expression - číselný výraz

Instrukce TEMPO nastavuje tempo, ve kterém bude instrukce MUSIC prováděna. Nastavení tempa se děje v rozsahu od 1 do 7. Tempo jedna je nejpomalejší a tempo sedm nejrychlejší.

SOUND

pitch - stupeň

duration - délka trvání

Instrukce SOUND generuje zvuk určený stupněm a délkou trvání.

(i) Stupeň

Stupeň je určen tabulkou, která je obdobná jako pro noty vyjádřené čísly. Komorní A zde má hodnotu 440 Hz.

(ii) Délka trvání

Délka trvání je určena jednotkou 1/100 sekundy práce zvukového generátoru. Délka trvání musí být numerický výraz nabývající hodnoty od 0 do 65535.

SOUND = (register,data) instrukce používá přímé řízení zvukového generátoru. Programovatelný zvukový generátor (zkratka PSG) umí najednou generovat tři tóny jako jeden zvuk. Tabulka PSG registrů je následující.

Číslo registru			Data
0	Tón 0	Frekvence	Celé číslo od 1 do $2^{10}-1$
1	Tón 0	Intenzita	Celé číslo od 0 do 15
2	Tón 1	Frekvence	stejná jako registr 0
3	Tón 1	Intenzita	stejná jako registr 1
4	Tón 2	Frekvence	stejná jako registr 0
5	Tón 2	Intenzita	stejná jako registr 1
6	Zvuk	Frekvence	Zvuková data
7	Zvuk	Intenzita	stejná jako registr 1

NOISE

melody	- melodie
note name	- jméno noty
duration	- délka noty

Instrukce NOISE generuje zvuk určený melodií n parametrem. Význam parametru melodie n je stejný jako instrukce MUSIC. Tato instrukce může generovat najednou dva zvuky. Parametry určující tyto dvě části musí být odděleny středníkem. Všechny melodie po sobě jdoucí musí být odděleny čárkou.

6.6 INSTRUKCE OVLÁDÁNÍ TISKÁRNY

PTEST

Příkaz PTEST vytiskne na tiskárně čtverce v barvě černé, modré, zelené a červené.

Tento příkaz lze použít pouze v textovém režimu.

PMOOE

PMOOE příkaz určuje režim použití barevné plottové tiskárny.

PMOOE TN

Příkaz PMOOE TN převádí tiskárnu z grafického režimu do režimu textového a nastaví velikost znaků na 40 znaků na řádek.

PMODE TL

Příkaz PMODE TL převádí tiskárnu z grafického režimu do režimu textového a nastaví velikost znaků na 26 znaků na řádek.

PMODE TS

Příkaz PMODE TS převádí tiskárnu z grafického režimu do textového režimu a nastavuje velikost znaků na 80 znaků na řádek.

PMODE GR

Příkaz PMODE GR přepíná z textového režimu do grafického režimu. Po přepnutí do tohoto režimu je nutno v programech v BASICu před použitím znaků určit jejich velikost. O textového režimu také přejdeme po stisknutí tlačítek SHIFT a BREAK a nebo po provedení příkazu STOP.

Na straně 6 - 74 nahoře je znázorněna velikost znaků při režimu TL (80 znaků), TN (40 znaků) a TS (26 znaků).

Při použití příkazu PMODE GR převedeme tiskárnu do grafického režimu a musíme vzít v úvahu, že pero vychází z posice 0,0. Je nutné zadávat pouze pohyby pera, které je možno realizovat na papíře a není možné zadávat pohyby pera, které vedou mimo papír a mohly by poškodit tiskárnu. Na obrázcích na straně 6 -74 vidíme možné pohyby pera.

Direction of paper flow - směr posunu papíru

X souřadnice je maximálně 480 a Y souřadnice je od -999 až 999.

PCOLOR

Příkaz PCOLOR určuje barvu jaká bude použita při tisku znaků i grafiky. Tento příkaz může být použit jak v textovém tak i grafickém režimu. Barvy korespondují s čísly podle tabulky

- 0 - černá
- 1 - modrá
- 2 - zelená
- 3 - červená

Příklad 10 PCOLOR 1 Tiskárna je nastavena na modrou barvu

PSKIP

number of lines - počet řádků.

