

7-8/1988

KLUB  
MIKROELEKTRONIKY



**ATARI**®

Z P R A V O D A J

O L O M O U C

S U P E R M O N V 2-Ø - 2-1  
-----

Rožnov pod Radhoštěm  
Svazarmovská 1574

Petr Vičar  
10.12.1987

## 1. Úvod

Program slouží především ke kopírování a úpravám nahrávek z magnetických pásek (kazet) různého formátu. Hlavním důvodem pro jeho vytvoření byla potřeba převádět programy standartního ATARI formátu (dvoutónová modulace, přenosová rychlost 6ØØ-8ØØ Bd) do formátu ZX - tedy stejného formátu jako u mikropočítače ZX Spectrum. Zavést do paměti a spustit takovéto programy lze pouze pomocí zaváděče ZXI, některé z nich rovněž zaváděčem Turbo 2ØØØ. Omezení se týká přenosové rychlosti a typu souboru (kódu) jak bude uvedeno dále v textu.

Vzhledem ke značnému rozšíření programu SUPERTURBOSAVE jsou příkazy SUPERMONU voleny tak, aby jejich syntaxe byla stejná nebo alespoň podobná. Dalším hlediskem bylo co nejmenší obsazení paměti, při co největším komfortu.

Z toho vyplynul např. příkaz GOTO, který vlastně umožňuje mít další libovolné příkazy např. na kazetě a v případě potřeby je načíst a spustit. Další úspory paměti je dosaženo použitím pouze 4 zobrazovaných řádků, což se v praxi ukázalo jako dostatečující a přitom ušetří 3/4 kbytu. Protože programy prochází vždy řadou úprav, je důležité znát s jakou verzí právě pracujeme. Každá jiná verze se může od tohoto popisu lišit. Tento popis platí pro verzi V2.Ø s uvedením některých odlišností u verze V2.1.

## 2. Všeobecný popis

Program je v absolutním kódu a zavádí se do počítače po stisknutí kláves OPTION+START. Existuje rovněž verze ve formátu ZX.

Veškeré údaje v popisu, při vstupu přes klávesnici a zobrazení na obrazovce televizoru jsou ve formátu hexadecimálním. Proto není třeba uvádět před číslem znak \$ (dolar). Podobná konvence je zavedena v ATMASu. Hexadecimální formát je pro 8-bitové mikropočítače mnohem vhodnější než běžný dekadický

z důvodů, které vyplynou z následujících příkladů.

### Popis programu SUPERMON

Př.1: zadání startovací adresy

#### A. Dekadicky

- pomocí monitoru přečteme adresu ve dvou bytech 125;12
- výsledná adresa se vypočítá:  $12 * 256 + 125 = 3072$

#### B. Hexadecimálně

- monitorem přečteme 2 byty 7D;0C
- výsledná adresa je 0C7D

Př.2: ještě lépe je vidět výhodu hexadecimálních čísel na opačném převodu, tj. při potřebě uložit číslo větší než 255 do paměti.

#### A. Dekadicky (postup běžný při použití kalkulačky)

- číslo 8253 se rozloží do dvou bytu takto:

$$8253 : 256 = 32,2\dots$$

$$32 * 256 = 8192$$

$$8253 - 8192 = 61$$

- do paměti se uloží dvě čísla v pořadí 61;32

#### B. Hexadecimálně

- číslo 203D se zapíše do paměti jako dvě čísla v pořadí 3D;20

Neplatné nuly není nutno při zápisu uvádět. Místo 000C stačí psát C, místo 02 jenom 2 apod.

### 2.1. Obsazení adresního prostoru

Platí při použití SUPERMONU pro ATARI 800

0000 - 03FF	pracovní oblast operačního systému
0400 - 047F	bufer magnetofonu
0480 - 0DFF	vlastní kód SUPERMONu + obrazová paměť
0E00 - BFFF	(44 kbytu) volná oblast pro data (kopírované nebo upravované programy)
C000 - FFF9	operační systém (ROM)
FFFA - FFFF	vektory přerušování (IRQ, NMI, RESET) těchto 6 bytů se nesmí přepsat

Pozn. V některých případech lze pro ukládání dat použít

i v oblasti C000 - FFFA, tedy dalších 16 kbytu s výjimkou adresní oblasti periferních obvodů tj. D000 - D800

### 3. Příkazy SUPERMONU

Všechny příkazy jsou jednoznakové, případné zadání parametrů si program vyžádá dotazem. Kteroukoliv činnost je možno přerušit stisknutím klávesy RESET. Program se vrátí do stavu, kdy očekává vstup příkazu.

Při každém stisknutí NOTE RESETu v y m a ž e automaticky operační systém oblast BC20 - BFFF (nastavuje zde DL programova obrazovou paměť. Tuto nepříjemnou vlastnost systému se zatím nepodařilo obelstít. Proto je třeba užívat RESET uvážlivě, zejména, máme-li již zavedena v RAM užitečná data.

Zadání hexadecimálních čísel lze ukončit obvyklým stisknutím klávesy (RETURN). Případný další text se potom začne vypisovat na novém řádku. V případě, že místo (RETURN) stiskneme jinou libovolnou klávesu, která nepatří mezi hexadecimální číslice (0123456789ABCDEF) zadání čísla se rovněž ukončí, ale další text se vypíše bezprostředně za napsaný znak.

Této vlastnosti lze s úspěchem využít pro zobrazení většího počtu informací na zmenšené obrazovce. Opravovat lze pouze právě zadávané údaje a to vymazáním zadaného znaku pomocí klávesy DELETE BACK SPACE. Použití šipek pro pohyb kurzoru nelze použít.

#### 3.1 Přehled příkazů:

- B - BLOCKTRANSFER - přemístění bloku dat v RAM
- C - COMPARE - porovnání souboru na kazetě s obsahem RAM
- D - DOUBLE - opakování příkazu SAVE
- F - FASTTURBO - čtení formátu FASTTURBO z kazety
- G - GOTO - skok na zavedenou adresu
- I - ZX-LOAD - čtení z kazety ve formátu ZX
- L - LOAD - čtení z kazety ve formátu ATARI
- M - MONITOR - čtení a změna obsahu RAM
- N - NAME - změna jména programu (jen V 2.1)
- R - opakování příkazu ZX-SAVE
- S - SAVE - zápis na kazetu ve formátu ATARI 600 Bd



### 3.3 C - COMPARE

Příkaz se používá ke komparaci (porovnání) přečteného nebo zapsaného programu ve formátu ZX. Protože zabezpečení celé nahrávky jedním kontrolním součtem je z hlediska možných chyb nedokonalé, je dobré mít možnost nahrávku při kopírování zkontrolovat.

Toto oceníme zejména u programů, které potřebujeme mít bezpečně zkopírované bez chyb. Chyba se totiž nemusí projevit ihned po spuštění programu.

Příkaz COMPARE porovná byt po bytu obsah paměti RAM s obsahem nahrávky a v případě nesouhlasu okamžitě hlásí chybu a místo, na kterém ji zjistil. Pokud se při opakování komparace ohlásí chyba na stejném místě, jde o "tvrdou chybu" při zápisu nebo čtení, pokud se chyba projevuje na různých místech, pak půjde spíše o vadu magnetofonu nebo nahrávku na špatném materiálu apod.

Postup při komparaci je jednoduchý.

#### Čtení z kazety:

Po přečtení programu z kazety do RAM (příkazem I) vrátíme pásek zpět na začátek před hlavičku, zmáčkne na magnetofonu tlačítko PLAY a stiskneme klávesu C. Na obrazovce blikne pouze hlavička a je-li program v pořádku, objeví se na konci hlavička a délka zkomparované nahrávky.

#### Zápis na kazetu:

Po zapsání dat z RAM na kazetu (příkazem T nebo R) opět vrátíme pásek zpět před zapsanou hlavičku a postupujeme stejně jako je uvedeno výše při kontrole čtení.

#### NOTE

Mezi příkazem I (R,T) a komparací C nesmíme stisknout RESET, neboť se vymaže hlavička. To neplatí u verze SUPERMONU V2.1 a vyšší. U některých magnetofonů může při zapnutí motoru při zápisu vzniknout klik, který za určitých okolností je při vstupu rozpoznán jako hlavička. To se týká jak příkazu C, tak příkazu I. V tom případě je nutné najít hlavičku a pásek vrátit cca o 1 cm zpět a potom teprve použít příkaz C nebo I.

### 3.4 D - opakování SAVE

Příkaz D zopakuje poslední provedenou operaci SAVE. Při více kopiích tak není nutné opakovat zadání počáteční adresy a délky přenosu.

### 3.5 F - FASTTURBO

Tento příkaz se používá při úpravách speciálních formátů záznamu jako jsou např. kilobytové bloky nebo formát fastturbo. Příkaz F umožní sejmout z kazety zadaný počet bytů bez ohledu na formát. Z kazety se přečtou a uloží od zadané adresy v RAM všechny byty včetně řídicích. Takto lze číst z kazety jednotlivé standartní bloky zadáním délky 83 (tj. dekadicky 131), tedy tři úvodní řídicí byty + 128 vlastních datových bytů. Kontrolní součet se do počtu přenášených bytů nepočítá.

Po zadání příkazu F čte operační systém data z kazety okamžitě bez jakékoliv časové prodlevy. K tomu je však nutné předem otevřít příslušný vstupní kanál. Toho lze dosáhnout např. příkazem LOAD. Pokud LOAD nepotřebujeme, lze ho jen vyvolat a klávesou BREAK zrušit. Z tohoto důvodu nebylo nutné zavádět zvláštní příkaz OPEN, který by otvíral vstupní kanál.

Použití příkazu F předpokládá dobrou znalost formátu dat na kazetě. Zadávají se dva parametry:

od adr: adresa, od které se začnou ukládat data čtená z kazety  
délka: počet bytů, které budou přečteny.

### 3.6 G - GOTO

Příkaz GOTO slouží k odstartování programu, který je uložen v RAM. Pokud tento program končí instrukcí RTS, vrátí se řízení do SUPERMONu. Jiný způsob návratu je pomocí klávesy RESET. V tomto případě však spuštěný program nesmí přepisovat důležité systémové adresy v nulté stránce RAM, zejména adresu RESETU (adresy C,D) a stavové slovo RESETU (adresa 9).

