

Mikromonitor

64

a

MD-DOS

# MDDOS

## Obsah:

1.	Úvod.....	2
2.	Spuštění a ovládání programu.....	2
3.	Práce s MDDOSem.....	3
3.1.	Práce s podzařízením Dx:.....	4
3.2.	Důležité adresy v MDDOSU.....	6
4.	Popis MDDUP.....	6
5.	Popis jednotlivých příkazů.....	7
5.1.	1-8 DIR Dx:.....	7
5.2.	^1-8 Special DIR.....	7
5.3.	A:Disk directory.....	7
5.4.	B:Basic B:ON/OFF.....	7
5.5.	C:Copy file.....	8
5.6.	D:Delete file.....	8
5.7.	E:Rename file.....	8
5.8.	F:Lock file.....	9
5.9.	G:Unlock file.....	9
5.18.	H:New RAMTOP.....	9
5.11.	I:Format Disk.....	9
5.12.	J:Undelete file.....	9
5.13.	K:Binary save.....	10
5.14.	L:Binary load.....	10
5.15.	M:Run at Adress.....	10
5.16.	N:Turbo copy.....	10
5.17.	Ø:Turbo->File.....	11
5.18.	P:Set Printers.....	11
5.19.	Q:Mikro Monitor.....	12
5.20.	R:TURBO 2000.....	12
5.21.	S:Tape Speed.....	12
5.22.	T:Tape Mode.....	12
5.23.	U:Turbo load.....	12
5.24.	V:Configurate.....	12
5.25.	W:Basic Turbo Load.....	13
5.26.	X:Basic Turbo Save.....	13
5.27.	nevyužito.....	13
6.	Chybová hlášení operačního systému.....	13
7.	Závěr.....	14
8.	Připojení zapisovacích zařízení k počítači.....	15

## 1. Úvod

Operační systém MDDOS vznikl ze snahy vytvořit jeden operační systém, který by dokázal obslužit co nejvíce periferních zařízení bez nutnosti nahrávat samostatný program. Poslední vytvořená verze umí obsluhovat tato periferní zařízení:

Disketová jednotka se třemi formáty

- single
- medium
- double

Magnetofon s možností komunikace

- Turbo 1kB bloky
- Turbo 2000
- standardní formát

- Turbo Tape (TTDOS)

Tiskárny: standardní sér. rozhraní

sériové rozhraní  
parallelní rozhraní  
- BT100  
plotry

- ATARI 1029
- DTR protokol
- CENTRONICS
- ALFI
- XY 4140-50
- ARITMA 0507

Kromě toho dokáže MDDOS obslužit RAMdisk až 256kB a RAMdisk vytvořený z paměti umístěné pod operačním systémem (22kB).

Obslužné programy všech těchto zařízení jsou značně rozsáhlé a DOS napsaný tradičním způsobem ve dvou souborech DOS a DUP by zabíral natolik velkou část paměti RAM, že programy psané pro standardní DOS 2.0 nebo DOS 2.5 by nešly použít. Částečně tento problém řeší Sparta DOS rozdělením jednotlivých podprogramů do mnoha krátkých souborů, které se v případě potřeby nahrávají do paměti počítače. Toto řešení sice vyhovuje pro konfiguraci počítač-disketa, ale pro uživatele magnetofonů je prakticky nepoužitelné. Problém se vyřešil umístěním co největší části DOSu do cartridge.

Operační systém MDDOSu je spolu s Mikro Monitorem umístěn v 28kB paměti EPROM a zbytek paměti do 32kB nebo do 64kB lze využít pro programy požadované uživatelem při výrobě cartridge.

## 2. Spuštění a ovládání programu

Po zapnutí počítače se zasunutým MDDOSem stačí krátce stisknout tlačítko cartridge s následným RESET. Objeví se černá obrazovka a nápis:

Mikro Monitor 64kB by Milan Dadok <c> 29.3.91

OK <kurzor>

Pro studený start z Mikro Monitoru do MDDOSu se používá 2xESC a RETURN. provedou se následující operace:

- iniciace charakteristických hodnot zařízení
- nastaví se vektory RESINI a DOSINI
- provede se skok přes RESINI a DOSINI

Z toho plyne nutnost použití studeného startu jen při prvním startu MDDOSu. Pro teplý start z Mikro Monitoru pak stačí příkaz &, v Basicu příkaz DOS, v ACTION! D apod.

TURBO Tape Disk Plotr / MDDOS /  
<c> 29.03.91 by M. Dadok

D:12BTEM-R Memory:128kB Tape Speed:12  
Memlo: 1849 Ramtop:A0 Tape mode :SS

1-8.-DIR Dx:	1-8 Special DIR
A:Disk directory	N:Turbo copy
B:Basic B:On/OFF	O:Turbo->File
C:Copy file	P:Set printers
D:Delete File	Q:Micro Monitor
E:Rename file	R:TURBO 2000
F:Lock file	S:Tape Speed
G:Unlock file	T:Tape Mode
H:New Ramtop	U:Turbo load
I:Format Disk	V:Configurate
J:Undelete file	W:Basic Turbo Load
K:Binary save	X:Basic Turbo Save
L:Binary load	Y:
M:Run at adress	Z:

## 2. Práce s MDDOSem

MDDOS definuje do systému zařízení D: a P:. Zařízení D: obsluhuje disketovou jednotku, RAMdisky a Turbo magnetofon. Zařízení P: obsluhuje tiskárny a plotry. MDDOS využívá skutečnosti, že každé definované zařízení může mít až 8 podzarizení standardně definována následovně:

Tiskárna	č.1 P1: - Parallel Printer
	č.2 P2: - BT100
	č.3 P3: - Serial Printer
	č.4 P4: - DTR Printer
	č.5 P5: - Plotr
	č.6 P6: - Plotr
	č.7 P7: - Plotr
	č.8 P8: - Plotr

Disketa	č.1 D1: - disketová jednotka	č.1	kód 1
č.2 D2:	- disketová jednotka	č.2	2
č.3 D3:	- TTDOS Magnetofon		B
č.4 D4:	- Turbo 1kB bloky		T
č.5 D5:	- ROMdisk		E
č.6 D6:	- malý RAMdisk		M
č.7 D7:	- CMOS RAMdisk		S
č.8 D8:	- velký RAMdisk		R

Pokud máme v počítači zabudován pouze CMOS RAMdisk, bude se prezentovat jako D8:R.

Toto uspořádání lze pomocí funkce MDDUP měnit. Viz 5.24 - Configurate. U zařízení D: musí být zadán také název souboru - filename, sestávající z jednoho až osmi znaků názvu a tří znaků typu souboru.

Např.

D5:CAPEK30L.COM	
D6:TEST	=D6:TEST. - nedefinovaný typ souboru
D8:B	

V názvu je dovoleno používat pouze velká písmena a číslice, pokud nejsou na prvním místě názvu, a lze také použít tzv. WILDE CARDS, hvězdičku a otazník. Otazník nahrazuje jeden znak, hvězdička skupinu znaků.  
Např.

D1:*.*	- libovolný soubor
D1:*.BAS	- libovolný soubor typu BAS
D1:*.?IC	- libovolný soubor s typem souboru končícím na IC
D1:?????1*.*	- libovolný soubor, který má na pátém místě znak 1

Znaky, uvedené za hvězdičkou, jsou kromě tečky ignorovány. V případě tvoření nového názvu souboru je nezbytné uvést celý název, neurčité znaky jsou nahrazeny mezerou.

### 3.1. Práce s podzařízením Dx

Lze provádět tyto operace:

A. Podzařízení označované kódem 1,2,M,R

příkaz OPEN	ICCOM=03
ICAX1	funkce
\$04	čtení souboru
\$06	čtení adresáře - DIR
\$07	spec. čtení adresáře - SPEC.DIR
\$08	zápis souboru
\$09	připsání na konec souboru
\$0C	aktualizace souboru (čtení i zápis)
příkaz INPUT	ICCOM=\$05 čtení ze souboru po nejbližší RET.
příkaz GET	ICCOM=\$07 čtení daného počtu byte
příkaz PRINT	ICCOM=\$09 vyslání po první RETURN (\$9B)
příkaz PUT	ICCOM=\$0B vyslání daného počtu byte
příkaz CLOSE	ICCOM=\$0C ukončení operace
příkaz STATUS	ICCOM=\$0D zjistit stav souboru
po provedení STATUS	ICSTA=\$01 soubor nalezen, není uzamčen
	ICSTA=\$A7 soubor nalezen, je uzamčen
	ICSTA=\$AA soubor nenalezen

Následující příkazy se používají bez předchozího OPEN (kromě NOTE a POINT)

příkaz RENAME	ICCOM=\$20 přejmenování souboru
příkaz DELETE	ICCOM=\$21 vymazání specifikovaných souborů
příkaz UNDELETE	ICCOM=\$22 obnovení souboru
příkaz LOCK	ICCOM=\$23 uzamčení souboru
příkaz UNLOCK	ICCOM=\$24 odemčení souboru
příkaz POINT	ICCOM=\$25 nastavení absolutní adresy sektoru
příkaz NOTE	ICCOM=\$26 čtení absolutní adresy sektoru
příkaz FORMAT	ICCOM=\$FE formátování disku
ICAX2	druh formátování
\$00	SINGLE
\$01	DOUBLE
\$02	MEDIUM
>\$80	CLEAR - maže adresář, nastavuje VTOC a zachová původní hustotu

Pozn. U podzařízení R je vždy formát DOUBLE (adresář pro 64 souborů), u podzařízení M vždy SINGLE (110 sektorů, adresář pro 8 souborů)

B. Podzařízení označované kódem E

příkaz OPEN	ICCOM=03
ICAX1	funkce
\$04	čtení souboru
\$06	čtení adresáře - DIR
\$07	spec. čtení adresáře - SPEC.DIR
příkaz INPUT	ICCOM=\$05 čtení ze souboru po nejbližší RET.
příkaz GET	ICCOM=\$07 čtení daného počtu byte
příkaz CLOSE	ICCOM=\$0C ukončení operace
příkaz STATUS	ICCOM=\$0D zjistit stav souboru
příkaz POINT	ICCOM=\$25 nastavení absolutní adresy sektoru
příkaz NOTE	ICCOM=\$26 čtení absolutní adresy sektoru Ostatní příkazy nelze v ROMdisku realizovat.

#### C. Podzařízení označované kódem B

příkaz OPEN	ICCOM=03
ICAX1	funkce
\$04	čtení souboru
\$06	čtení adresáře - DIR
\$08	zápis souboru
příkaz INPUT	ICCOM=\$05 čtení ze souboru po nejbližší RET.
příkaz GET	ICCOM=\$07 čtení daného počtu byte
příkaz PRINT	ICCOM=\$09 vyslání po první RETURN (\$9B)
příkaz PUT	ICCOM=\$0B vyslání daného počtu byte
příkaz CLOSE	ICCOM=\$0C ukončení operace

#### D. Podzařízení označované kódem T

příkaz OPEN	ICCOM=03
ICAX1	funkce
\$04	čtení souboru
\$06	čtení adresáře - DIR
\$08	zápis souboru
příkaz INPUT	ICCOM=\$05 čtení ze souboru po nejbližší RET.
příkaz GET	ICCOM=\$07 čtení daného počtu byte
příkaz PRINT	ICCOM=\$09 vyslání po první RETURN (\$9B>)
příkaz PUT	ICCOM=\$0B vyslání daného počtu byte
příkaz CLOSE	ICCOM=\$0C ukončení operace Ostatní příkazy nejsou definovány.