Tento příkaz posune papír v tiskárně o zadaný počet řádků. Pokud je počet řádků kladný je papír posunován kupředu, pokud je záporný je posunován nazpět. Příkaz lze použít pouze v textovém režimu.

PAGE

number of lines - počet řádků

Příkaz PAGE určuje počet řádků na řádku. Počet řádků musí být celé číslo od 1 do 72. Příkaz PAGE také nastavuje první řádek následující stránky. Tento příkaz lze také použít v textovém režimu. V příkladu je určena stránka 20 řádků.

PRINT/P

Instrukce PRINT/P tiskne data na tiskárně ve stejném formátu jaký je použit na obrazovce. Obě rozešlé čárka i středník a tabelátor mají stejný význam jako při normálním použití instrukce PRINT. Funkce tiskárny se zadávají následujícím formátem

PRINT/P CHR\$ (kontrolní kód)

V příkladě je vytištěno ABCD a ve druhém případě je proveden posun o jeden řádek.

Poznámka:

V instrukci PRINT/P musí být ovládací kódy buď jako zvláštní instrukce a nebo mohou být sdruženy do jedné, ale není možné tyto ovládací kódy sečítat pomocí znaménka (+). Viz příklad
 Invalid - nemožné
 valid - možné zapsání instrukce.

PRINT/P USING

palette code - kód palety
 format string- formátový řetězec

Instrukce má stejný význam s tím rozdílem, že výstupní zařízení je tiskárna.

PLINE

line type - typ čáry

Instrukce PLINE kreslí plnou nebo čárkovanou čáru z běžné polohy pera do polohy určené souřadnicemi a nebo nakreslí čáru z bodu označeného souřadnicemi x1,y1 do bodu x2,y2. Souřadnice x-ové musí být v rozsahu -480 až 480, a y-ové souřadnice musí být v rozsahu -999 až 999. Určení typu linky musí být celé číslo od 1 do 16. Přehled čar a jejich určení je nastr. 6 -77. PLINE instrukce lze použít pouze v grafickém režimu. V příkladu je nakreslen čtverec o straně 200 plnou čarou.

RLINE

line type - typ čáry

Instrukce RLINE nakreslí čáru z bodu daného prvými souřadnicemi do bodu označeného druhými souřadnicemi atd. Rozsah souřadnic a typ čáry je stejný jako u instrukce PLINE. Rozdíl mezi těmito dvěma instrukcemi je v tom, že PLINE používá absolutní hodnotu souřadnic a RLINE používá relativní hodnoty souřadnic.

PMOVE

X - coordinate - souřadnice X

Y - coordinate - souřadnice Y

Tato instrukce posune pero do místa určeného souřadnicemi x,y. Rozsah použití souřadnic musí být stejný jako u instrukce PLINE. Instrukce je použitelná pouze v grafickém režimu.

RMOVE

X- coordinate - souřadnice X

Y- coordinate - souřadnice Y

Instrukce RMOVE přesune pero do místa určeného souřadnicemi x,y. Rozdíl mezi instrukcí PMOVE a RMOVE je v tom, že instrukce PMOVE používá absolutní hodnotu souřadnic a instrukce RMOVE relativní hodnotu souřadnic. Rozsah použití souřadnic je stejný jako u instrukce RLINE. Při provádění příkladů si buďte jisti, že máte v tiskárně papír. Instrukce lze použít jen v grafickém režimu.

PHOME

Instrukce PHOME posune pero do základní polohy (0,0). Instrukce lze použít pouze v grafickém režimu.

HSET

Instrukce HSET určí pro pero nový počátek souřadnic. Přesuneme pero do nové polohy pomocí instrukce PMOVE a potom instrukcí. HSET stanovíme toto místo jako nový počátek. Tato instrukce lze použít pouze v grafickém režimu. V příkladu je vidět posunutí počátku do bodu (-240,240).

GPRINT

size - velikost

angle - úhel

Instrukce tiskne daný znak v určené velikosti a v určeném úhlu. Velikost může být jakékoliv číslo od 0 do 63. Pokud je velikost rovna 0 je možno zapsat 80 znaků na řádek. Při velikosti = 1 je možno psát 40 znaků na řádek. Úhel může být celé číslo od 0 do 3. Úhel otáčí znakem podle levého spodního rohu. Na straně 6 - 81 nahoře jsou příklady tisku. Pokud velikost a úhel nejsou určeny je velikost = 1 a úhel = 0.