Příkaz GOTO umožňuje mít na kazetě vlastně další příkazy SUPERMONu, které se používají jen občas a zbytečně by zabíraly paměť RAM.

Dalším využitím je možnost okamžitého odstartování upravova-

ného programu bez opětovného čtení z kazety. Je třeba ovšem d' na to, aby program byl uložen přesně na adresách, na kterých při zavedení být. K tomuto účelu lze použít příkaz BLOCKTRAN<sup>2</sup>.

Parametr je jeden a udává se jím startovací adresa:

Pr. GOTO: A200

### 3.7 I - ZX LOAD

Příkaz I načítá data z magnetické pásky ve formátu Z' paměti RAM. Formát ZX má v podstatě dva bloky: hlavičku a vlastní data. V hlavičce jsou uloženy následující údaje.

- 1.byte - rozlišení hlavičky od bloku dat (hlavička - data - FF)
- 2.byte - typ souboru (kód). Zatím se používají dvě typy:
  - 03 - absolutní program (jeden logický blok)
  - 04 - relativní program (několik logických bloků)
- 3. až 12.byte - název programu (maximálně deset libovolných znaků)
- 13.-14.byte - zaváděcí adresa
- 14.-15.byte - délka datového souboru (počet bytů)
- 16.-17.byte - startovací adresa
- 18.byte - kontrolní součet

Datový blok obsahuje 1. rozlišovací byte (hodnota) vlastní data a jeden byte jako kontrolní součet.

Před vlastním čtením je zapotřebí předem nastavit rychlost, pokud není standardní (tj. 1). Bližší popis je u příkazu V.

Po zadání ZX LOAD hledá program na pásku hlavičku. Jakmile ji nalezne, tak ji zobrazí a zastaví posuv magnetofonu. Zobrazí se všechny informace z hlavičky v pořadí:

Kód, název, zaváděcí adresa, délka, startovací adresa

Poté se vypíše otázka "Od adr" a očekává se vstupní adresa v RAM, od které mají být ukládána čtená data. Nejbližší povolenná adresa je E00; maximální použitelná adresa je BFFu verze V2.0. Z toho vyplývá, že nejdelší program může mít max. délku B200.

tudíž načíst souvisle programy až do délky C200 tj. po adresu CFFF.

Po zadání adresy proběhne čtení datového bloku a výpis počtu čtených bytů. Tento údaj je o 1 byte větší než údaj v hlavičce, protože je do čtených dat započítán i kontrolní součet. Platný je tedy údaj z hlavičky.

Pokud dojde při čtení k chybě (přerušená nahrávka apod.), program ukončí čtení a ohlásí chybu hlášením ERROR. Stejně hlásí se může objevit až po ukončeném čtení tehdy, jestliže nehlásí přečtený kontrolní součet se součtem, který se v průběhu čtení vytváří u všech datových bytů.

V obou případech je nutné čtení opakovat.

Znovu upozorňuji na potřebu zvýšené opatrnosti při používání paměti RAM nad C000. Přepsání adres v oblasti D000 až D nebo FFFA - FFFF může vést k havárii programu a ke ztrátě počítači přímé nebezpečí nehrozí).

### 3. - LOAD

Příkaz se používá pro čtení standartního formátu ATARI s posuvnou rychlostí 600 - 800 Bd. Zadává se počáteční adresa RAM, odkud budou načtená data uložena. Použitelný rozsah je od E00 BFFF. Při standartním čtení se používá operační systém v ROM, tudíž ji nelze odpojit. U delších programů rozdělených do více částí však lze použít nejdříve příkaz LOAD do povolené oblasti a potom pomocí BLOCKTRANSFER přesun nad adresu C000.

Čtení se ukončí automaticky timeoutem (přestávka v datech delší než cca 40 s) nebo stisknutím klávesy BREAK nebo nalezením konce bloku. V případě chyby se objeví chybová zpráva ERROR čtení se přeruší. Vždy se po skončení vypíše délka tj. počet načtených bytů.

#### NOTE

Pokud jsou meziblokové mezery nestandardní délky, pak při příhodném SAVE se jejich původní délka nezachová. Tuto vlastnost má ze známých kopírovacích programů pouze CASDUP (CITY VERVAN).

### 3.9 M - MONITOR

Pro opravy a úpravy programu je nezbytné mít prostředek k prohlížení obsahu paměti a jeho změně. To usnadňuje právě příkaz MONITOR. Po zadání počáteční adresy program vypíše:

adresa 1 - obsah 1

a očekává vstup některého z následujících znaků:

a) X - ukončí práci v MONITORU. Program je ve stavu, kdy očekává vstup příkazu SUPERMONU.

b) RETURN

na další řádek se vypíše následující adresa a její obsah

adresa 1 - obsah 1

adresa 2 - obsah 2

c) mezera - bezprostředně za vypsany obsah paměti se vypíše obsah následující adresy. Tím je dosaženo úsporného výstupu obsahu větší části paměti na obrazovku.

Např. použití 3x mezera vypíše:

adresa 1 - obsah 1 - obsah 2 - obsah 3 - obsah 4

d) klávesa se symbolem "-" (pohyb kurzorem nahoru). Po jejím stisknutí (bez CTRL) se na nový řádek vypíše předcházející adresa a její obsah.

Např. adresa 1 - obsah 1

adresa 2 - obsah 2 (RETURN)

adresa 1 - obsah 1 -

Tím je umožněno listovat pamětí oběma směry.

e) hexadecimální číslo složené z jednoho až dvou hexadecimálních znaků, ukončené jakýmkoliv jiným znakem mimo X, je chápáno jako nový obsah právě vypsané adresy. Pokud chceme zadávat obsah větší části paměti, lze psát hexadecimální čísla za sebou a oddělovat je čárkou.

Př.: MONITOR - od adr: 1200 (RETURN)

-----

1200 - A0 - B2,19,24,1,F3,C,6A

-----

uloží na adresy 1200 - 1206 hodnoty B2,19 ....

(Podtržené části vypisuje program.)

#### NOTE

Opravit lze pouze právě zadávanou hodnotu! Napsáním ukon-

čovacího znaku, tedy i čárky se obsah paměti změní a použití klávesy DELETE BACK SPACE již nemá vliv. V tomto případě je nutno pro opravu použít klávesu "-", kterou se vrátíme k inkriminované adrese a pak její obsah můžeme změnit.

- f) jakákoliv jiná klávesa mimo uvedené v bodech a) až e) způsobí výpis další adresy a jejího obsahu bez přechodu na nový řádek. Tento způsob výpisu není standartní a je nepřehledný.

### 3.10 N - NAME

Příkaz platí pouze pro verzi V2.1 a umožňuje změnit název programu v aktuální hlavičce. Po načtení programu není nutné zadávat celý příkaz T. Stačí změnit název pomocí N a zapsat příkazem R. Změnu názvu použijeme hlavně v případě, kdy by se při kopírování na jednu kazetu sešly dva programy stejného názvu.

Zadávání názvu se ukončí pomocí (RETURN), nebo automaticky po napsání desátého znaku.

### 3.11 R - REPEAT ZX SAVE

Příkaz lze použít pro tyto účely:

- a) opakování zápisu na kazetu ve formátu ZX podle předcházejícího příkazu T
- b) zápis na kazetu ve formátu ZX po předchozím čtení příkazem I. V příkazu se nezadávají žádné parametry, všechny hodnoty jsou převzaty ze zadání příkazu T nebo I.

Mezi příkazy T (I) a příkazem R lze měnit obsah paměti nebo záznamovou rychlost. Příkaz REPEAT je vhodný pro pořizování více kopií, případně kopií s určitými změnami bez nutného zadávání všech parametrů jako u ZX SAVE.

Pozor u verze V2.0 na použití RESETu, který likviduje obsah hlavičky. Proto po něm nelze použít R, ale je nutno znovu zadávat příkaz T.

### 3.12 S - SAVE

Příkaz je určen k zápisu libovolné oblasti RAM na magnetofon ve standartním formátu ATARI 600 Bd. Při výstupu jsou použity krátké meziblokové mezery. Zadávají se dva parametry:

od adr: adresa počátečního bytu zapisované oblasti RAM  
délka: počet bytů, které mají být zapsány.

Save lze využít zejména při úpravách zavaděčů, které mají zůstat ve formátu ATARI, případně k převodu mezi formáty ZX a ATARI.

### 3.13 T - ZX SAVE

Příkaz pro zápis libovolné oblasti paměti RAM na magnetofon ve formátu ZX. Zadávají se následující parametry:

od adr: adresa v RAM, od které se začnou brát data určená pro zápis  
typ: zadává se 3 nebo 4 (viz příkaz ZX LOAD)  
název: 1 až 10 libovolných znaků, jako název, který bude uložen v hlavičce programu.

Doporučuji nepoužívat v názvech mezeru. Přípravují se programy, které umožní vyhledávat soubory na kazetě podle názvu. V tom případě se každý nezobrazovaný znak (mezera, tabelátor..) snadno při zadávání přehlédne.

od adr: (nebo "zav.adr:" na tomto místě se zadává adresa, od které má být zapsaný program skutečně uložen po jeho zavedení zavaděčem. Tato adresa musí přesně odpovídat adrese, od které je program sestaven. V případě posunutí nebude program po zavedení fungovat.

délka: počet bytů vlastního programu

start: startovací adresa. Od této adresy začne vykonávání programu po zavedení.

několik poznámek:

- pro typ 4 jsou zaváděcí adresa a startovací adresa pouze formální a zadává se na jejich místo hodnota 1 (není podmínkou). Zaváděcí program ZXI si skutečné hodnoty vyhledává v programu sám.
- programy typu 4 nejsou zaveditelné zavaděčem TURBO 2000
- před povelom T je možné nastavit jinou přenosnou rychlost příkazem V
- popis úprav standartních programů přesahuje rámec tohoto doku-

mentu. Přesto bude na závěr uveden příklad, jak lze zkusit "zturbovat" některé programy bez hlubších znalostí programování.