U všech podzařízení jsou při čtení definovány tyto stavy operace:

- \$01 - čtení v pořádku
- \$03 - právě přečten poslední byte
- \$88 - pokus o čtení byte, než soubor má

#### E. Jednotlivá podzařízení P

příkaz OPEN	ICCOM=\$03
ICAX1	funkce
\$04	čtení z původního zařízení E: se současným zápisem na tiskárnu
\$08	tisk na -tiskárně
\$09	tisk na tiskárně
\$0C	čtení ze zařízení E: se současným zápisem jen na tiskárnu

příkaz INPUT                    ICCOTM=\$05 čtení ze zařízení E: se současným

	zápisen na tiskárnu po nejbližší RETURN
příkaz GET	ICCOM=\$07 čtení daného
počtu byte z E: se zápisen	na tiskárnu
příkaz PRINT	ICCOM=\$09 vyslání jen po první RETURN
příkaz PUT	ICCOM=\$0B vyslání daného počtu byte
příkaz CLOSE	ICCOM=\$0C ukončení tisku

Používá-li paralelní obsluha paralelní port, OPEN jej vždy nastaví pro výstup a CLOSE jej nastaví pro vstup, vyjma XY 4140. Při obsluze XY 4140 CLOSE nenastavuje port do vstupního režimu. Paralelní a sériové DTR rozhraní standardně mění kód \$9B na \$0A (par.) nebo na \$0D (DTR). Pro ovládání záměny jsou zvoleny tyto řídící kódy:

ESC \$9B	vypnouti záměny
ESC \$9C	zapnouti záměny

Záměnu je nutno vypnout při HAROCOPY!!

### 3.2 Důležité adresy v MDDOSu

\$0700	RESET procedura
\$0709	RESET procedura
\$070C	obsluha SIO s RAMdisky
\$070F	čtení binárního souboru
\$0712	start MDDUP
\$0735	\$00 - binární soubor se bude startovat, \$FF - nebude startovat

### 4. Popis MDDUP

Tento program (Disk Utility) slouží k jednoduchému provádění příkazů DIR, COPY, LOCK ad., k nastavování některých parametrů MDDOSu, a také ke komunikaci s magnetofonovým systémem Turbo 2000.

Po odstartování MDDOSu se zobrazí hlavní menu a v záhlavi je možné čist některé údaje.

Memory: signalizuje velikost paměti RAM počítače zjištěnou během studeného startu MDDOSu.

Memlo: ukazuje hexadecimálně první volnou adresu v paměti RAM. Od adresy \$0700 až po tuto adresu je uložen MDDOS. Některé programy, např. Turbo-Basic, tuto adresu mění, proto pozor!

Popis ostatních údajů naleznete v kap. 5.

MDDOS reaguje na některé kombinace stisknutých kláves:

SHIFT+CONTROL+TAB	studený start počítače, skok na \$E477
SHIFT+CONTROL+I	nastaví paralelní port do vstupního režimu
SHIFT+CONTROL+O	nastaví paralelní port do výstupního režimu

Každý prováděný příkaz MDDUPu lze přerušit stiskem klávesy BREAK.

Všechna čísla, kromě příkazu Tape speed, se zadávají v hexadecimálním tvaru.

Pro zjednodušení zadávaných názvů, souborů se každý název před vykonáním povelu zpracuje podle následujících příkladů:

zadání	výsledek po úpravě MDDUPem
5:TEST.DAT	D5:TEST .DAT
8TEST.DAT	D8:TEST .DAT
*.*	D1:*. *

## 5. Popis jednotlivých příkazů

### 5.1. 1-8:DIR Dx:

Výpis adresáře - Directory. Stiskem klávesy 1 až 8 se na obrazovku vypíše obsah adresáře zařízení D1 až D8.

Např.

```
MEDIUM : SINGLE  
CAPEK30 COM 047  
880 FREE SECTORS
```

Na prvním řádku se vypíše, jaké je dané podzařízení médium - SINGLE, DOUBLE, MEDIUM, TAPE.

Za jménem souboru je uvedena jeho sektorová délka (SINGLE, MEDIUM-125 byte, DOUBLE-252), a podzařízení B (TTDOS) číslo za jménem programu signalizuje číslo pravé přečteného bloku.

Je-li jméno v závorkách, soubor neumí být přečten DOS 2.0.

Je-li před jménem hvězdička je soubor uzamčen, to znamená, že ho nelze přepsat nebo vymazat.

U podzařízení B (TTDOS) se v případě, že nahrávka má dvojitě bloky, zobrazí před jménem #.

### 5.2. ^1-8:Special DIR

Výpis adresáře včetně zrušených (skrytých) souborů. Před zrušeným souborem se zobrazí znak =, před skrytám souborem znak ? (vznikne např. vypnutím disketové jednotky nebo stiskem, tlačítka BREAK během zápisu do souboru). Soubor lze do jisté míry obnovit viz 5.12.

### 5.3. A:Disk directory

dotaz: Spec. filename,Output filename

Výpis adresáře na specifikované zařízení nebo výpis jen určitých souborů.  
Např.

*.PIC	na obrazovku se vypíší všechny názvy souborů s typem PIC
,P:	na tiskárnu se vypíše celý adresář
*.PIC,P:	na tiskárnu se vypíší všechny názvy souborů s typem
PIC ,D8:DIRA1.TXT	do RAMdisku se pod názvem DIRA1.TXT vypíše celý adresář 5*.COM,8DIRA0.TXT do RAMdisku se pod názvem DIRA0.TXT vypíší z ROMdisku
	všechny názvy souborů s typem COM

### 5.4. B: Basic ^B:ON/OFF

Skok do Basicu nebo Cartridge, Není-li připojen, zobrazí se NO BASIC/CARTRIDGE. Pro připojení Basicu nebo cartridge slouží příkaz SHIFT+B. Stiskem této kombinace kláves se zobrazí dotaz:

```
Initialize Basic or Carriage (0-3)  
B      - připojí se Basic  
0      - start cartridge v nejnižší části 64kB EPROM  
1      - start cartridge v druhé nejnižší části 64kB EPROM  
2,3    - start cartridge ve vyšších částech 64kB EPROM
```

Pak se zobrazí dotaz:

```
Restart (Y/N)
```

Startujeme-li Basic nebo cartridge poprvé, musíme stisknout Y, což značí studený start. N provede teplý start.

Opakováný stisk SHIFT B zapíná/vypíná Basic.

volby 0-3 jsou využity pro startování programů speciálně vytvořených pro práci v cartridge (ACTION!, Basic XE, MAC 65) a v každém MDDOSu 32/64kB nemusí být využity.

#### 5.5. C:Copy file

dotaz: Input file (@output file)

Univerzální kopírovací program. Lze kopírovat jeden nebo více souborů na stejně nebo jiné zařízení.

Např.

CAPEK30.COM	- kopírování souboru CAPEK30.COM z D1: na D1: Počítač počká na výměnu disku.
CAPEK30.COM,CAPEK.COM	- na D1: se vytvoří soubor CAPEK.COM shodný se souborem CAPEK30.COM
CAPEK.TXT,E:	- obsah souboru z D1: na obrazovku. V tomto případě nelze použít Wilde cards.
CAPEK30.COM,8	- soubor CAPEK30.COM z D1: do D8:
*.COM,6	- všechny soubory s typem COM z D1: do D6:
? ,8	- soubor s jednoznačkovým názvem z D1: do D8:

Při použití Wilde cards v názvu se během kopírování vypisuje úplný název souboru.

Zadáme-li za jméno souboru /A (lomítko A), budou se soubory kopírovat na konec již existujících souborů (APPEND).

Při použití Wilde cards v názvu, můžeme zadáním /Q (lomítko Q) za jméno souboru způsobit dotaz programu, zda daný soubor odpovídající zadanému názvu kopirovat.

#### 5.6. D:Delete file

dotaz: Delete file spec

Smazání souboru. Na dotaz odpovíme názvem souboru, který chceme smazat (popř. pomocí Wilde cards, skupinu souborů). Po RETURN se budou vypisovat názvy souborů, jež odpovídají zadanému názvu. Po výpisu názvu souboru, která se má vymazat, program čeká na odpověď, zda vypsaný soubor mazat nebo nemazat. Pro vymazání souboru se musí stisknout klávesa Y a RETURN. Stisk jen klávesy RETURN daný soubor nevymaze a zobrazí následující soubor, která vyhovuje zadámu názvu. Zadáme-li při prvním dotazu za název /N (lomítko N), nebude se program ptát na každý jednotlivý soubor, který odpovídá názvu, zda-li mazat a vymaze všechny specifikované soubory najednou.

#### 5.7. E:Rename file

dotaz: What file to rename?

Mění název v adresáři disku nebo RAMdisku. Na dotaz se odpoví zadáním původního a pak nového názvu.

Např.

CAPEK38.COM,CAPEK.BAK - v adresáři D1: se přejmenuje soubor CAPEK30.COM na CAPEK.BAK

Před použitím E musí být přejmenovávaný soubor odemčen.

## 5.8. F:Lock file

dotaz: What file to lock?

Uzamykání jednoho nebo skupiny souborů. Po uzamčení souboru není možné jej smazat či přepsat název.

Např.

D2:MAN\*,\* - uzamkne všechny soubory začínající názvem MAN na disketě č.2.

## 5.9. G:Unlock file

dotaz: What file to unlock?

Odemčení uzamčeného souboru, popř. několika souborů.

## 5.10. H:New Ramtop

dotaz: New Ramtop-

Změna hodnoty Ramtop se současným přemístěním obrazové paměti. Adresu zadáváme v hex, tvaru pouze vyšší byte.

New Ramtop-C0 - nastaví Ramtop na adresu \$C000 a pod tuto adresu přemístí obrazovou paměť počítače.

Ramtop se také mění při připojování a odpojování Basicu a cartridge.

## 5.11. I:Format disk

dotazi: Format disk :1  
dotaz2: Density S/M/D/X/C :M  
dotaz3: Format D1: Y/V/N :A

Formátování disku nebo RAMdisku. Můžete použít pro podzařízení 1, 2, M, R a S.

Dotazi - volíme příslušné číslo zařízení Dx:, které chceme formátovat.

Dotaz2 - hustota formátování: S - SINGLE  
M - MEDIUM  
D - DOUBLE  
X - oboustranný formát XF 551  
C - Maže adresář, nastavuje UTOC

Dotaz3 - způsob formátování : Y - pro diskety se zrychlenou komunikací  
V - normální formátování  
N - formátování se neprovede

U RAM disku D6: je formát vždy SINGLE a u RAMdisku D8: vždy DOUBLE.