AXIS

axis - osa
pitch - rozteč na stupnici
repetition - opakování

Instrukce AXIS nakreslí osu X pokud je osa = 1 a osu Y pokud je osa = 0. Počet značek na stupnici je dán opakováním a jejich vzdálenost roztečí. Rozteč musí být celé číslo od -999 do 999, opakování musí být celé číslo od 1 do 255. Instrukce AXIS lze použít pouze v grafickém režimu. V příkladě je vidět nakreslení osového kříže v kartézské soustavě se stupnicí od -240 do 240 s intervalem 10.

PCIRCLE

X - coordinate	- souřadnice X	rozsah od -999 do 999
Y - coordinate	- souřadnice Y	rozsah od -999 do 999
radius	- poloměr	rozsah od 0 do 999
starting angle	- výchozí úhel	
ending angle	- koncový úhel	
step angle	- krok úhlu	

Instrukce PCIRCLE kreslí kruh nebo oblouk ve směru otáčení hodinových ručiček. Kruh (oblouk) má poloměr, střed (x,y) počáteční úhel, koncový úhel. Celý kruh má počáteční úhel roven 0, koncový 360 a krok úhlu 0.2. Tato instrukce ve skutečnosti kreslí mnohoúhelník, proto krok úhlu musí být tak malý jak je to jen možné, aby byl obrazec nakreslen "hladce". Startovací úhel musí být menší než konečný. Pokud je krok úhlu roven nule je střed spojen s počátečním bodem a také je spojen se středem koncový bod. Instrukce lze použít pouze v grafickém režimu.

LIST/P

starting line number - číslo počátečního řádku
 ending line number - číslo koncového řádku

Tento příkaz vytiskne část nebo všechny řádky programu z paměti. Podívejte se na instrukci LIST, kde jsou popsány podrobnosti. Pokud jsou použity v tištěném programu grafické symboly řada z nich bude vytištěna v jiné barvě. Tento příkaz lze použít pouze v textovém režimu.

HCOPY

Příkaz HCOPY provede vytištění obrazovky na tiskárnu. Lze použít pouze pro mozaikové tiskárny. Ne pro MZ 1P16.

PLOT

ON - zapnuto
 OFF - vypnuto

Instrukce PLOT ON umožňuje použít barevnou plottrovou tiskárnu jako displej. Instrukce PLOT ON nastaví počet tištěných znaků na řádek podle toho jak je nastavena obrazovka. (40 nebo 80 znaků na řádek). Tato instrukce je použitelná pouze při připojení tiiskárně a textovém režimu. Příkaz CONSOLE je při použití instrukce PLOT ON nepoužitelný.

Pokud není znak, který má být tištěn naprogramován v tiskovém generátoru je místo něj vytištěna tečka (.) Tlačítka INST, DEL a posuv kurzoru naz-ět nelze použít. tlačítka CTRL + G lze používat k výměně barva pera.

Instrukce PLOT OFF zruší instrukci PLOT ON. Také instrukce INIT"CRT.Mn" zruší instrukci PLOT ON.

6.7 INSTRUKCE OVLÁDÁNÍ STROJOVÉHO JAZYKA

PEEK

address - adresa

Tato funkce přenáší obsah určené adresy jako desetinné číslo od 0 do 255. Adresa může být také desetinné číslo od 0 do 65535 a nebo 4 místné hexadecimální číslo od \$0000 do \$FFFF.

V příkladě jsou zobrazena data uložená od adresy 40960 (\$A000) do adresy 40975 (\$A00F).

POKE

address - adresa

data - data

Instrukce POKE zapíše serii datových bytů začínajíc od určené adresy. Adresa může být desetinné číslo od 0 do 65535 nebo 4 místné hexadecimální číslo od \$0000 do \$FFFF. Data mohou mít rozsah od 0 do 255 nebo od \$00 do \$FF. Tato instrukce zapisuje data do jakékoliv části paměti v soulase s nastavením instrukcí LIMIT. Buďte opatrní při použití této instrukce, aby nedošlo k narušení MONITORu a nebo interpretru BASIC. V příkladě je znázorněno použití čísel desetinného a hexadecimálního stejným způsobem.