### 3.14 V - rychlost (verze 2.0)

Příkaz se týká zobrazení, případně změny přenosové rychlosti pro čtení a zápis formátu ZX. U první verze SUPERMONu byla plynule nastavitelná v rozmezí cca 1000 - 8000 Bd. Po přechodu na formát kompatibilní s pražským TURBO 2000 by si plynulé nastavení vyžádalo podstatné zvýšení nároků SUPERMONu na paměť. Proto bylo zvoleno 6 rychlostí, které postihují celý použitelný rozsah. Rychlost se tedy zobrazuje a zadává jako číslo v rozsahu 0 - 5 s tímto významem:

- 0 - 1500 Bd
- 1 - 2500 Bd - TURBO 2000
- 2 - 3000 Bd (dvojnásobek rychlosti 0)
- 3 - 4000 Bd
- 4 - 5000 Bd
- 5 - 6000 Bd (čtyřnásobek rychlosti 0)

Uvedené rychlosti jsou pouze orientační. Způsob záznamu je takového charakteru, že skutečná přenosová rychlost je závislá na obsahu dat (doba přenosu jednoho bitu = 1 je dvojnásobná než doba přenosu jednoho bitu = 0). Tak např. reálná změřená rychlost na programech délky několika desítek tisíc bytů u rychlosti 1 byla v rozmezí 2200 - 2800 Bd. U rychlosti 5 to bylo až přes 7000 Bd. Rychlost zůstává nezměněna až do dalšího zadání povelu V, nemění se ani RESETEM.

### 3.15 V - rychlost (verze 2.1)

U verze V 2.1 lze příkazem V zadat rozdílné rychlosti pro čtení a pro zápis. Hodnoty zůstávají platné, jak jsou uvedeny v předchozí kapitole, ale je nutné vždy zadat dvě čísla hned za sebou v pořadí čtení, zápis. Čísla se neoddělují. Pořadí napovídá výpis, kde v závorce je uvedeno (RW), tedy READ,WRITE. Po zadání V se vypíše právě platné rychlosti, pak lze zadat nové nebo ponechat původní (odpověď RETURN ).

Př. rychlost (RW)-11 13

tj. čtení i zápis byly ze standartních hodnot změněny na

čtení "jedničkou" a zápis "trojkou".

Pro příkaz COMPARE se vždy pamatuje rychlost posledně použitého přenosu. Předcházel-li příkaz T nebo R použije COMPARE rychlost jako při zápisu, jestliže se četlo příkazem I, použije se pro komparaci rychlost čtení.

Oddělené zadávání rychlosti je výhodné zejména pro kopírování většího počtu programů z jedné rychlosti do druhé.

#### 4. Vytvoření vlastní varianty SUPERMONu

Po zavedení SUPERMONu jsou nastaveny rychlosti na standardní hodnotu 1 (Turbo 2000). Pokud chcete mít verzi s jinou rychlostí lze postupovat takto:

- a) příkazem V změnit rychlost na požadovanou hodnotu
- b) uložit novou verzi SUPERMONu na kazetu příkazem SAVE od adr: 480 délka: 800 (u verze 2.1 je délka 880)

#### 5. Konverze programu do formátu ZX

Především upozorňuji, že tato činnost není triviální a principiálně nelze sestavit obecně platný návod na převedení libovolného programu ve formátu ATARI do formátu ZX. Proto zde uvádím postup, který lze vyzkoušet pro jeden druh programů, i když ani v tomto případě se nedá zaručit stoprocentní úspěšnost. Trochu blíže se o tomto problému zmiňuji ve stati o formátech souborů na kazetách počítačů ATARI, která by měla vyjít ve zpravodaji Olomoucké pobočky ATARI klubu.

Popisovaná úprava se týká tzv. relativních souborů, tj. všech programů, které lze nahrát zavaděčem BL/C, vykřičníkovým zavaděčem apod.

- a) příkazem L zavedem program do paměti např. od adr. 1000. Zavádí se pouze druhá část (bez zavaděče). Zapamatujte si délku, kterou po dokončení vstupní operace SUPERMON ohlásí.
- b) příkazem MONITOR - Od adr: 1000 ... se přesvědčíme zda program začíná hodnotami 1000 - FF  
1001 - FF

-  
-

Pokud ano, je to dobré, pokud ne, nemusí to být ještě chyba.

c) příkazem ZX SAVE (T) zapíšem program na kazetu:

ZX SAVE - od adr: 1000

Typ: 4

název: (zkratka názvu programu 1 až 10 znaků)

Od adr: 1 (zde na hodnotě nezáleží)

délka: hodnota, kterou vypsál příkaz LOAD

Start: 1 (opět na hodnotě nezáleží)

d) Provést případně komparaci

e) Zkusit zavést a spustit pomocí zavaděče ZX1 (zavaděč TURBO 2000 pro tento typ souboru n e f u n g u j e ).

## 6. Seznam změn verze V 2.1

1. Testovací smyčka vstupu ZX je kratší (cca 7  $\mu$ s proti 10  $\mu$ s ve verzi 2.0). Teoreticky tak lze dosáhnout ještě vyšší přenosové rychlosti.
2. Počet pulzů nutných k identifikaci zaváděcího kmitočtu se zvýšil z 255 na 765 pro zlepšené hledání programu na kazetě.
3. Paměť ROM se odpojuje automaticky v příkazech B,C,I,R a T.
4. U všech vstup/výstupních operací je uplatněna funkce BEEPWAIT; po zvukovém signálu (jedno, nebo dvě zavrčení) čeká program na stisk libovolné klávesy než zahájí vlastní operaci.
5. Je přidán příkaz N - změna názvu v hlavičce
6. Rychlost přenosu ZX lze volit různou pro vstup i výstup
7. U příkazu I (ZX-load) lze na dotaz "Od adr:" odpovědět stisknutím (RETURN) nebo mezera. Nastaví se automaticky adresa E00. Tím se zjednodušuje prosté kopírování na posloupnost příkazů:
  1. I + mezera
  2. (po vypsání hlavičky) mezera
  3. (po výměně kazet) R + mezera

## 7. Praktické zkušenosti s rychlostí

1. U DATASETu upravených použitím OZ MAA 741 nemusí být spoleh-

livá již ani "dvojka".

2. DATASETY s rožnovskou úpravou s využitím OZ MAC 111 spolehlivě fungují s "trojkou", většina z nich i se "čtverkou" na kvalitní kazetě.
3. "Pětka" je zatím použitelná pouze pro kvalitní magnetofony s vyrovnanou frekvenční charakteristikou a minimálním kolísáním rychlosti.

#### 8. Použití zavaděče ZXL V 3.0

Zavaděč ZXL verze 3.0 má tabulkově stanoveny stejné rychlosti jako SUPERMON V 2.0 a 2.1. Fyzicky existuje zavaděč jednak na kazetě jako absolutní program délky 5 bloku, nebo se s ním můžete setkat na cartridge.

##### 1. ZXL na kazetě:

1. zavede se do paměti běžným způsobem jako strojový program (tlačítka OPTION + START)
2. do magnetofonu se vloží kazeta s programy ve formátu ZX a stiskne se tlačítko PLAY
3. na klávesnici stiskneme číslici 0 až 5 podle rychlosti, jakou je zaváděný program zapsán na kazetě
4. jakmile zavaděč najde na kazetě hlavičku, zobrazí jméno programu a zastaví motor
5. chceme-li tento program nahrát, stiskneme libovolnou klávesu. Program se zavede a spustí.
6. pokud nechceme program, jehož jméno se zobrazilo, stiskneme RESET posuneme pásek na potřebné místo a opakujeme postup od bodu 3

##### 2. ZXL na cartridge

1. vypnout počítač
2. vložit cartridge
3. zapnout počítač
4. zmáčknout tlačítko na cartridge a držet



```
      ORG  *AD00
      LDX  #A7      ; počet přemísťovaných stránek
ZAC   LDY  #0
OD    LDA  (*600), Y
NA    STA  (*400), Y
      INY
      BNE  OD
      INC  OD+2
      INC  NA+2
      DEX
      BNE  ZAC
      JMP  START
```

\* - znamená znak "dolar"

Zavaděč může pracovat ve čtyřech různých režimech, které se volí stisknutím určité klávesy (nebo kombinací dvou kláves) po startu nebo po RESETu nebo po přerušení BREAKem.

### 1. Režim 1.

Číslice 0 až 5 - nastaví příslušnou rychlost čtení (odpovídá rychlostem v SUPERMONU; jednička je shodná s rychlostí TURBO 2000), přečte první hlavičku na pásce, zobrazí ji a čeká na potvrzení.

L - používá se výhradně po stisknutí klávesy BREAK a znamená: pokračovat ve zvolené činnosti (vyhledávání, zobrazování...)

ostatní klávesy - stejně jako při stisknutí číslice ovšem bez nastavení rychlosti. Ta zůstává nastavena podle předcházejícího povelu. Na počátku je rovna 1.

Současně s předchozími klávesami je možno stisknout OPTION, SELECT nebo START s tímto významem:

### 2. Režim 2 - listiny programů

OPTION - zobrazuje postupně všechny názvy programů, které na mag. pásce nalezne. Po zobrazení vždy vyčkává cca 5 sec a pak pokračuje ve čtení.

### 3. Režim 3.

SELECT - vypíše se dotaz NÁZEV. Lze zadat název programu

(1-10 znaků), který chceme vyhledat a spustit. Každý nalezený program se zobrazí stejně jako u OPTION a porovná se tolik znaků od začátku názvu, kolik bylo zadáno. Pokud souhlasí, program je zaveden a odstartován.

#### 4. Režim 4.

START - zavede a odstartuje první program, který na kazetě najde. Není nutné čekat na nalezení hlavičky a potvrzovat zavedení programu stiskem klávesy.

V režimu 1 je nutné po vypsání hlavičky stisknout libovolnou klávesu, aby mohlo zavádění pokračovat. Pokud stiskneme L, program se nezavede, ale zavaděč najde a nabídne další.