## 5.12. J:Undelete file

dotaz: Wnat file to undelete?

Pokus o obnovení smazaného souboru. Tato operace nemusí být vždy úspěšná, byl-li již proveden nějaký zápis na dané zařízení. I v případě, že příkaz bude proveden, může dojít jen k obnovení názvu v adresáři, ale soubor bude narušen popř. mohou být narušeny jiné soubory.

#### 5.13. K:Binary save

dotaz: Filename,begin,end,from,init,start

Příkaz umožňuje vyslat na definované zařízení soubor z paměťové oblasti počítače v binární formě.

Filename	- zařízení a jméno souboru s typem: D2:TEST.COM
begin	- adresa, od které bude soubor při závádění ukládán
end	- adresa konce souboru při ukládání od adresy Begin
from	- adresa, od které je daný soubor zapisován na definované zařízení
init	- inicializační adresa souboru
start	- startovací adresa

Adresy zadáváme v hexadecimálním tvaru bez znaku \$, oddělené od sebe mezerami nebo čárkami.

Např.

D8:DALI.COM,37AB,578E,4000,,37AB

Uložení souboru DALI.COM na zařízení D8: v binární formě. Soubor je v paměti uložen od adresy \$4000, při závádění do pracovního prostoru bude ukládán od adresy \$37AB po \$578E a bez inicializační procedury bude spuštěn od \$37AB.

#### 5.14. L:Binary load

dotaz: Load file?

Zavedení a spuštění programu v binárním formátu. Příkaz má volitelný parametr /N, umístěný za názvem programu, který způsobí zavedení a neodstartování programu.

#### 5.15.M:Run at Adrese

dotaz: Run at Adress?

Spuštění programu od udané adresy v hexadecimálním tvaru bez \$. Místo adresy je možné zadat znak \*. Pak bude proveden skok JMP(\$02E0). Na adrese \$02E0,1 je umístěna startovací adresa posledně spuštěného programu. Skok nemusí být úspěšný, jestliže spuštěný program uvedené adresy využívá k vlastním účelům.

(Místo adresy je možné zadat ještě T . Bude proveden teplý start Turbo-Basicu, pokud byl již předtím zaveden.

#### 5.16. N:Turbo copy

Kopírování souborů uložených na magnetofonu ve formátu Turbo 2000 nebo jiném kompatibilním formátu. Nahrané soubory nelze přímo spuštět příkazem Run at Adress. Hlavíčku kopirovaného souboru lze na obrazovce zobrazit stlačením klávesy START při čtení hlavičky souboru.

### 5.17. 0:Turbo -> file

Převod souboru ve formátu Turbo 2000 do binárního formátu a uložení do naformátovaného RAMdisku D8:. Takto uložené soubory lze přímo spouštět příkazem Binary load.

### 5.18. P:Set Printers

```
dotaz:      Parallel Printer  P1:  
              Serial Printer   P3:  
              DTR Printer     P4:  
              BT 100 Printer  P5:  
              Plotter        P2:  
  
          Type plotter    (ALFI, XY 4140  
                                ARITMA 0507): X
```

Základní nastavení lze podle potřeby měnit. Nezpomeňte, že příkaz LPRINT funguje vždy jen pro zařízení zvolené jako P1:.

---

Stručný popis povelů pro definované souřadnicové zapisovače.

- \$0A - LF (nový řádek)
- \$0C - CR (návrat pera k levému okraji)
- \$7F - Delete (posun o znak zpět)
- \$1C - Posun pera o jeden znak nahoru
- \$1D - posun pera o jeden znak dolů
- \$1E - posun pera o jeden znak doleva
- \$1F - posun pera o jeden znak doprava
- \$0F - tiskne přehlas
- \$11 - tiskne čárku
- \$15 - tiskne kroužek
- \$17 - tiskne háček

ESCAPE povely:

- \$1B,\$00 - pero nahoru
- \$1B,\$01 - pero dolů
- \$1B, cc - nastavení velikosti znaků, kde cc=\$30 až \$39
- \$1B,\$40 - inicializace grafické jednotky: nuluje souřadnice X a Y, směr zleva doprava
- \$1B,\$4E - směr psaní textu zleva doprava
- \$1B,\$4F - otočení směru písma o 90 stupňů
- \$1B,s,l,h - s může nabývat hodnot \$1C-\$1F (šipky). Povel způsobí posun pera ve směru šipky o n kroků přičemž n=h\*256+1
- \$1B,\$41,c - nastavení šířky řádku, která je ale závislá na velikosti řádku
- \$1B,\$43,c - znak umožňující tisk znaků s hodnotou stejnou jako řídící povely
- \$1B,\$70,c - rychlosť psaní znaků
- \$1B,\$72,c - rychlosť návratu pera
- \$1B,\$7A,c - rychlosť zdvihu pera
- \$1B,\$6F,c,1,h - určuje pravý okraj papíru, standardně \$0708, po překročení této hodnoty se provede CR,LF
- \$1B,\$50 xx yy - přejezd s pisátkenem nahore (PLOT)
- \$1B,\$44 xx yy - přejezd s pisátkem dole (DRAW)

Souřadnice xx a yy se skládají každá ze dvou byte a jejich ASCII hodnoty lze zjistit:

Např.: Chceme provést přejezd do bodu x=1480, y=898.  
byte 2.= 5 1480/256=5.78  
byte 1.=200 1480-(5\*256)=200  
byte 4.= 3 898/256=3.5  
byte 3.=130 898-(3\*256)=130

#### 5.19. Q:Micro monitor

Opuštění MDDOSu a start Mikro Monitoru.

#### 5.20. R:TURBO 2000

Nahrávání souborů z kazety v Turbo formátu.

#### 5.21. S:Tape Speed

dotaz: New speed (00-20):

Změna výstupní rychlosti nahrávání pro zařízení B (standardně D3:). Nastavená rychlosť je zobrazená v hlavním menu. Přijímací rychlosť se nastavuje automaticky při načítání hlavičky. 20 představuje nejvyšší rychlosť (asi 5000 baudů), standardně je nastavena rychlosť 12 (4000 baudů). Zadávaná čísla jsou v desítkové soustavě.

#### 5.22. T:Tape Mode

Cyklické přepínání jednotlivých vstupních módů pro zařízení B (standardě D3:).

První znak:

- S (short) - krátké meziblokové mezery
- L (long) - dlouhé meziblokové mezery

Druhý znak:

- S (single) - jednoduchý záznam
- D (double) - zdvojený záznam

Nejúspornější je záznam v módu SS, který však nemusí být vždy optimální. V případě, že soubor se bude při nahrávání komplikovat (např. ACTION!), doporučuje se použít mód LS. Mód XD se doporučuje používat při záznamu důležitých souborů.

#### 5.23. U:Turbo load

Nahrávání souborů, z kazety v Turbo formátu. Na rozdíl od příkazu 5.20. nelze nahrávat programy s počáteční adresou nižší než je aktuálně nastavené MEMLO. Předpokládá se, že i po zavedení programu budeme moci využívat služeb operačního systému MDDOS, proto jej nelze prepisáním zničit. Jelikož MDDOS zabírá v paměti poměrně malý prostor, je zajištěna plná kompatibilita s nejpoužívanějším systémem DOS 2.5.

#### 5.24. V:Configure

dotaz: configurate D: 12BTEM-R

dotaz: Data Bufers : 2

Změna konfigurace V/V zařízení. Po vypsání dotazu máme možnost změnit pořadí definovaných podzařízení nebo jejich vynechání (-). V řetězci je možno se pohybovat kurzorem pomocí šipek.

Máme rovněž možnost změnit počet komunikačních bufferů. Nejčastěji vystačíme s volbou 2. Tento počet určuje, že mohou být zároveň otevřeny dva komunikační kanály z níže jmenovaných zařízení.

Základní nastavení:

- 1 - disketová jednotka s přepinačem v poloze č.1
- 2 - disketová jednotka s přepinačem v poloze č.2
- B - kazetopáskový komunikační systém použity v operačním systému TTDOS
- T - starší kazetopáskový komunikační systém s 1kB bloky
- E - ROMdisk s uživatelskými programy

- M - malý RAMdisk využívající paměti pod operačním systémem
- znak pro nepoužité zařízení
- R - RAMdisk, jehož velikost je dána příslušným rozšířením počítače.

Každé definované podzařízení a buffer potřebuje ke své práci zásobník o délce 256 byte. O tuto velikost se také zvyšuje nebo snižuje MEMLO. Veškerá tato podzařízení je možno používat pouze přes komunikační zařízení D1: až D8:, neboť nejsou samostatně v tabulce V/V zařízení definována. Jejich celkový počet to neumožnuje.

#### 5.25. W:Basic Turbo load

Zavedení Basic programů do počítače v Turbo formátu. Z Basicu se do menu MDDOSu dostaneme snadno příkazem DOS. Funguje pro standardní ATARI Basic, Turbo-Basic i Basic XE. Po návratu do Basicu je program připraven k další práci nebo ke spuštění příkazem RUN. Pro návrat do Turbo-Basicu použijeme příkaz R se zadáním T.

#### 5.26. X: Basic Turbo Save

dotaz: Name of program?

Nahrání Basic programu z počítače na magnetofonový pásek v Turbo formátu. Z Basicu se do menu MDDOSu dostaneme snadno příkazem DOS. Funguje pro standardní ATARI Basic, Turbo-Basic i Basic XE. Po návratu do Basicu je program připraven k další práci nebo ke spuštění příkazem RUN.

Pro návrat do Turbo-Basicu použijeme příkaz R se zadáním T.