INP@

port number - číslo portu

variable - proměnná

Instrukce INP@ čte data ze vstupního portu (data jsou 8-bitová) určeného číslem portu, převádí je na desetinné číslo a uloží je do příslušné paměti. Číslo portu může být od 0 do 127 (hexadecimálně \$00 do \$7F). Adresa portu může být od 128 do 255 (\$80 až \$FF) jsou rezervovány pro použití přídatných zařízení.

OUT@

port number - číslo portu
 numeric expression - číselný výraz

Instrukce OUT@ převádí desetinné číslo určené v číselném výrazu (0 až 255) do binárního formátu a posílá je výstupním portem určeným číslem portu. Číslo portu může být od 0 do 127 (\$00 do \$7F). Adresa portu 128 až 255 (\$80 až \$FF) je určena pro přídatná zařízení.

Přídatná zařízení jsou řízena přenášenými daty na I/O port. Následek špatného čísla portu může být za následek špatnou funkci přídatného zařízení.

USR

address - adresa
 input string variable - vstupní proměnný řetězec
 output string variable - výstupní proměnný řetězec

Funkce USR přenáší ovládání na strojový jazyk (program ve strojovém jazyce) začínající na dané adrese.. Jako s CALL (adresa) kontrola je vrácena na instrukci následující funkci USR pokud je návratná instrukce RET nebo RETcc v programu ve strojovém jazyce. Adresa musí být desetinné nebo 4-místné hexadecimální číslo.

Parametry jsou převáděny do následujících registrů pokud hlavní program přivádí ovládání do strojového jazyka.

DE registr : Počáteční adresa (vstupní proměnný řetězec) v paměti

B registr : délka vstupního proměnného řetězce

IX registr : adresa chybového programu, pokud je deklarován

Strojový jazyk nahrává výsledky do následujících registrů pokud je ovládání vráceno do hlavního programu.

OE registr : počáteční adresa (výstupního proměnného řetězce) v paměti

B registr : délka výstupního proměnného řetězce

Následující kroky jsou nezbytné pokud je ve strojovém jazyku zadán program chyb :

1. Deklarovat program chyb v BASICu pomocí instrukce ON ERROR GOTO
2. Zapište části programu do A registru s kódem chyb tak, aby program při chybě skočil na adresu určenou v registru IX.

LIMIT

address - adresa

Instrukce LIMIT limituje místo použitelné pro BASIC. Adresa nastavuje horní hranici BASICké části, část následující za adresou do adresy \$FEFF (65279) je nastavena jako uživatelská část. Část od \$FF00 do \$FFFF je použito jako monitorem jako pracovní část, nemůže být použita pro uživatelský program. Adresa může být buď celé dekadické číslo nebo 4-místné hexadecimální číslo. Pokud používáme dohromady BASICký program dohromady s programem ve strojovém jazyce je nutné si udělat místo pro uživatelskou část. Instrukce LIMIT musí být použita na začátku programu. Instrukce LIMITMAX ruší nastavení provedené instrukcí LIMIT. Vpříkladě vidíme limitování části paměti použitelné pro BASIC

BASIC program area - část paměti pro BASIC

User area - uživatelská část paměti

Instrukce LIMIT MAX ruší toto nastavení.

6.8 CHYBOVÉ INSTRUKCE

ON ERROR GOTO

line number - číslo řádku

label - návěstí

Instrukce ON ERROR GOTO převede provádění programu na řádek určený číslem nebo návěstím v případě, že dojde k chybě. ERN a ERL systémové proměnné mohou být použity ke kontrole chybového režimu na daném řádku. V příkladě vidíme použití instrukce

UN ERROR GOTO a instrukce RESUME.

RESUME

line number - číslo řádku

Instrukce RESUME vrací ovládání do hlavního programu z programu chybového.

Systém si ponechá v paměti číslo řádku, na kterém došlo k chybě. Provádění programu je vráceno na tento řádek nebo na jiný řádek po opravě chyby. Instrukce RESUME může být použita v následujících formách.

RESUME

- vrátí se na chybový řádek

RESUME NEXT

- vrátí se na řádek, který následuje za chybovým řádkem

RESUME (číslo řádku)

- vrátí se na řádek určený instrukcí

RESUME 0

- vrátí se na začátek hlavního programu

Obvykle použijeme instrukci RESUME k návratu běhu programu z chybového programu do hlavního programu. Pokud nemůže být instrukce realizována nastane chyba.