Klávesa BREAK přeruší vždy činnost zavaděče a obnoví výpis na obrazovku. To lze využít především v režimech 2 a 3 k zviditelnění programů na obrazovce. Obnovit činnost zavaděče lze v těchto režimech klávesou L.

Pozor! Pokud již probíhá zavádění programu, pak jeho přerušování klávesou BREAK způsobí nefunkčnost takového programu. Pokračování klávesou L pak nemá význam.

Některé příklady k osvětlení uvedeného popisu

1. START + 3 - okamžité zavedení a start prvního programu na kazetě rychlostí 3.

2. SELECT + mezera - po výpisu NÁZEV: SPA

Zavaděč postupně zobrazí např.:

CRYSTAL

SPY

SPACEPANIC

Zadané znaky souhlasí s názvem. Program je zaveden a odstartován. Rychlost přenosu zůstává nastavena od předchozího příkazu (po zavedení z cartridge je rovna jedničce t.j. TURBO 2000).

3. OPTION + 1 - nastaví se čtecí rychlost 1 a na obrazovku se postupně vypisují názvy nalezených programů rychlostí "1". Tento listing lze zviditelnit klávesou BREAK a obnovit stisknutím "L".

Zavaděč ZXI 3.1 zavádí následující typy programů.

Ø3 - absolutní

Ø4 - relativní

FE,FF - BASIC-ovské

Poslední skupina programů vzniká jako výstup z BASIC-DOS, nebo TURBOBASIC-DOS. Lze je rovněž získat převedením standartně zapsaného programu z BASICu (příkazem CSAVE) do turboverze např. pomocí programu SUPERMON. V tomto případě na hodnotě zavaděcí ani startovací adresy nezáleží, důležité je pouze uvedení typu FF nebo pro TURBOBASIC FE a celkové délky.

ZXI takovéto programy zavede a spustí ATARI ROM-BASIC. Programy z TURBOBASICU proto nemusí fungovat!

Potřebujeme-li po zavedení jedné části programu pokračovat v zavádění, je nutné k 1.části přidat rutinu, která po odstartování nastaví na adrese 2A hodnotu 1 a znovuodstartuje zavádění . pomocí horkého RESETu (JMP E474). To způsobí, že následující pokus o čtení z klávesnice nečeká na stisk klávesy, ale chová se stejně jako při stisknutí klávesy RETURN . V případě ZXI tedy začne zavádění další části při nezměněné přenosové rychlosti.

## MODUL CARTRIDGE

-----

AK Rožnov pod Radhoštěm

J. Halamík, P. Vičar

### 1. Základní informace

Domácí počítače ATARI jsou vybaveny, na rozdíl od jiných levných osmibitových počítačů, vstup/výstupním rozhraním (interface), přes které může počítač komunikovat s okolím. U typu ATARI 800, XL, XE, (130XE) jsou tato datová rozhraní:

1. paralelní systémová sběrnice
2. seriový vstup/výstup
3. vstup/výstupní porty ovladačů
4. zásuvka pro moduly CARTRIDGE.

Následující publikace se bude zabývat popisem a využitím zásuvky typu CARTRIDGE (CART).

Starší provedení osmibitových počítačů ATARI 600 a 800 měly dvě zásuvky typu CART, které v principu fungovaly obdobně. Jedna byla adresována jako spodní polovina dále popisovaného modulu CART a druhá jako horní polovina. Využití těchto dvou zásuvek je obdobné a nebudeme se jím podrobně zabývat.

U počítačů ATARI XL, XE a 130XE je jedna zásuvka typu CART, má však různé umístění na počítači. U typu 800XL se nachází na horní straně počítače, u typu 800 XE a 130 XE na zadní straně počítače. Elektrické a funkční vlastnosti obou provedení zásuvek u těchto typů jsou shodné.

Zásuvka CART je určena pro rychlé zadávání programů do operační paměti počítače. V případě jejího použití nahrazuje nahrávání dat (programu, her) do počítače z datamagnetofonu či diskové jednotky. Modul CART obsahuje paměti typu ROM, ve kterých jsou uložena potřebná data (program). V základním provedení umožňuje pojmout až 16kbytu programu s téměř okamžitým přístupem (ms) po zasunutí do zásuvky. Moduly CART jsou prodávány naprogramované určitým programem podobně jako data-kazety.

### 2. Originální zapojení modulu CARTRIDGE

V originálním provedení se modul CART skládá z krabičky a

tištěného spoje s přímým konektorem. Tištěný spoj propojuje dvě paměti typu ROM, z nichž každá má kapacitu 8k bytů. Elektrické zapojení (viz obr.1) tvoří dvě uvedené ROM paměti, které mají společnou 13 bitovou adresovou a 8 bitovou datovou sběrnici. Paralelně nejsou zapojeny pouze vstupy výběru obou obvodů (CS). Vyšší adresový prostor (paměť A 602) je ovládán signálem "S5, nižší adresový prostor signálem "S4. Tento výběr provádí obvod MMU (memory management unit), který je součástí počítače.

Obvod MMU provádí organizaci výběru paměťového prostoru celého počítače s danou prioritou, která je aktualizována vždy po inicializaci počítače (zapnutí, RESET). Nejvyšší prioritu má paměťový modul CART. Schematicky je organizace sdíleného paměťového prostoru znázorněna na obr. 2.

Pracuje-li počítač s modulem CART jsou ROM paměti modulu (ROM CART) využívány jako "stínové RAM" paměti. Spodní ROM CART (A601) zabírá paměťový prostor 8000 - 9FFF(hex), horní ROM CART (A602) A000 - BFFF(hex).

Elektricky je přítomnost zasunutého modulu CART testována jednotkou MMU přítomností log úrovně H na vstupech modulu RD5 a RD4 v době inicializace počítače. Poté je generován signál log L výběru ROM CART (na "S5 pro H na RD5). Podrobně bude vysvětleno v odstavci popisujícím softwarovou část spolupráce modulu CART a počítače.

Destička tištěného spoje modulu je oboustranná a tvoří zároveň pokovanými ploškami přímý konektor 2 x 15 vývodů s roztečí 2,50mm. Kromě této rozteče je třeba dodržet celkovou šířku přímého konektoru, která je upravena vedením v zásuvce modulu CART na 40 mm.

### 3. Popis zásuvky modulu cartridge

Pohled od klávesnice, orientace přímého konektoru

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
S	R	P	N	M	L	K	J	H	F	E	D	C	B	A

Popis vývodů	1 "S4	A RD4
	2 A3	B GND
	3 A2	C A4
	4 A1	D A5
	5 A0	E A6
	6 D4	F A7
	7 D5	H A8
	8 D2	J A9
	9 D1	K A12
	10 D0	L D3
	11 D6	M D7
	12 "S5	N A11
	13 +5V	P A10
	14 RD5	R R/W
	15 "CCTL	S B02

#### Funkce vývodů

A0 ... A12	Adresy paměti
D0 ... D7	Data
"S4	Výběr spodního adresového prostoru (8000-9FFF)
"S5	Výběr horního adresového prostoru (A000-BFFF)
RD4	Příznak osazení spodního adresového prostoru
RD5	Příznak osazení horního adresového prostoru
R/W	Řízení čtení/zápis (v orig. nepoužito)
B02	Hodiny procesoru (nepoužito)
"CCTL	Výběr paměťového prostoru (D500-D5FF)

Poznámka: "X" znamená negace X

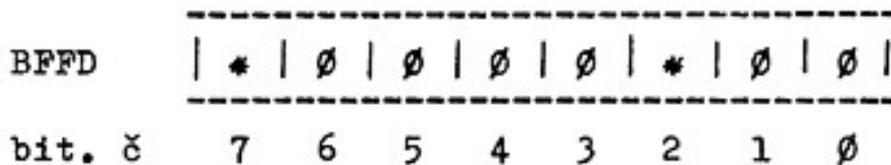
Modul CART může být využíván buď v původním určení tj. jako externí paměťový modul (programový modul) pracující jako "stínová RAM" - v tomto případě je nutné, aby byl modul zapojen po celou dobu využívání jeho programové náplně nebo jako externí ROM DISK, ze kterého jsou data překopírována do vnitřní RAM a dále je modul odpojen.

#### 4. Spolupráce CART s operačním systémem

Operační systém (OS) obsahuje celou řadu podprogramů určených k identifikaci CART, kontrole zasunutí, inicializaci apod.

Přítomnost CART je indikována v registru DØ13 hodnotou 1. Využívá se port B obvodu PIA, kde u typu ATARI 4ØØ byly připojeny joystiky 3 a 4. U novějších verzí 8ØØ XL, XE, 13ØXE apod. je tento port použit jako memory management (jednotka řízení paměti). Nenulová hodnota registru DØ13 je hardwarově nastavena v okamžiku, kdy se stane aktivním signál RD4 nebo RD5.

Při RESETEu je jednak přepsána hodnota z DØ13 na adresu 3FA, která se později používá ke kontrole zasunutí CART. Dále při RESETEu v přítomnosti CART testuje OS adresu BFFC, která má mít hodnotu Ø. Je-li to v pořádku, řídí se další činnost obsahem adresy BFFD.



Je-li bit č.7 nastaven (tj. roven 1) považuje systém tuto CART za tzv. diagnostickou a předává jí okamžitě řízení přes adresu BFFE.

Je-li hodnota bitu č.7 nulová, dokončí OS inicializaci běžnou při RESETEu a provede inicializační rutinu, jejíž adresa je v CART na BFFE. Po návratu z této rutiny je odstartován program podle hodnoty bitu č.2 adresy BFFD. Je-li nastaven, je startovací adresa na BFFA. Je-li nulový, pak je startovací adresa na A (DOSVKT).