#### 5.27. Y:nevyužito

Z:nevyužito

### 6. Chybová hlášení operačního systému

dec.	hex.	význam
128	\$80	stlačeno tlačítka BREAK
129	\$81	pokus o otevření již otevřeného kanálu
130	\$82	neexistující zařízení (není v HATABS)
131	\$83	pokus o čtení z kanálu otevřeného pro výstup
132	\$84	chybný kód povelu pro IOCB
133	\$85	komunikační kanál není otevřen
134	\$86	chybné číslo kanálu
135	\$87	pokus o zápis do kanálu otevřeného pro čtení
136	\$88	dosaženo konce souboru (End of file)
137	\$89	záznam uříznut. Pokus o čtení delší věty, než je maximální délka pro IOCB
138	\$8A	zařízení neodpovídá (timeout)
139	\$8B	zařízení nemohlo odpovědět díky chybným parametry
140	\$8C	chyba sériové komunikace
143	\$8F	chyba v kontrolním součtu
146	\$92	požadovaný povел není v zařízení implementován
147	\$93	nedostatečná paměť počítače
160	\$A0	chybné číslo disketové jednotky
161	\$A1	otevřeno příliš mnoho souborů
162	\$A2	disk plný
163	\$A3	disk vadný
164	\$A4	číslo souboru v sektoru nesouhlasí s adresářem
165	\$A5	chybné jméno souboru
166	\$A6	chyba v počtu byte při operaci POINT
167	\$A7	pokus o zápis do uzamčeného souboru

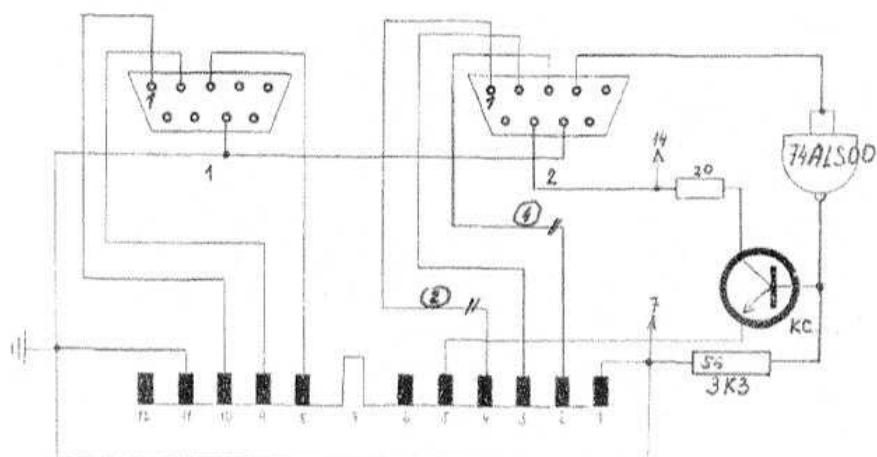
168 \$A8 neplatný povel pro komunikační handler  
169 \$A9 v adresáři disku není místo pro další soubor  
170 \$AA soubor nenašel  
171 \$AB pokus o POINT na souboru, který není otevřen pro aktualizaci  
172 \$AC chybný pokus o připojení souboru  
173 \$AD vadný sektor nebo překročení času při formátování  
174 \$AE soubor nelze obnovit

## 7. Závěr

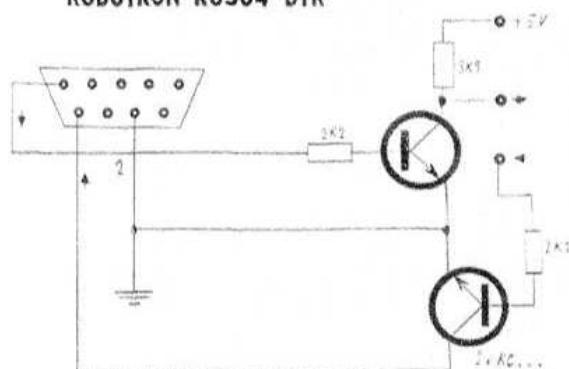
Lze říci, že programové vybavení a konstrukce nové cartridge předčí vše, co bylo do této doby na poli ATARI uděláno. Dlouhodobá práce a skloubení veškerých poznatků v oblasti počítače ATARI do jediného celku činí tuto cartridge nenahraditelnou.

8. Připojení zapísavacích jednotek k počítači

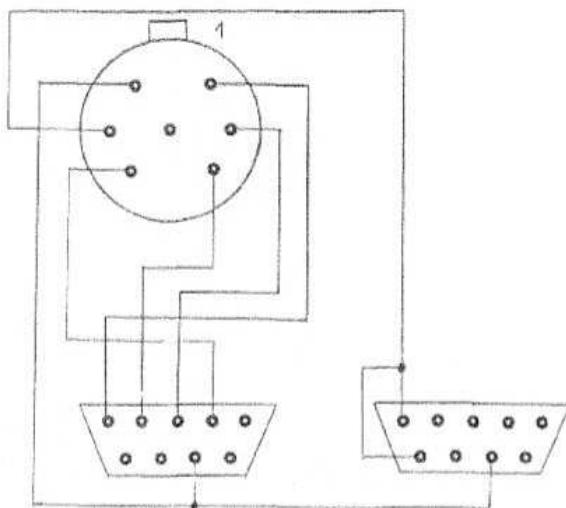
### ARITMA 0507



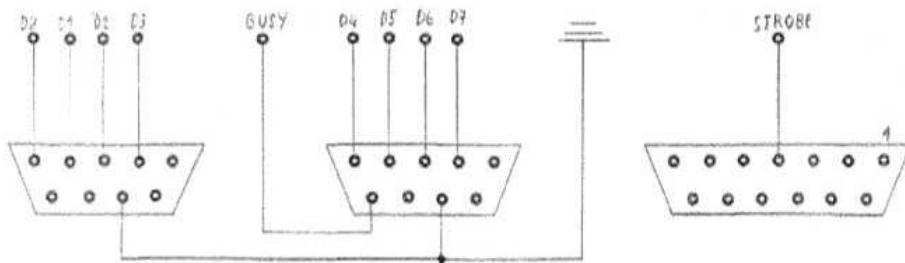
### ROBOTRON K6304 DTR



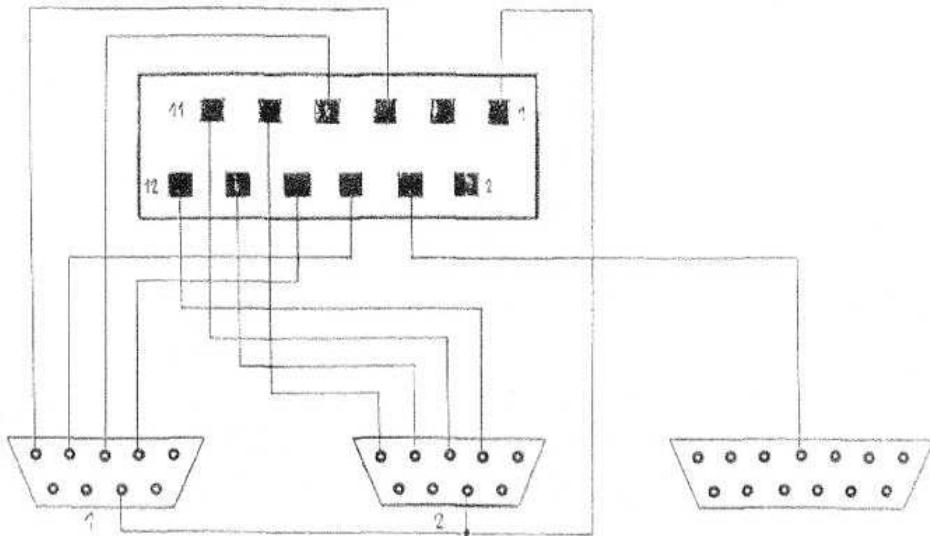
**XY 4140-50**



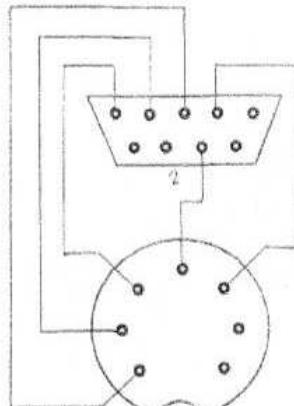
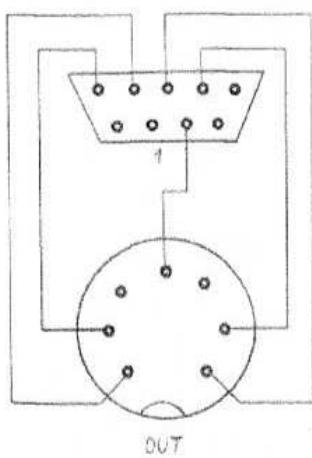
## PARALLEL CENTRONICS



# ALFIGRAF



# BT 100



**Obsah :**

1. Úvod
2. Spuštění a ovládání programu
3. Příkazy k prohlížení obsahu paměti
  - 3.1. M - hexadecimální výstup
  - 3.2. A - výstup ATASCII znaků
  - 3.3. inv.A - výstup interních kódů v ATASCII znacích
  - 3.4. @ - binární výstup
4. Příkazy pro zápis do paměti
  - 4.1. ; - vstup hexadecimální hodnoty
  - 4.2. , - vstup ATASCII znaku
  - 4.3. inv., - vstup interních ATASCII znaků
  - 4.4. > - vstup binární hodnoty
5. Práce s instrukcemi pro mikroprocesor 6502
  - 5.1. D - disassembly - výstup strojových instrukcí
  - 5.2. .adr - assembly - vstup strojových instrukcí
6. Plnění, prohledávání a přemístování obsahu paměti
  - 6.1. J - plnění obsahu paměti číslem
  - 6.2. F - vyhledávání v obsahu paměti ( Find )
  - 6.3. C - porovnání dvou oblastí paměti ( Compare )
  - 6.4. B - přemístění oblasti paměti ( Block transfer )
  - 6.5. W - přemístění oblasti paměti s přečislováním
7. Startování pracovních programů
  - 7.1. G - skok na adresu ( GoTo )
8. Ostatní obslužné příkazy
  - 8.1. K - nastavení registru PORTB ( \$D301 )
  - 8.2. ! - skok do ATARI BASICu
  - 8.3. T - kopie OS z ROM do RAM
  - 8.4. \$ - převod hexadecimální hodnoty na dekadickou
  - 8.5. # - převod dekadické hodnoty na hexadecimální
  - 8.6. + - součet dvou hexadecimálních čísel
  - 8.7. - - rozdíl dvou hexadecimálních čísel

- 8.8. Z - zapnutí/vypnutí MOTOR CONTROL
- 8.9. inv.Z - zapnutí/vypnutí COMMAND
- 8.10. U - přemístění RAMTOP
- 8.11. ESC ESC - přepnutí CARTRIGE
- 8.12. % - skok dle vektoru \$C,D
- 8.13. & - skok dle vektoru \$A,B
- 8.14. ' - skok dle vektoru \$2,3
- 8.15. ESC-SHIFT-DEL - skok na adresu \$E477
- 8.16. P - prohlížení obsahu paměti ( Wglendownica )

## 9. vstupní a výstupní operace

### 9.1. Vstup a výstup v T2000

- 9.1.1. I - natažení TURBO hlavičky
- 9.1.2. ? - adresa uložení programu
- 9.1.3. R - uložení oblasti paměti s hlavičkou T2000
- 9.1.4. H - tvorba hlavičky pro T2000
- 9.1.5. N - tvorba názvu programu pro T2000

### 9.2. Standardní vstup a výstup

- 9.2.1. L - natažení programu ( 600 baud )
- 9.2.2. S - uložení programu ( 600 baud > )

### 9.3. Příkazy CIO a STO

- 9.3.1. X - operace se vstup/výstup kanály
- 9.3.2. Y - operace se sériovým portem

## 10. Modifikace výstupu

- 10.1. [ - otevření pro výstup na zařízení < OPEN > )
- 10.2. ] - uzavření výstupu ( GLOSE )

## 11. Další možnosti cartrige M.Monitor 64kB

## 12. Obsazení paměti počítače programem M.Monitor 64kB

## 13. Souhrn instrukcí pro ovládání programu M.Monitor 64kB

## 14. Závěr

## 1. Úvod

M.Monitor 64kB je nejúčinnějším cartrigovým programem pro práci s počítači ATARI řady 800 XL, XE a 130 XE. Je neocenitelným pomocníkem nejen pro profesionály ale i pro amatéry. Jeho profesionální provedení autory Milanem Dadokem (software) a Milanem Dostálém (hardware) ho staví na úroveň firemních testovacích cartrigů. Svým provedením zabírá v paměti počítače cca 100 byte, čímž umožňuje využití celé zbylé paměti pro práci.