Při každém přerušení VBI porovnává OS obsah adresy 3FA s hodnotou registru DØ13 a při jejich nesouhlasu přejde do rutiny WAITFRST, tedy čekání na RESET. Jde o jednoduchou smyčku typu:

```
ADR      JMP  ADR      ; skok zpět na sebe
```

Z předchozího rozboru vyplývá, že CART s běžným programem musí v posledních šesti bytech obsahovat tyto údaje:

- BFFA - dolní byte startovací adresy
- BFFB - horní byte startovací adresy
- BFFC - Ø (příznak CART)
- BFFD - 4 (inic. rutina a start přes BFFA)
- BFFE - dolní byte inicializační rutiny
- BFFF - horní byte inicializační rutiny

Nevýhodou tohoto způsobu použití je automatické zablokování paměti RAM v rozsahu A000 až BFFF tedy 8 kbytu bez ohledu na skutečnou velikost ROM na CART. To platí v případě aktivního signálu RD5. Je-li aktivní i RD4, odpojí se navíc RAM v oblasti 8000 - 9FFF, tedy dalších 8 kbytu.

Navíc program musí být tvořen tak, aby pracoval při umístění na uvedených fyzických adresách. Protože automatická transformace již přeložených programů není možná, není tento způsob použití nejlepší. Proto byl pro potřeby uložení zavaděče ZXI do CART vytvořen a odzkoušen následující mechanismus, který uvedené nedostatky plně odstraňuje a je obecně použitelný i pro jiné programy.

Struktura obsahu paměti v CART je takováto:

vlastní program	INIT2	INIT1	HL
-----------------	-------	-------	----

- HL - hlavička, která byla již popsána
- INIT1 - inicializační rutina v ROM
- INIT2 - inicializační rutina v RAM

INIT1 - jeho adresa je na BFFE a pracuje přímo v CART po RESETu. Jeho úkolem je přenést vlastní program včetně INIT2 z CART do RAM na skutečnou adresu, pro kterou byl program vytvořen.

INIT2 - je spuštěn operačním systémem po návratu z INIT1. Jeho adresa je na BFFA a musí být při spuštění umístěn mimo CART. Jeho hlavním úkolem je řízení odpojení CART. Kromě toho může být tato část použita pro nastavení např. adresy RESETu, nulování paměti apod. Tedy činnosti, které mají proběhnout pouze při zavedení programu. Po skončení činnosti INIT2 odstartuje vlastní program.

Připojení a odpojení CART je umožněno spínačem na CART, který v sepnutém stavu připojí napájecí napětí +5 V na paměť ROM (PROM nebo EPROM) a současně aktivuje signál RD5. Po jeho stisknutí při nejbližším přerušení VBI přejde systém do dynamického stopu, jak bylo popsáno výše. Držíme spínač dále a krátkodobě stiskneme RESET. Tím se odstartuje INIT1 a po jeho

zkončení INIT2. Tato rutina v sobě obsahuje následující sekvenci instrukcí:

```
---  
SEI                zákaz přerušeni  
TEST LDA DØ13     odpojena CART ?  
BNE TEST          ne, testuj znovu  
STA 3FA           ano, poznač do systému  
CLI              obnov přerušeni  
---
```

V okamžiku, kdy pustíme tlačítko, odpojíme napájení CART, signál RD5 přestane být aktivní a v DØ13 se objeví stav Ø. Ten se přepíše do stínového registru 3FA a může se obnovit přerušeni. Tím jsme vlastně přemluvili systém, že žádnou CART vloženou neměl a zavedený program pracuje tak, jako by byl zaveden z kteréhokoliv jiného média.

Na tomto místě je třeba upozornit, že některé odlišnosti v systému přece zůstávají (velikost paměti, umístění DL v RAM apod.) takže pro další výzkum jsou ještě dveře otevřeny.

## 5. Příklad programu v CART

Prvním programem rutinně používaným v modulu CART je zaváděč ZXL V3.Ø (přibližně Turbo 2ØØØ). Vlastní program zabírá necelý 1 k Byte paměti a zahrnuje kromě vlastního zaváděcího programu ve formátu ZX také tabulku volby záznamové rychlosti, což je velký přínos oproti programu Turbo 2ØØØ. Záznamová rychlost "1" je přitom kompatibilní s Turbo 2ØØØ.

Hexadecimální výpis obsahu PROM se zaváděcím programem ZXL V3.Ø je v příloze, proto několik slov k jeho činnosti. Program slouží ke čtení a spuštění programu z magnetofonu (DATASETu) v kódu ZX (Turbo 2ØØØ). Po dokončení inicializačních rutin očekává ZXL zadání číslice v rozsahu od Ø do 5. Tím se zadává přenosová rychlost, s jakou byl zaváděný program zapsán na kazetu. Označení přenosových rychlostí odpovídá SUPERMONu. Programy v Turbo 2ØØØ se tedy zavádějí rychlostí 1 (jednička).

Podle zadaného čísla se nastaví příslušné časové konstanty pro zaváděcí kmitočet, synchronizační pulz a pro délky pulzů odpovídajících logické "Ø" a log. "1". Potom zaváděč čeká ve smyč-

ce tak dlouho, dokud se na vstupu počítače nezmění stav. Od tohoto okamžiku se měří délky pulzů na až do doby, kdy 255 po sobě jdoucích pulzů vyhovuje svojí délkou očekávané délce pulzů zaváděcího kmitočtu. U následujících pulzů se kontroluje správná délka a po zjištění odchylky se porovná se synchronizačním pulzem. Je-li to on, přečte se jeho druhá polovina a začíná čtení vlastního datového bloku tak, že se podle zjištěné délky pulzu do následujícího bitu čteného bytu nastaví buď "0" (kratší pulz) nebo "1" (delší pulz).

Uvedeným způsobem se nejdříve přečte hlavička, na obrazovku se vypíše název a zjistí se o jaký typ souboru se jedná.

Typ 3 - absolutní. Celý datový blok se přečte a uloží na adresu uvedenou v hlavičce a program se odstartuje od adresy, rovněž uvedené v hlavičce.

Typ 4 - relativní. Při zavádění se vždy nejdříve z dat přečte počáteční a koncová adresa, pak se čte příslušný počet bytů a ukládá na uvedené adresy. Znovu se čte počáteční a koncová adresa ... a postup se opakuje až do přečtení všech dat k magnetofonu (celková délka je uvedena v hlavičce). Program je odstartován z adresy uložené na adresách 2E0, 2E1.

Zavaděč je uložen v nejméně používané oblasti paměti RAM a to od 400 do 600. Proto je možné po skončení některých programů jej znovu vyvolat stiskem RESETu. Pro bezchybnou funkci je ale vhodnější znovu zavést ZXL z CART. Pro ty, kteří nevlastní popis SUPERMONu popíší znovu způsob zavádění:

1. vypnout počítač
2. vložit cartridge
3. zapnout počítač
4. zmáčknout tlačítko na cartridge a držet
5. stisknout a pustit RESET
6. až se na obrazovce objeví kurzor, pustit tlačítko na cartridge. Na obrazovce se musí objevit název ZXL.
7. na klávesnici stiskneme číslici 0 až 5 podle rychlosti, jakou je zaváděný program zapsán na kazetě
8. jakmile zavaděč najde na kazetě hlavičku, zobrazí jméno programu a zastaví motor
9. chceme-li tento program nahrát, stiskneme libovolnou klávesu.

Program se zavede a spustí.

10. pokud nechceme program, jehož jméno se zobrazilo, stiskneme RESET posuneme pásek na potřebné místo a opakujeme postup od bodu 7

Ke znovuzavedení ZXI z cartridge není nutné vypínat počítač, stačí postupovat od bodu 4.

## 6. Využití modulu CART jako externí "ROM disk"

Tento způsob využití modulu CART je výhodný když potřebujeme zavést do počítače programy typu: monitor, editor, kopírovací program či zaváděč programů resp. jinak modifikovat operační systém počítače, který potom běžně manipuluje s daty. Snad nejvíce se výhoda rychlého nahrání systémového programu do počítače využije v případě Zaváděče formátu ZX (Turbo 2000). Zaváděcí program pak může být nahrán do počítače pouhým inicializováním modulu CART - stisknutím tlačítka.

Zapojení modulu bylo realizováno ve dvou provedeních, které se liší realizací tištěného spoje.

Tištěný spoj označený KART 1(2) vychází z originálního řešení, které bylo rozšířeno o možnost použít až 6 druhů paměti EPROM (PROM), resp. umožňuje skládat kapacitu modulu KART s více pamětí. Toto provedení obsahuje na oboustranném tištěném spoji dvě pozice PROM paměti, každá s 28 vývody. Umožňuje proto propojit paměti s kapacitou 5 až 16 k bytů přímo, 64k bytů s přepínačem adresy (tj. 512x8 až 16384x8)

Minimalizované provedení (KAR) je realizováno také na dvojstranném tištěném spoji, má však jednu pozici pro PROM paměť s 24 vývody. Může zapojit proto paměť s kapacitou max 4k bytů.

Obě provedení modulu CART jsou zakončena oboustranným přímým konektorem s roztečí 15 x 2,5 mm. Kromě použité paměti (paměti) PROM či EPROM jsou na tištěném spoji zapojeny pouze blokovací kondenzátor (M1, M15 či tantal  $\mu$ F/6V) a tlačítko T1 (mikrospínač).

## 7. Popis zapojení modulu CART s různými paměťmi

Dále budou uvedena zapojení modulů CART různé kapacity re-  
alizované paměťmi s různou kapacitou. Pro program ZXL V3.0  
(i pro připravovanou verzi 3.1) postačuje kapacita 1 k byte.  
V úvahu však přichází další programy (připravuje se SUPERMON tj.  
monitor ZX formátu, který umožňuje turbování, kopírování a úpra-  
vy programů), které budou vyžadovat větší kapacitu modulu CART.

### 7.1 Modul CART 1 k byte (obr.:3)

Umožňuje naprogramování např. programem ZXL. Paměť modulu  
je osazena PROM pamětí typu 93451 s kapacitou 1024 x 8 bitů.  
Vývody výběru obvodu "CS1 a "CS2 jsou spojeny s vývodem 12 na  
přímém konektoru - log. H adresou A10, vstup CS4 adresou A11.  
Deset adresovacích vstupů A0 až A9 je připojeno na odpovídající  
adresové vývody modulu CART.