Program umožňuje celou řadu operací s oblastí paměti, počínaje tvorbou programů v ASSEMBLERU 6502, přes EDITACI hotových programů (např. nesmrtevnost ve hrách) a přímý přístup do paměti v reálném čase až po přímé ovládání vstup/výstupních operací, hardverových doplňků a pod.

## 2. Spuštění a ovládání programu

- a) Při vypnutém počítači - vložíme cartridge, zapneme počítač a stiskem mikrospínače aktivujeme program M.Monitor 64kB.
- \* b) Při zapnutém počítači - stiskneme a držíme tlačítko RESET zasuneme cartridge, aktivujeme stiskem mikrospínače a pustíme tlačítko RESET.

Hlavička s názvem programu, jménem autora a číslem verze nasvědčuje tomu, že program M.Monitor 64kB je aktivován a čeká na další příkazy. V případě, že se hlavička neobjeví, opakujeme aktivaci dle bodu a) nebo b).

Ovládání programu je velice jednoduché. Příkazy jsou zadávány 1 znakem, popř. znakem a adresou. Editační klávesy fungují shodně s ATARI BASICem. (šipky = směr pohybu kurzoru, mazání a vkládání znaku nebo řádku atd.)

\* Upozornění : Výrobce nedoporučuje jakoukoliv manipulaci s periferiemi - konektory při zapnutém přístroji. Nedodržení tohoto může vést ke ztrátě záruky, což u nového počítače může být nemilé překvapení.

Přesto můžeme být někdy donuceni tento způsob použít např. při záchraně vícehodinové práce nad textem ze zhrouceného textového editoru. Vždy ale na vlastní riziko.

## 3. Příkazy k prohlížení obsahu paměti

### 3.1. M – hexadecimální výstup

Syntax ; M (adresa) [RETURN] např. M E000 [RETURN]

Provede výpis obsahu 8 paměťových bytů od adresy E000). Stiskem klávesy SPACE pokračujeme ve výpisu dalších 8-mi paměťových bytů. Stiskem klávesy RETURN ukončíme výpis. Vypsáním hlášení OK je program připraven přijmout nový příkaz.

### 3.2. A - výstup ATASCII znaků

Syntax : A (adresa)[RETURN] např. A E000 [RETURN]

Provede výpis obsahu 32 paměťových bytů od adresy (E000) v podobě ATASCII znaků. Stiskem klávesy SPACE pokračujeme ve výpisu dalších 32 paměťových bytů. Stiskem klávesy RETURN ukončíme výpis. Hlášení OK umožnuje další volbu.

### 3.3. inv.A - výstup interních kódů v ATASCII znacích

Syntax : inv.A (adresa)[RETURN] např. inv.A E000 [RETURN]

Provede výpis obsahu 32 paměťových bytů od adresy (E000) v interním kódu pomocí ATASCII znaků. Stiskem klávesy SPACE pokračujeme ve výpisu dalších 32 paměťových bytů. Stiskem klávesy RETURN ukončíme výpis.

Interní kód se v počítači používá při zobrazování, v tomto kódu jsou uložena data v obrazové paměti počítače. Takto lze zobrazit na obrazovce texty, které jsou uložené v obrazové paměti počítače.

Inverzní A lze nahradit stiskem kláves **CTRL+A**.

### 3.4. @ - binární výstup

Syntax : @ (adresa)[RETURN] např. @ E000 [RETURN]

Provede výpis 1 bytu v binárním tvaru v pořadí bitů od Nejvyššího k nejnižšímu. Znak "\*" znamená, že dany bit má hodnotu 1, znak "." znamená, že daný bit má hodnotu 0. Takto lze zobrazit např. znakovou sadu, uloženou v oblasti \$E000 - \$E3FF.

Stiskem klávesy SPACE pokračujeme ve výpisu dalšího 1 bytu. Stiskem klávesy RETURN ukončíme výpis s hlášením OK.

## 4. Příkazy pro zápis do paměti

Příkazy pro zápis do paměti jsou součásti příkazů k prohlížení obsahu paměti. Po ukončení prohlížení obsahu paměti (stiskem RETURNu) lze obsah paměti přepsat najetím kurzoru na výše vyspaný obsah paměti s následným přepsáním a potvrzením RETURNem. Přepis obsahu paměti lze provádět i bez předchozího použití příkazů pro prohlížení obsahu paměti.

### 4.1. ; - vstup hexadecimální hodnoty

Syntax : ;(adresa)[RETURN] např: ;95E0 FD A2 [RETURN]

Provede zápis dvou hexadecimálně zadaných čísel (FD A2) na adresy \$95E0 a \$95E1.

### 4.2. , - vstup ATASCII znaků

Syntax : ,(adresa)[RETURN] např: ,96EA Ahoj [RETURN]

Provede zápis čtyř ATASCII znaků (Ahoj) na adresy \$96EA, \$96EB, \$96EC a \$96ED.

### 4.3. inv., - vstup interních ATASCII znaků

Syntax : inv.,(adresa) [RETURN] např.inv.,9543 Q [RETURN]

Provede zápis jednoho interního ATASCII znaku na adresu

\$9543. Interní ATASCII znaky zadávané do oblasti obrazové paměti jsou zobrazovány v nezměněné formě, což neplatí u zadávání dle bodu 4.2.

#### 4.4. > - vstup binární hodnoty

Syntax : >(adresa) [RETURN] např. >236E \*.\*.\*. [RETURN]

Provede zápis binárního čísla (bin 10011010) na adresu \$236E. Tento zápis umožnuje rychlou tvorbu vlastních znakových sad.

### 5. Práce s instrukcemi pro mikroprocesor 6502

Příkazy pro práci s instrukcemi pro mikroprocesor 6502 tvoří z programu M.Monitor 64kB silnou zbraň při editaci již hotových strojových programů, popř. při tvorbě vlastních programů ve strojovém jazyce.

#### 5.1. D - disassemblace - výstup strojových instrukcí

Syntax : D (adresa)[RETURN] např. D FE00 [RETURN]  
D'(adresa)[RETURN] např. D'FE00 [RETURN]

V prvním případě provede disassemblaci strojové instrukce od adresy \$FE00, v druhém případě provede totéž, avšak i s výpisem kódu instrukce v hexadecimálním tvaru.

Stiskem klávesy SPACE se vypíše následující instrukce, stiskem jiné klávesy se vypíše OK a lze zadat další příkaz. Neznámá instrukce se vypisuje jako ???, ostatní instrukce odpovídají standardu mikroprocesoru 6502.

#### 5.2. . - assemblace - vstup strojových instrukcí

Syntax: .(adresa)(instrukce)[RETURN]  
např. .2FE1 PLA [RETURN]

Výše uvedený příkaz uloží na adresu \$2FE1 kód instrukce PLA, tedy \$68, a vypíše adresu pro následující instrukci. Pokud se překlad nezdařil, (právě jsme vymysleli "novou" instrukci) pak M.Monitor 64kB přejde k levému okraji bez vypsání následující adresy.

Překlad instrukce probíhá postupně - nejdříve kód instrukce a pak operandy. Tohoto se například využívá při odladování programů. Např.: chceme, aby se provedla určitá část programu, která končí instrukcí JMP (adresa). Přepíšeme instrukci JMP instrukcí RTS a pomocí příkazu G (adresa) odstartujeme program. Po provedení instrukce RTS se nám program vrátí zpět do M.Monitoru 64kB (ne vždy, záleží co spuštěný program provede). Chceme-li teď napsat místo instrukce RTS povodní instrukci JMP, přepíšeme RTS instrukci JMP \$QQQQ. Protože QQQQ není číslo, přeloží se jen kód instrukce. Operand, který se instrukcí RTS nezměnil, se obnoví. Musí se napsat 4 písmena Q, protože podle počtu písmen určuje M.Monitor 64kB typ adresace.

V případě použití instrukcí s relativním operandem (adresou), se adresa uvádí v absolutním tvaru. Musí se však dodržet povolený rozsah pro relativní operand (+-\$80).

## 6. Plnění, prohledávání a přemisťování obsahu paměti

### 6.1. J - plnění obsahu paměti číslem

Syntax : J (adresa1) (adresa2) (operand) [RETURN]  
 např. J 0600 06FF 00[RETURN]

Výše uvedený příkaz uloží hodnotu 00 do oblasti paměti od adresy (\$0600) do adresy2 (\$06FF) včetně. Hodnota operandu může být \$00 - \$FF.

### 6.2. F - vyhledávání v obsahu paměti (Find)

Syntax : F (operandy) (-adresa1 adresa2[RETURN])  
 Např. F 00 0A 0B 0C [RETURN]  
 F"AHOJ" -4000 5000 [RETURN]  
 F'LDA #\$00 [RETURN]

Příkaz F umožňuje hledat v obsahu paměti 1-12 bytů. A to v podobě hexadecimálních čísel, textů nebo strojových instrukcí. Standardně se prohledává oblast od adresy \$0200 do adresy \$FFFF. Zadáme-li za hledané byty, texty nebo strojovou instrukci pomlčku (**14 pozice od začátku řádku !!**) a za ní uvedeme adresu OD a adresu DO, vymezíme oblast v které se daný operand bude vyhledávat.

Jestliže se daný operand v oblasti paměti nenachází, vypíše se OK. Je-li operand nalezen, vypíše se adresa, na které daný operand začíná. Stiskem klávesy SPACE se v hledání pokračuje, stiskem jakékoli jiné klávesy se vypíše OK a lze zadat další příkaz. Stiskneme-li klávesu SPACE a vypíše se OK, znamená to, že od poslední adresy, na které se daný operand nacházel, do konce vymezené oblasti paměti se již daný operand nenachází.

Příklady:

F 00 0A 0B 0C [RETURN] - vyhledává sekvenci 4 bytů od adresy \$0200 až po adresu \$FFFF.

F"AHOJ" [RETURN] - vyhledává sekvenci 4 ATASCII znaků od adresy \$0200 až po adresu \$FFFF.

F'LDA #\$33 -4000 45FF [RETURN] - vyhledává uvedenou strojovou instrukci mikroprocesoru 6502 v oblasti paměti od adresy \$4000 po adresu \$45FF (včetně).

### 6.3. C - porovnání dvou oblastí paměti (Compare)

Syntax : C (adresa1) (adresa2) (délka) [RETURN]  
 např. C 2000 4000 1000 [RETURN]

Příkaz provádí porovnání dvou oblastí paměti o stejně délce. Parametry (adresa1) a (adresa2) určují začátky porovnávaných oblastí a parametr (délka) určuje velikost porovnávané oblasti, v případě neshodnosti bytů v první a druhé oblasti se vypíše adresa neshodného bytu v první oblasti. Stiskem klávesy SPACE se pokračuje v porovnávání, stiskem jakékoli jiné klávesy se porovnávání ukončí a lze zadávat jiné příkazy. Stiskneme-li klávesu SPACE a porovnávání se ukončí výpisem hlášení OK, znamená to, že od poslední adresy, na které se data neshodovala, do konce porovnávané oblasti se již všechna data shodují.

#### 6.4. B - přemístění oblasti paměti (Block transfer)

Syntax : B (adresa1) (adresa2) (adresa3) (kód) [RETURN]

např. B 0600 0700 1000 [RETURN]

např. B 0600 06FF 0601 2 [RETURN]

Parametr (adresa1) určuje počátek a parametr (adresa2) určuje konec přemístované oblasti. Parametr (adresa3) určuje adresu nového počátku přemístované oblasti. Po vykonání příkazu se vypíše koncová adresa přemístěného bloku.

Např. B 0600 0700 1000 - přemístí oblast od adresy \$0600 do adresy \$0700 včetně a uloží na adresu \$1000 až \$1100 včetně a vypíše se 1100.

Použijeme-li i parametr (kód = 2), bude se přemístění provádět od zadu. Toho způsobu přemístování se používá při přemístování oblasti které se překrývají.

Např. B 0600 06FF 0601 2 - přemístí oblast paměti od adresy \$0600 po adresu \$06FF včetně a uloží od adresy \$0601 do adresy \$0700 včetně a výpisce 0700.

V případě, že nepoužijeme parametr (kód = 2), se oblast Od adresy \$0600 po adresu \$0700 vyplní bytem uloženým na adrese \$0600.

#### 6.5. M - přemístění oblasti paměti s přečislováním

Syntax : M adr1 adr2 adr3 zx adr4 adr5 [RETURN]

Např. W 0600 06FF 0700 3 0480 0800

Instrukce slouží k přemístování oblasti paměti s přečislováním 3 bytových strojových instrukcí. Parametr adr1 je počátek, parametr adr2, konec přemístované oblasti. Parametr adr3 je nový začátek přemístované oblasti.

Parametr z = mezera - přemístění zepředu.

Parametr z = 2 - přemístění ze zadu.

Parametr x = mezera - přečislování jen uvnitř přemístovaného bloku.

Parametr x = 3 - přečislování oblasti od adr4 po adr5, nejsou-li tyto adresy udány, přečisluje oblast od adresy \$0400 po adresu \$FFFF.

Např. W 0600 06FF 0700 3 0480 0800 - přemístí oblast od adresy \$0600 po adresu \$06FF na adresu \$0700 a provede přečislování od adresy \$0480 po adresu \$0800, tedy oblast kde se nachází program spolupracující s přemístovanou oblastí.

### 7. Startování pracovních programů

Při tvorbě programů je nutné, vytvořený program vyzkoušet, zda-li vykonává požadovaný úkol dle našich představ. K tomuto účelu slouží níže uvedená instrukce.

#### 7.1. G - skok na adresu (GoTo)

Syntax : G adresa aa xx yy příznaky [RETURN]

např. G 0600 0A 04 FF ..\*\*\*\*.. [RETURN]

např. G 0600 [RETURN]

např. G'F2B0 [RETURN]

Tato uvedená instrukce provede skok na zadanou adresu. Při tomto skoku se registr K viz kapitola 8.1 (stínový registr \$D301) upraví tak, aby byl vždy připojen operační systém, to znamená, že nultý bit registru K je vždy roven 1, ostatní bity zůstanou stejné. Toto platí i pro ostatní funkce, které provádějí skoky mimo M.Monitor 64kB (např. v/v operace). U instrukce G lze toto obejít tak, že zadáme G' (apostrof) (např. G'F2B0). V tomto případě se registr K nebude upravovat.

Instrukce G umožňuje nastavit před skokem všechny registry mikroprocesoru, kromě registru S.

Význam jednotlivých operandů :

adresa - adresa skoku  
aa - obsah registru A před skokem  
xx - obsah registru X před skokem  
yy - obsah registru Y před skokem  
příznaky - binární obsah registrů příznaků v pořadí:  
NV.BDIZC

Návrat do M.Monitoru 64kB je možný pomocí instrukce RTS, pokud ovšem program, do kterého skáčeme, neznamí návratovou adresu, která je uložena v zásobníku.

Vrátit se do M.Monitoru 64kB můžeme taky skokem na adresu \$0138. Jestliže má program funkci podobnou funkci G M.Monitoru 64kB, a byl-li M.Monitor 64kB už aspoň jednou nastartován a daný program nevyužívá oblast zásobníku (\$0100-\$0180), můžeme skokem na adresu \$0138 (dekadicky - 312) odstartovat M.Monitor 64kB. Např. BASIC jsme inicializovali z M.Monitoru 64kB pomocí funkce ! a chceme se vrátit do M.Monitoru 64kB bez stisku tlačítka RESET, tak provedeme instrukci X =USR(312).

Pro odlaďování programu pomocí instrukce G, lze také velmi úspěšně použít instrukci mikroprocesoru BRK.

Tato instrukce způsobí přerušení, které zpracuje M.Monitor 64kB. Po průchodu touto instrukcí se na obrazovku vypíše adresa na které je umístěna instrukce BRK, která vyvolala přerušení. Dále se vypíšou obsahy jednotlivých registrů, jednotlivé příznaky mikroprocesoru a ukazatel vrcholu zásobníku S. Celkový výpis vypadá takto :

```
PC AC XR YR NV.BDIZC SP
0400 0A 04 FF ..****.. FC
```

Stiskem klávesy SPACE se řízení počítače vrátí na adresu o 2 větší, než na jaké se nalézá instrukce BRK, s nezměněným obsahem všech registrů. Stiskem jakékoli jiné klávesy se vypíše OK a lze zadávat další příkazy.

## 8. Ostatní obslužné příkazy

### 8.1. K - nastavení registru PORTB (\$D301)

Syntax : K FE [RETURN]

Příkazem K se provede nastavení registru K, který je stínovým registrém \$D301 = PORTB. Tento registr se používá k organizaci paměti. Z principu funkce M.Monitoru 64kB vyplývá, že nemůžeme měnit hodnotu přímo na adrese \$D301, ale musíme měnit stínový registr K pomocí funkce K. Např. K FE uloží do stínového registru K hodnotu FE, která určuje, že na adresách \$C000 - \$FFFF bude RAM. Pomocí různého nastavení tohoto registru můžeme zpřístupnit různé banky ramdisku u ATARI 130XE apod. Příkaz K [RETURN] nám vypíše obsah registru K.

**8.2. ! - skok do ATARI BASICu**  
Syntax : ! [RETURN]  
      ! '[RETURN]

Příkaz provede skok do ATARI BASICu. Vykřičník s apostrofem provede skok do ATARI BASICu s novým nastavením ukazovátek BASICu, tedy provede BASIC instrukci NEW.

**8.3. T - kopie OS z ROM do RAM**  
Syntax : T [RETURN]

Příkaz provede kopii operačního systému z ROM do RAM v oblasti \$C000 - \$FFFF.

**8.4. \$ - převod hexadecimální hodnoty na dekadickou**  
Syntax : \$(číslo) [RETURN]  
      např. \$0600 [RETURN]

Tento příkaz umožňuje rychlý převod hexadecimální hodnoty na hodnoty dekadické.

**8.5. # - převod dekadické hodnoty na hexadecimální**  
Syntax : #(číslo) [RETURN]  
      např. #1536 [RETURN]

Tento příkaz umožňuje rychlý převod dekadické hodnoty na hodnoty v hexadecimálním tvaru.

**8.6. + - součet dvou hexadecimálních čísel**  
Syntax : + (číslo1) (číslo2) [RETURN]  
      např. + 0400 0500 [RETURN]

Příkaz slouží k rychlému výpočtu součtu dvou hexadecimálně zadaných čísel. Např. + 0400 0500 [RETURN], provede a vypíše 0400+0500=0900.

**8.7. - - rozdíl dvou hexadecimálních čísel**  
Syntax : - (číslo1) (číslo2) [RETURN]  
      např. - 0400 0200 [RETURN]

Příkaz slouží k rychlému výpočtu rozdílu dvou hexadecimálně zadaných čísel. Např. - 0400 0200 [RETURN], provede a vypíše 0400-0200=0200.

**8.8. Z - zapnutí/vypnutí MOTOR CONTROL**  
Syntax : Z [RETURN]

Příkaz střídavě zapíná a vypíná MOTOR CONTROL. Používá se pro přesné nastavení nahrávky nebo pro odposlech nahrávky.

**8.9. inv.Z - zapnutí/vypnutí COMMAND**  
Syntax : inv.Z [RETURN]

Příkaz střídavě zapíná a vypíná signál COMMAND. Pro urychlení zadávání lze zadat CTRL+Z. Používá se při práci s periferiemi.

#### 8.10. U - přemístění RAMTOP

Syntax : U (operand) [RETURN]  
Např. U C0 [RETURN]

Příkaz přemístí RAMTOP a tedy i obrazovou paměť, která se vždy umisťuje pod RAMTOP. Např. U C0 [RETURN]. Operand udává horní byte adresy RAMTOP. v tomto případě obrazová paměť bude umístěna v oblasti BC40-BFFF a DISPLAY LIST v oblasti BC20-BC3F. U tohoto příkazu se operand může pohybovat v rozmezí 08-C0, tedy jen v oblasti RAM. Jiný operand zhroutí počítač.

#### 8.11. ESC ESC - přepnutí CARTRIGE

Syntax : ESC ESC [RETURN]

Příkaz provede u 16 kB cartridge start druhé poloviny paměti cartridge.

#### 8.12. % - skok dle vektoru SC,D

Syntax : % [RETURN]

Příkaz provede skok podle vektoru \$C,D. Provede se instrukce, která by měla syntaxi JSR(\$000C), kdyby však existovala.

#### 8.13. & - skok dle vektoru \$A,B

Syntax : & [RETURN]

Příkaz provede skok podle vektoru \$A,B. Provede se neexistující instrukce, která by měla syntaxi JSR(\$000A).

#### 8.14. ' - skok dle vektoru \$2,3

Syntax : ' [RETURN]

Příkaz provede skok podle vektoru \$2,3. Provede se neexistující instrukce, která by měla syntaxi JSR(\$0002).

#### 8.15. ESC-SHIFT-DEL - skok na adresu \$E477

Syntax : ESC-SHIFT-DEL [RETURN]

Příkaz provede skok na adresu \$E477 a vykoná takzvaný studený start počítače.