### 7.2. Modul CART 2 k byte

Tento modul lze realizovat dvěma způsoby podle použitých  
pamětí. Na obr: 4 jsou zapojeny 2 paměti PROM, každá s kapaci-  
tou 1024 x 8 bitů. Adresové vývody A0 až A9 jsou zapojeny para-  
lelně. Systémová adresa A10 provádí výběr z obou pamětí přes vý-  
vody CS3 u obvodu P2 a "CS1 u P1. Adresa A11 je použita pro vý-  
vod výběru paměti CS4. Signál vybavení modulu "S5 ovládá vývody  
výběru "CS1 a "CS2 u paměti P2 a "CS2 u paměti P1. K zajištění  
sníženého odběru napájecího proudu je možné zařadit propustně  
pólovanou diodu do přívodu napájení Ucc. Tím se sníží odběr pa-  
mětí PROM asi o 20%.

Modul CART 2 k byte můžeme realizovat také s jednou EPROM  
pamětí typu 2716, která má kapacitu 2048 x 8 bitů, viz obr: 5.  
Vývody vybavení obvodu "CE a vybavení výstupu "OE jsou připoje-  
ny na signál vybavení modulu CART. Vstup Vpp je připojen na na-  
pájení Ucc. K adresování paměti jsou využity systémové adresy  
A0 až A10.

### 7.3. Modul CART 4 k byte

Na obr: 6 je zapojení modulu s EPROM pamětí s kapacitou  
4096 x 8 bitů, typ 2732. K adresování je využito 12 adresovacích

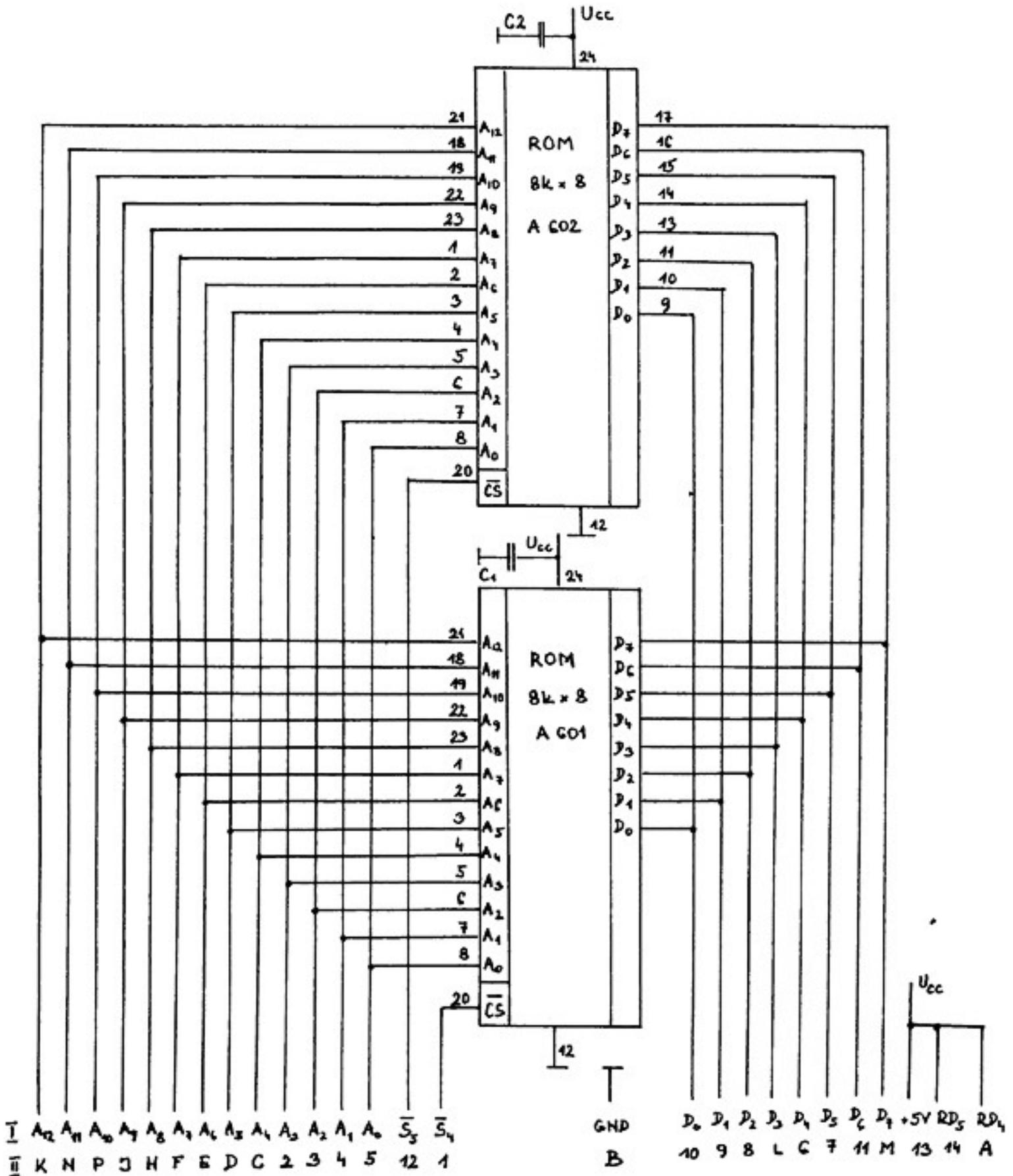
signálů A0 až A11. Signál výběru modulu "S5 ovládá vývod výběru obvodu i výstupu EPROM paměti. Modul CART s kapacitou 4k byte může být též realizován pomocí dvou pamětí s kapacitou 2k byte, např: EPROM 2716.

Modul CART je možné osadit maximálně dvěma pamětmi, každou o kapacitě 8192 x 8 bitů a tak realizovat modul CART s kapacitou 16 k bytů. V zahraničí jsou tyto paměti EPROM (typ 2764) nejdostupnější a nejlevnější a tudíž relativně nejvýhodnější. Perspektivně bude možné uložit do jedné takové paměti více programů zároveň, což dále zvýší technickou úroveň využití modulu CART. Tištěný spoj KART 1(2) umožňuje osazení tímto druhem EPROM paměti, proto se typ 2764 jeví pro moduly CART perspektivně jako nejvýhodnější. Elektrické zapojení je pak obdobné, jako u originálního provedení.

#### 8. Poznámky k použití

Je třeba upozornit na opatrnost při manipulaci s modulem CART. Tento modul je totiž přímo připojen na systémové (datové i adresové) sběrnice počítače bez oddělovačů. V případě elektrické kolise na vývodech modulu CART (zkrat vodivým předmětem, dotyk s vyšším elektrostatickým nábojem apod.) by mohlo dojít ke zničení obvodů uvnitř počítače, neboť tyto jsou přímo spojeny se systémovou sběrnicí. Aby k tomu nedošlo, je třeba tištěný spoj modulu KART uzavřít do obalu - krabičky !!! U počítače typu 800 XL je to navíc žádoucí kvůli otevření dvířek zásuvky modulu.

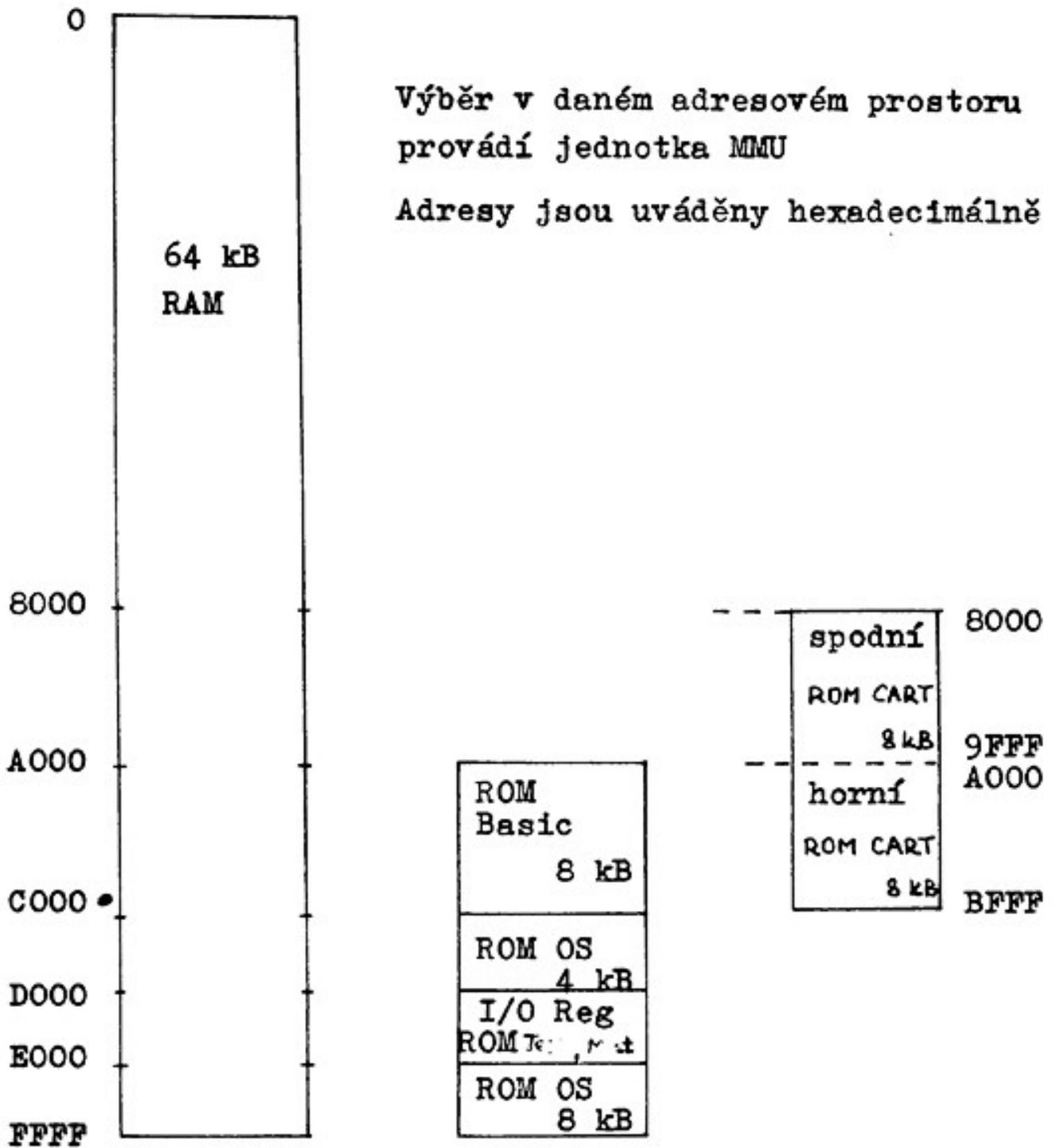
Závěrem je možné říci, že moduly CART byly ověřeny (v režimu "ROM DISK") ve více exemplářích. Programovou náplní byl výše uvedený zavaděč rychlého nahrávání ZXL Komfort obsluhy počítače mnohonásobně vynahradil náklady na realizaci modulu cartridge (CART). Práce na dalším možném využití modulu CART pokračují oživováním dalších (obslužných) programů, v modulech typu cartridge.