#### 8.16. P - prohlízení obsahu paměti (Wglendownica)

Syntax : P [RETURN]

Tento příkaz patří mezi nejsilnější zbraně M.Monitoru 64kB, pokud často pracujeme s hardware (opravy doplňků nebo i samotného počítače). Umozňuje prohlízení a přepisování obsahu paměti počítače v reálném čase. Po spuštění příkazu se objeví obsah 8 stránky a kurzor na pozici 0001. Zadáním např. 06 skočí přímo na 0602 (6 stránka). Posun po celých stránkách pomocí kláves CLEAR, INSERT. Posun po jednotlivých řádcích a sloupcích pomocí směrových šipek. Nacházíme-li se v oblasti, kde lze provést přepis, můžeme přímo přepisovat hodnoty v hexadecimálním tvaru. Práce s Wglendownicou ukončíme stiskem klávesy SPACE.

## 9. Vstupní a výstupní operace

### 9.1. Vstup a výstup v T2000

#### 9.1.1 I - natažení TURBO hlavičky Syntax : I [RETURN]

Po odeslání tohoto příkazu počítač pípne a po stisku libovolné klávesy se začne nahrávat hlavička programu. Stisk klávesy BREAK po pípnutí nebo během nahrávání způsobi přerušení tohoto příkazu. Vypíše se chybové hlášení (error \$80) a OK nám oznamuje, že lze zadávat další příkaz M.Monitoru 64 kB.

Po nahrání hlavičky se tato hlavička vypíše na obrazovku v tomto tvaru :

```
xx adr1 del adr2
N název
```

Parametr xx je typový byte hlavičky. Určuje většinou typ souboru. U většiny programů má hodnotu 03. Tato hodnota znamená, že se jedná o program, který je celý nebo jeho hlavní část napsaná v assembliu.

V případě, že tento byte má hodnotu FF nebo FE, znamená to, že se jedná o datový soubor, který byl vytvořen programem BASIC DOS nebo TURBO BASIC s BASIC DOS (jestliže je řídicí byte FE, po nahrání pomocí téhoto programu se provede instrukce RUN). Je-li typový byte 04, jedná se o tzv. binární soubor - používaný hlavně spolu se zaváděcím ZXL umožňujícím jednodušší převod programů z disketové jednotky na magnetofon. Ostatní hodnoty v typovém bytu mohou znamenat vždy něco jiného, neboť jejich typový byte je 04, FE a FF, žádný význam.

Parametr adr1 znamená ukládací adresu, tj. adresu od které se má začít ukládat daný program. Parametr adr2 určuje startovací adresu. Tyto parametry nemají u nahrávek v TURBU, jejichž typový byte je 04, FE a FF, žádný význam.

Parametr del určuje délku programu. Má význam u všech nahrávek, kromě tzv. 1kB bloků - zařízení "D:", "T:", "Q:".

Na druhý řádek se vypisuje název programu. Samotný název začíná až od třetí pozice. První dva znaky ( N a mezera ) určují, že se jedná o název - N je příkaz pro zadání názvu nahrávky viz níže.

Na třetím řádku se objeví ? (kurzor). Viz. příkaz níže.

#### 9.1.2. ? - adresa uložení programu Syntax : ?(adresa) [RETURN]

Příkaz slouží pro zadání adresy pro uložení programu natahovaného přes příkaz I. Zadávaná adresa nemusí být stejná jako v hlavičce (parametr adr1), většinou se však zadává stejná. Po odeslání tohoto příkazu počítač pípne a po stisku libovolné klávesy začne nahrávat samotný program. Stisk klávesy BREAK po pípnutí nebo během nahrávání způsobi přerušení tohoto příkazu - vypíše se error \$80 a OK - lze zadávat další příkazy M.Monitoru 64kB. V případě úspěšného nahrání programu se vypíše jen OK

**POZOR!!** Během nahrávání i vysílání v TURBU je registr K  
(stínový registr \$D301) vždy \$FE !!

### 9.1.3. R - uložení oblasti paměti s hlavičkou T2000

Syntax : R [RETURN]

Příkaz R uloží na kazetu TURBO nahrávku oblasti paměti počítače podle hlavičky (příkaz H - viz níže). Tato hlavička se vytiskne a počítač dvakrát pípne. Stiskem libovolné klávesy začne vysílání. Stiskem klávesy BREAK po pípnutí se vypíše OK a protože vytisknutá hlavička začiná písmeny H a N, které jsou příkazy pro změnu hlavičky (viz 9.1.4.), můžeme přepsáním zobrazěnou hlavičku a potvrzením klávesou RETURN změnit obsah hlavičky. Najednou však lze měnit buď jen název nebo adresy + parametry.

### 9.1.4. H - tvorba hlavičky pro T2000

Syntax : H rr adr3 xx-adr1 del adr2 [RETURN]  
např. H 30 2000 03-4000 0dff 40fe [RETURN]

Příkaz H umožňuje tvorbu hlavičky pro nahrávku T2000.  
Syntax tohoto příkazu je následující :

Parametry xx,adr1,del a adr2 mají tentýž význam jako u příkazu I. Parametr adr3 znamená adresu, od které je uložen daný program nebo blok dat v paměti počítače (parametr adr1 znamená odkud by se měl daný program uložit, aby fungoval, a adr3 odkud je uložen před záznamem na pásek - tyto parametry mohou ale nemusí být stejné).

Parametr rr určuje rychlosť prenosu. M.Monitor &4kB má zabudované tzv. UNITURBO, které může komunikovat s magnetofonem od standardní rychlosti až po rychlosť 6400 baudů. Parametr rr může mít hodnotu v rozsahu \$11-\$30, přičemž rychlosť \$29-\$2F se nepoužívají. Standardní rychlosť pro T2000 je \$30 - nastaví se vždy při startu přes RESET. Výsledná průměrná rychlosť v Bd se vypočítá podle vzorce v=INT(109000/rr).

Tabulka rychlosťí :

rr	v	rr	v								
11	6411	16	4954	1B	4037	20	3406	25	2945		
12	6055	17	4739	1C	3892	21	3303	26	2868		
13	5736	18	4541	1D	3758	22	3205	27	2794		
14	5450	19	4360	1E	3633	23	3114	28	2725		
15	5190	1A	4192	1F	3516	24	3027	30	2270		

### 9.1.5. N - tvorba názvu programu pro T2000

Syntax : N (název) [RETURN]  
např. N Monitor [RETURN]

Příkaz umožňuje vytvořit nebo změnit název programu nebo datového souboru. U rychlosťí rr=\$30 je povolená délka názvu maximálně 10 znaků, u ostatních rychlosťí maximálně 20 znaků. Např. N ZALOHA.DAT [RETURN].

## 9.2. Standardní vstup a výstup

### 9.2.1. L - natažení programu (LOAD)

Syntax : L adří del zařízení  
např. L 1000 0300 C:

Příkaz L umožňuje nahrávání programu nebo datových souborů z definovaného zařízení. Parametr adří udává adresu odkud se budou data nahrávat, parametr del udává maximální délku vstupního souboru, kterou lze nahrát. Parametr zařízení určuje zařízení s uloženým souborem, např. C:, D:, D8: atd.

Po ukončení nahrávání se na obrazovku vypíše skutečná délka nahraného souboru nebo dat. Udáte-li větší délku del, než je skutečná délka nahrávky, počítač vyhlásí ERROR \$88, nejedná se však o chybu, protože bylo dosaženo konce dat.

Např. L 1000 0300 C:

Uvedený příklad nahraje program nebo soubor dat a uloží ho od adresy \$1000 v délce \$0300. Vstupním zařízením je magnetofon.

Zadáme-li L [RETURN] bez uvedení parametru, vypíše se délka poslední nahrávky.

### 9.2.2. S - uložení programu

Syntax : S adří del zařízení [RETURN]  
např. S 1000 0300 C: [RETURN]

Příkaz ukládá data na definované zařízení. Syntaxe pro jednotlivé parametry je stejná jak u příkazu L. Parametr adří udává adresu, kde začínají data, parametr del udává délku dat. Např. S 1000 0300 C: - uloží na magnetofon standardní rychlosť (600 bandu) data začínající na adrese \$1000 o délce \$0300 byteů.

V případě, že tyto příkazy použijeme ke kopírování, zadáme u příkazu L parametr del maximální (např. FFFF), a u příkazu S zadáme za parametr del skutečnou délku dat, která se vypsala za ERROR \$88.

Jestliže chceme data připojit na konec již existujícího souboru např. na disketu - chceme provést tzv. APPEND - zadáme S s apostrofem ('S'). Např. S' 1000 0300 D:AUTORUN.SYS - data se připojí na konec souboru s názvem AUTORUN.SYS uloženém na disketu. Oba tyto příkazy lze používat se všemi systémy, které byly na ATARI postupně vytvářeny. Od různých diskových operačních systémů (DOS 2.0, 2.5, 3.0, BIBODOS, SMART DOS) až po různé kazetové O.S. (TOS, DTURBO, T-EMO). Stačí pouze abychom příslušný OS zavedli do počítače a po startu M.Monitoru 64 kB (viz kapitola 2.) systém "oživili" jeho vlastním RESET vektorem (viz kapitoly 8.12.-8.14.). Většinou stačí zadat příkaz % [RETURN] a pokud se objeví OK můžeme s tímto zařízením pracovat.

## 9.3. Příkazy CIO a SIO

### 9.3.1. X - operace se vstup/výstup kanály

Syntax : X c pr adří del x1 x2 název [RETURN]  
např. X 1 03 ??? XXXX 04 00 C: [RETURN]

Příkaz X je značně složitý, avšak umožňuje provádět spoustu různých operací se V/V zařízením.

Parametr c určuje číslo kanálu, s kterým se bude

provádět CIO operace. Může nabývat hodnot 0 až 7. Parametr pr určuje druh operace, která se bude provádět. Tyto dva parametry jsou povinné, ostatní se mohou vynechat. Např. instrukce v BASICu CLOSE#1 se provede následovně :

X 1 0C [RETURN]

Pro lepší pochopení celého příkazu X je nutno uvést, jak a kam se který parametr ukládá. Parametr pr se ukládá na druhý byte bloku IOCDB, tedy na adresu \$0342 je-li parametr c=0, na adresu \$0352 je-li parametr c=1. Parametr pr nabývá těchto hodnot s těmito významy :

HODNOTA	VÝZNAM
03	OPEN
05	INPUT
07	GET
09	PRINT
0B	PUT
0C	CLOSE
0D	STATUS
21	DELETE
23	LOCK
FE	FORMAT
atd.	

Parametr adri se ukládá na adresu \$0344,5 (pro c=0) a určuje adresu od kde se čtou nebo kam se ukládají data při V/V operaci. Napišeme-li místo čísel na pozici parametru adri otazníky (????), dosadí se na adresy \$0344,5 adresu, na které začíná text parametru název. Při příkazu OPEN lze napsat místo adri ???? a na pozici, kde začíná název napsat zařízení - viz příklady.

Parametr del se ukládá na adresu \$0348,9 ( pro c=0 ) a určuje délku dat, které se budou přijímat nebo vysílat při V/V operaci (obdobně jako u příkladu L).

Parametr xi se ukládá na adresu \$034A (pro c=0)- ICAX1.