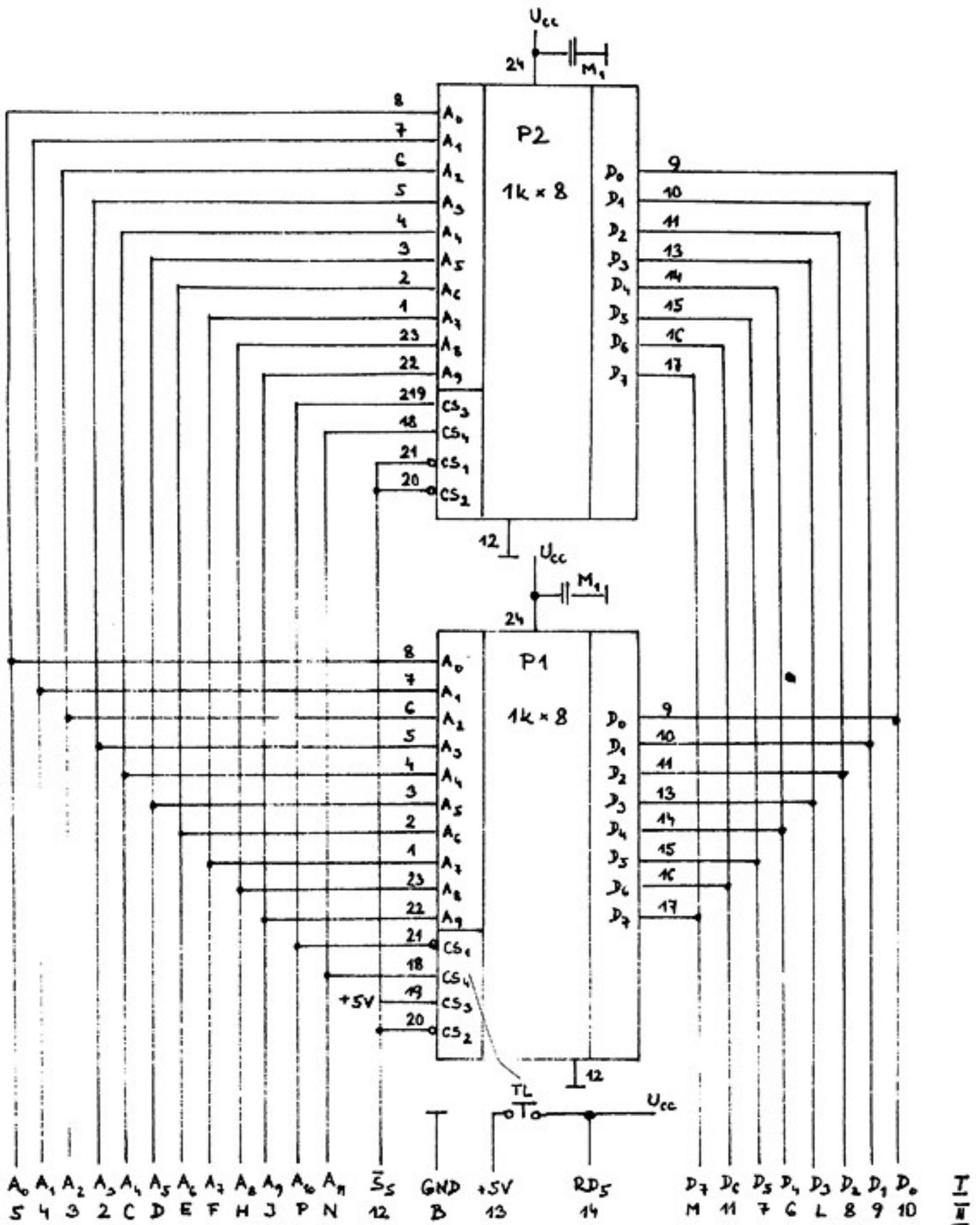
I Význam vývodů  
 II Označení vývodů přímého konektoru

Nepřipojeny: CCTL R/ $\bar{W}$  B02  
 15 A S

Obr. 1. Zapojení originál modulu CARTRIDGE



Obr. 2. Organizace paměti 800 XL (XE)



I... význam vývodů

II.. označení na přímém konektoru

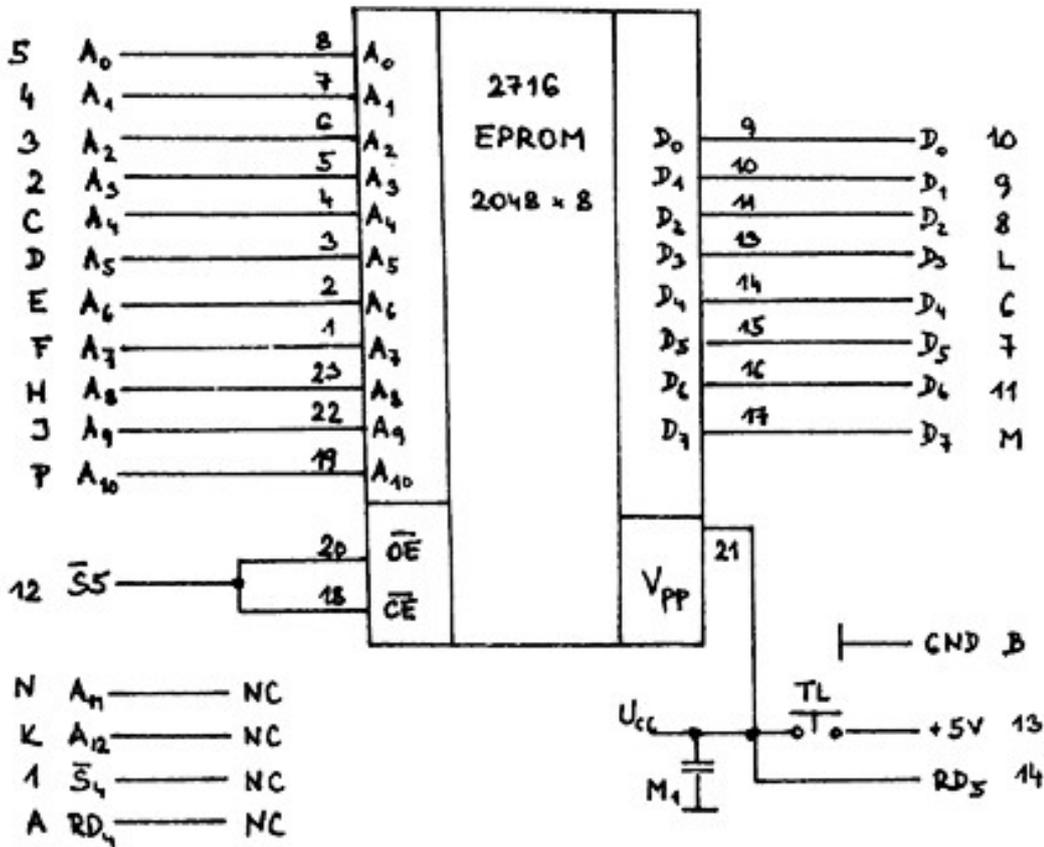
Obr. 4. Zapojení modulu KART 2k Bytes s pamětmi PROM 1k Bytes

93 451

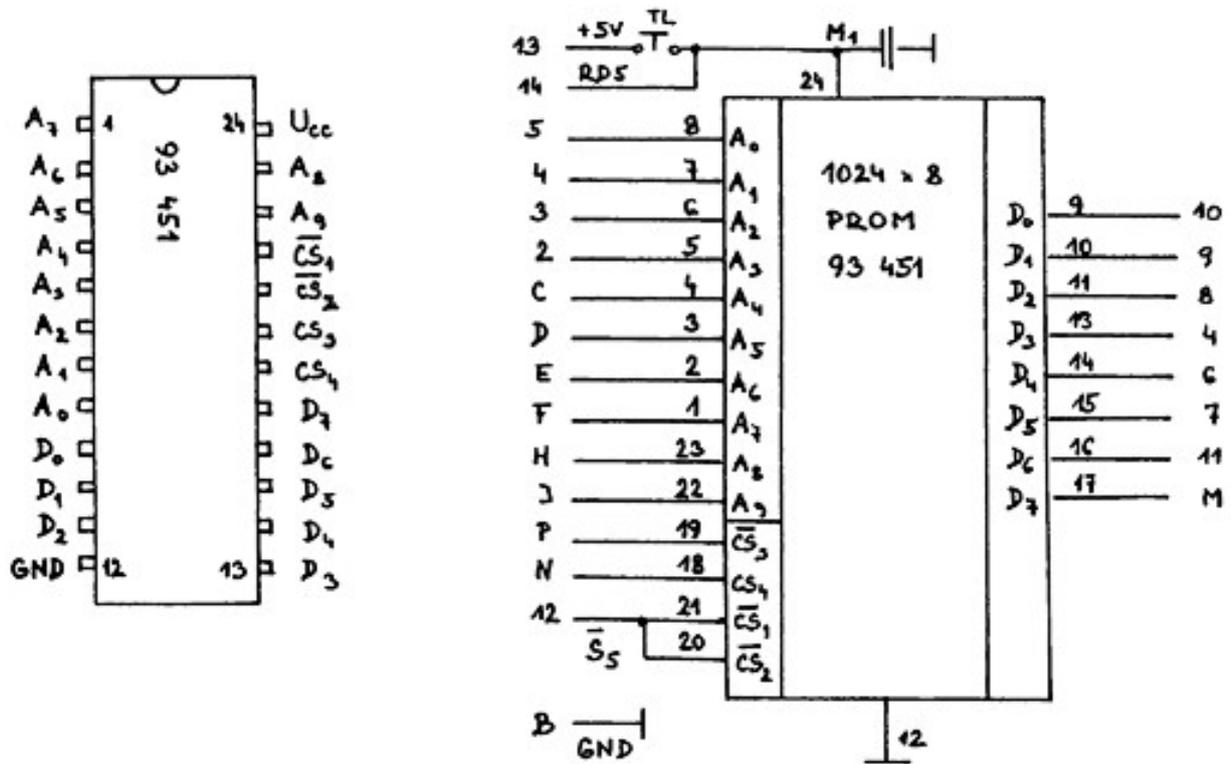


Piny Mod.	$\overline{CE}$ / PGM	$\overline{OE}$	$V_{PP}$	OUT
Read	L	L	+5V	Dout
Standby	H	X	+5V	Z
Progr.	$\overline{L}$	H	+25V	DIN
Inhibit	L	H	+5V	Z