Používá se při OPEN pro specifikaci, zda jde o čtení, zápis atd. Může nabývat těchto hodnot :

HODNOTA	VÝZNAM
04	čtení
06	čtení adresáře - DIR
07	speciál DIR
08	zápis
09	zápis - připojení (APPEND)
0C	čtení i zápis
atd.	

Parametr x2 se ukládá na adresu \$034B (pro c=0)- ICAX2. Tento parametr má u každého zařízení jiný význam, většinou se rovná nule. Jestliže se v něm u zařízení C: nachází \$80, pak pauza mezi bloky bude krátká.

Jestliže některý nepovinný parametr nezadáme (napišeme místo něho např. XXXX ), adresy na které se má tento parametr uložit se nezmění.

**Příklady :**

<b>BASIC, TBASIC</b>	<b>M.MONITOR 64 KB</b>
OPEN #1,4,0,"C:"	X 1 03 ???? XXXX 04 00 C:
BPUT #1,\$4000,\$0100	X 1 0B 4000 0100
BGET #1,\$3000,\$8000	X 1 07 3000 8000
PRINT #6;"AHOJ"	X 6 09 ???? 0004 XX XX AHOJ
CLOSE #2	X 2 0C
XIO 254,#1,0,0,"D8:*.**"	X 1 FE ???? XXXX 00 00 D8:*.**
atd.	

Příkaz X lze taky použít k vytváření binárních souborů. Binární soubor rozpoznáme tak, že musí mít na začátku FF FF. V další části binárního souboru jsou bloky dat. Pro vytvoření binárního souboru musíme nejdříve otevřít kanál na nějaké zařízení, které dovoluje posílat data po částech např. ramdisk, disketa, 1kB bloky TURBO apod., ne vsak C: s krátkými mezerami. Vytvoření binárního souboru pak provedeme např. :

X 1 03 ???? 0000 08 00 D:AUTOROM.SYS  
Pro vyslání dvou bytů FF použijeme příkaz :  
X'1 Z

Každý blok dat má svou hlavičku tvořenou počáteční adresou, kam se blok ukládá, a adresou, kam se uloží poslední byte bloku. Např. 41 00 41 00 00 - první čtyři byty jsou hlavičkou bloku, v tomto případě dlouhého 1 byte. K vytvoření bloku s hlavičkou se používá příkaz :

X'1 adr1 adr2 adr3

Parametr adr1 a adr2 určuje oblast, kam se dany blok Dat má uložit - je to vlastně hlavička bloku. Parametr adr3 určuje, na které adrese se data ze kterých se blok skládá, nacházejí (většinou adr1 = adr3).

Pro odstartování binárního souboru se musí startovací adresa uložit na \$02E0,1. Přes tento vektor se provede skok po dosazení konce binárního souboru. Existuje tzv. inicializační vektor \$02E2,3, přes který se skáče vždy po přečtení jednoho bloku dat. Nastavit tyto vektory lze pomocí příkazu :

X' 1 R star init

Lze zadat jen jednu adresu nebo obě najednou.

**9.3.2. Y - operace se sériovým portem**  
Syntax : Y de pr st adr1 delk sec ti

Příkaz Y umožnuje provádět základní V/V operace se Sériovým portem. Složitost příkazu je obdobná příkazu X. K významu jednotlivých parametrů lze uvést :

PARAMETR	ADRESA	VÝZNAM
de	\$0300	číslo přístroje (\$0301=1 vždy)
pr	\$0302	příkaz \$52 - čtení bloku
st	\$0303	\$40 - data vstup \$80 - data výstup \$00 - žádné data
adr1	\$0304,5	adresa buferu
delk	\$0308,9	délka dat
sec	\$030A,B	číslo sektoru (u diskety)

Bližší informace o tom jak využít příkazy X a Y najdete např. v PEEK - POKE I, Zpravodaj 4/88 PRAHA Interfacing apod.

**10. Módifixace výstupu**

**10.1. [ - otevření pro výstup na zařízení**  
Syntax : [ zařízení [RETURN]

Příkaz umožnuje výstup jednotlivých výše uvedených příkazů na specifikované zařízení především na tiskárnu.

Není-li zadán parametr zařízení, bude se tisknout na paralelní port - CENTRONICS - kde PORTA = DATA, COMMAND = = STROBE a STRIG1 = BUSY. Zadáme-li zařízení (např. [P;]) otevře se kanál 0, přes který se provádí tisk, na toto zařízení. V případě chyby při pokusu o otevření zařízení se tato chyba výpisem a výstup bude nastaven na obrazovku (S:).

Na tiskárnu se tiskne vše, včetně klaviček. Na tiskárně se tiskne nepřetržitě, přerušení tisku se provede stiskem klávesy ESC nebo klávesou BREAK, což však způsobí chybové hlášení.

U některých příkazů je změněna funkce. U příkazu M se tiskne na jeden řádek \$10 bytů, u příkazu A se tiskne na řádek \$40 znaků a znaky, které neumí tiskárna vytisknout (\$00-\$1F,\$7E-\$7F), se tisknou jako tečka (.), inverzní znaky se tisknou jako normální. U příkazu D se standardně (při volbě bez apostrofu) tisknou i kódy instrukcí. Výpis těchto kódů se vypne volbou s apostrofem.

Pokud máte v počítači program definující jiné zařízení pro výpis a provedete jeho inicializaci (přes příkaz G init adresa, tím se dostane do HATABS), můžete pro výpis použít i toto zařízení (viz kapitola 9.2.).

**10.2. ] - uzavření pro výstup**  
Syntax : ] [RETURN]

Příkaz provede uzavření tisku na otevřené zařízení a nastaví výstup na obrazovku (S:).

**11. Další možnosti cartridge M.Monitor 64kB**

Mezi další možnosti cartridge M.Monitor 64kB patří start uvedené cartridge přes tlačítko RESET s podržením tlačítek OPTION, SELECT a START.

**RESET - OPTION** = smaže paměť od adresy \$0000 do adresy \$BFFF, na adresu \$03F8 uloží 1, to značí - odpojí po resetu BASIC.

**RESET - SELECT** = zachová oblast paměti od adresy \$0000 do adresy \$03FF (mimo adresy \$01FE a \$01FF), kterou uloží pod Systém od adresy \$D800. Použijeme příkazy K FE a pak M D800 [RETURN]. Tato kombinace umožňuje nahlédnout např. do HATABS tabulky před RESETEM.

**RESET - START** = obrazovka zůstává celá černá. Program čeká na byte, který určí RAMTOP, tedy kam uloží obrazovou paměť. Standardně ukládá na \$9C20 DL, obraz na \$9C40 v GR.0. Toho lze využít v případě, že v oblasti od adresy \$9C20 po adresu \$FFFF máme nějaký program nebo datový soubor. Např. chceme mit obrazovou paměť pod adresou \$C000 - zadáme C0. S tímto bodem souvisí kapitola 8.10.

#### 11. Obsazení paměti počítače programem M.Monitor 64kB

\$0000-\$007F	využívá OS a M.Monitor 64kB
\$0080-\$00D1	využívá BASIC (pokud není připojen lze použít)
\$00CB-\$00D1	nejšou používané
\$0100-\$01FF	zásobník mikroprocesoru - využívá i M.Monitor
\$0200-\$03FF	pro různé účely OS viz PEEK, POKE
\$0400-\$047F	buffer magnetofonu pro rychlosť 600 Bd. Pokud nebude použit (C:), lze využít
\$0480-\$057D	volná paměť
\$057E-\$05FF	využity pro operace FP (jinak lze použít)
\$0600-\$9C1F	volná paměť
\$9C20-\$9FFF	obrazová paměť BASICu a standardně i M.Monitoru 64kB nebo volná paměť
\$A000-\$BFFF	ROM BASIC nebo RAM
\$C000-\$CFFF	operační systém
\$D000-\$D7FF	V/V obvody
\$E000-\$FFFF	operační systém

#### 13. Souhrn instrukcí pro ovládání programu M-Monitor 64kB

M	A	inv.A	@	;adr
,adr	inv.,adr	>adr	D	.adr
J	F	C	B	W
G	K	!	T	-
\$	#	+		
Z	inv.Z	U	ESC	ESC
%	&	'	ESC	-SHIFT-DEL
P	I	?	R	
H	N	L	S	
X	Y	[		]

#### 14. ZÁVĚR

Cartrigová verze M.Monitoru 64kB je šířena ve dvou základních verzích se spoustou různých modifikací, které se netýkají funkčnosti samotného programu M.Monitor 64kB.

První základní verze je 8kB cartridge, obsahující jen samotný program M.Monitor 64kB u něhož zůstává bez významu příkaz ESC ESC viz. 8.11.

Druhou základní verzi je 16kB cartridge, obsahující ve "spodních" 8kB program M.Monitor 64kB a v "horních" 8kB obsahuje celou řadu různých užitkových programů ( např. 7

druhů zaváděcích hlaviček, systémy pro práci s periferiemi -  
- BT100, Labora PLOTR, centronics atd.). Přechod mezi  
"spodní" a "horní" části 16kB cartridge se uskutečňuje  
příkazem ESC ESC viz. 8.11.

Tento popis se týká poslední verze programu  
M.Monitor 64kB, která pracuje výhradně na cartridge v zapojení  
kompatibilním s firmou OSS. Starší verze nemusí obsahovat  
všechny zde popsané příkazy.

TURBO Tape Disk Plotr / MD DOS \  
(c) 26.09.91 by M.Dadok

D:12BTEM-- Memory: 64k Tape speed: 12  
MemIO: 174A Ramtop: A8 Tape mode: 55

A:Disk Directory	N:Turbo copy
B:Basic ^B:ON/OFF	O:Turbo->File
C:Copy file	P:Set Printers
D:Delete file	Q:Micro Monitor
E:Rename file	R:TURBO 2000
F:Lock file	S:Tape speed
G:Unlock file	T:Tape Mode
H:New Ramtop	U:Turbo load
I:Format Disk	V:Configurate
J:Undelete file	W:Basic Turbo Load
K:Binary save	X:Basic Turbo Save
L:Binary load	Y:Type file
M:Run at address	Z:

### Command:

2524B0A4F5241073F3F3F0A3F3F3F0A  
3F3F3F0A4F52410441534C043F3F3F0A  
5048500A4F52410041534C0A3F3F3F0A  
3F3F3F0A4F52410141534C013F3F3F0A  
42504C094E5241083F3F3F0A3F3F3F0A  
3F3F3F0A4F52410541534C053F3F3F0A  
434C430A4F5241033F3F3F0A3F3F3F0A  
3F3F3F0A4F52410241534C023F3F3F0A  
4A535201414E44073F3F3F0A3F3F3F0A  
42495404414E4404524F4C043F3F3F0A  
504C500A414E4400524F4C0A3F3F3F0A  
42495401414E4401524F4C013F3F3F0A  
424D4909414E44083F3F3F0A3F3F3F0A  
3F3F3F0A414E4405524F4C053F3F3F0A  
5345430A414E44033F3F3F0A3F3F3F0A  
3F3F3F00414E4402524F4C023F3F3F0A

BRK\ORA\??? \??? \??? \ORA\ASL\???

9399 48950