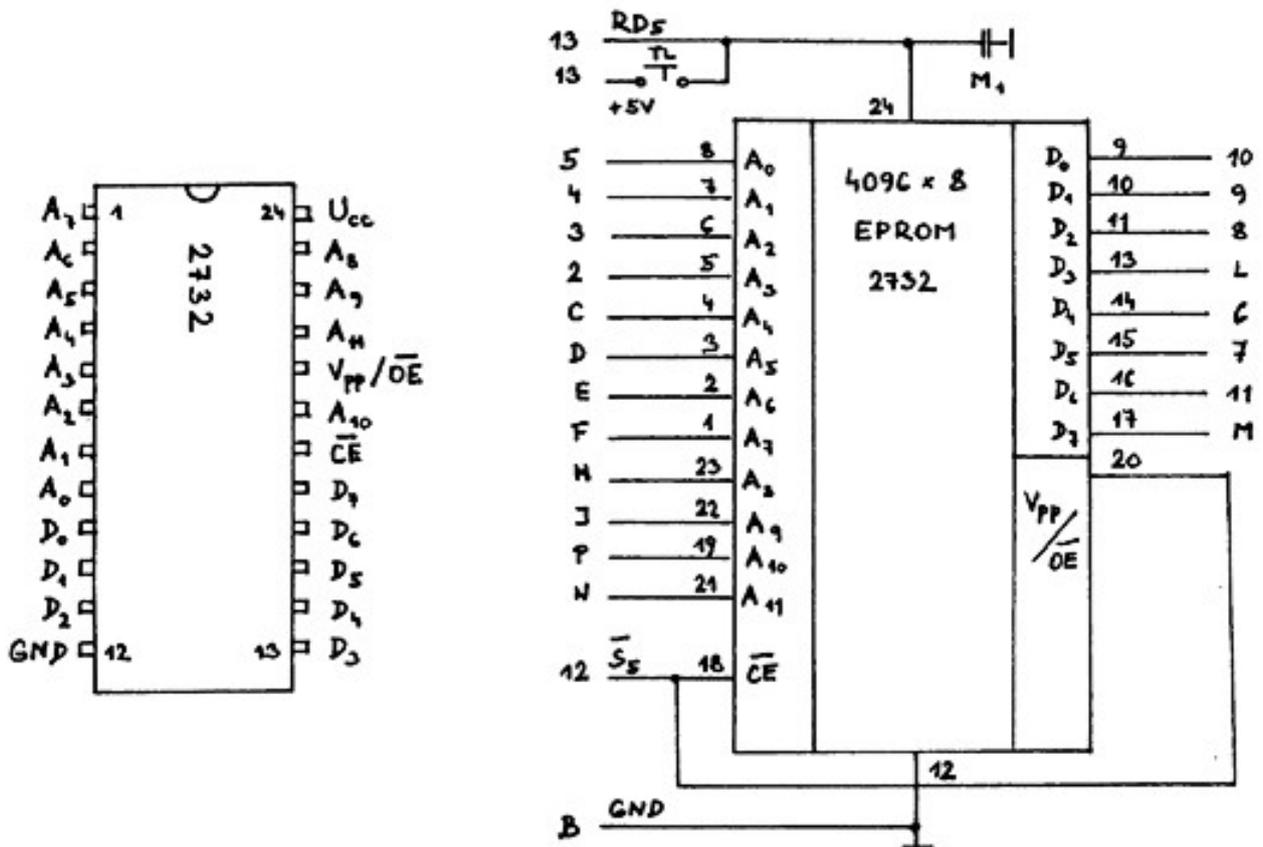
Paměti EPROM I 2716  
 K 573P05 SSSR  
 MHB 2716  
 U 2716 D NDR



Obr. 5. Zapojení modulu KART 2k Bytes s pamětí EPROM 2716



Obr. 3. Zapojení modulu KART 1k Bytes



Obr. 6. Zapojení modulu KART 4k Bytes

## TURBO 2000 OPERAČNÍ SYSTÉM

---

AK - Brno

© Milan Říha

Program TURBO 2000 Operační systém se nahrává zavaděčem TURBO 2000. Po nahrávce je nutno stisknout RESET.

Nabízí toto menu:

- Load T 2000 file
- Goto adress
- Run cartridge
- Save T 2000 file
- Duplicate T 2000 file
- Cartridge ON/OFF; now OFF

SELECT ITEM:

Nápisem SELECT ITEM oznamuje vždy ukončení předchozí akce a čekání na další volbu.

### Nahrávání programu v režimu TURBO 2000

L RETURN

- volba pro nahrávání do počítače

Spec: jméno RETURN

- zadat jméno programu, přesně tak, jak je uvedeno v hlavičce programu (respektovat velká a malá písmena, mezery, inverze apod.) Stačí zadat jen začátek jména. Ne zadá-li se jméno a stiskne samotné RETURN, nahraje se první nalezený program. Chceme-li vidět seznam všech nahrávek na kazetě, lze napsat neexistující jméno, kazeta je potom prohlížena a jsou vypisována všechna jména nalezených programů. Stiskne-li se v době zobrazování zaváděcích pruhů tlačítko OPTION, nahrávání se v okamžiku zápisu jména programu zastaví. Uvolněním OPTION se opět rozběhne.

Found: jméno

- hlášení o nalezeném programu.

Start:§.... , Run (Y/N)?

- po ukončení nahrávky oznamuje TURBO O.S. startovní adresu (hexadecimálně) a dotazuje se, zda se má program spustit. Odpovědí Y RETURN se program spustí, jakoukoliv jinou odpovědí je proces ukončen.

I/O Error ...

- oznamuje číslo chyby v případě neúspěšného průběhu nahrávání.

adr.error

- program TURBO O.S. chrání sám sebe, v případě, že by se měl při nahrávání poškodit, ohlásí chybu adresování.

Spuštění programu (GOTO)

G RETURN

- volba pro spuštění programu

Adress:§.... RETURN

- zadat hexadecimálně startovní adresu

Spuštění BASIC,interpreteru z ROM

R RETURN

- zajistí přechod do BASICu, pokud je cartridge ve stavu ON (viz menu). Výsledek je signalizován nápisem READY
- je-li ve stavu OFF je oznámena zpráva NO CARTRIDGE

Zápis programu na kazetu v režimu TURBO 20000

S RETURN

- volba zápisu na kazetu v režimu TURBO 20000. Program musí být uložen v počítači na správných adresách (pracovních).

From:§.... RETURN

- zadat hexadecimálně adresu, od které je program uložen

To:§.... RETURN

- zadat koncovou adresu programu

Run:§.... RETURN

- zadat startovní adresu programu

Name:..... RETURN

- zadat jméno programu (maximálně 10 znaků)

RETURN

- spustí zápis programu na kazetu.

Kopírování programu v režimu TURBO 20000

D RETURN

- volba pro kopírování v režimu TURBO 20000

Spec: jméno RETURN

- platí stejná pravidla jako pro LOAD

Found: jméno

- oznamuje jméno nalezeného programu.

File O.K.

- oznamuje úspěšné ukončení nahrávky do paměti počítače a dvakrát zahouká. Po vložení prázdné kazety stisknout RECORD + PLAY a libovolnou klávesu na počítači.

I/O Error ...

- oznamuje číslo chyby při vstup/výstupní operaci

### Zapnutí nebo vypnutí interpreteru BASIC

C RETURN

- změní stav zapnuto na vypnuto a naopak. Pro informaci je v menu uvedeno

now OFF - nyní vypnuto nebo

now ON - nyní zapnuto

Init (Y/N)

- operační systém se dotazuje, zda se má současně inicializovat (Y=ano) nebo nikoliv. Podle volby se provede nebo neprovede inicializace ukazovátek dat BASICu.

### Další informace o TURBO O.S.

TURBO 2000 Operační systém doplňuje zařízení "T:" do tabulky vstup/výstupních zařízení. Toto zařízení lze používat namísto "C:" v příkazech OPEN, LIST, SAVE, ENTER a LOAD. Vstup/výstupní operace jsou potom prováděny nikoli v blocích 128 bytových, ale 1024 bytových a rychlostí 2000 baudů.

TURBO 2000 Operační systém doplňuje zařízení "M:" do tabulky vstup/výstupních zařízení. Jde o RAMDISK o velikosti 14 kB. Lze do něj ukládat pomocí příkazů LIST, SAVE, PRINT a číst pomocí příkazů ENTER, LOAD a INPUT. RAMDISK je umístěn pod pamětí ROM počítače.

Přechod z BASICu do TURBO O.S. lze provést bez vypnutí počítače a bez ztráty dat v paměti:

DOS RETURN

- jestliže není zaveden tzv. TURBO-DOS (obsahující menu LOAD SAVE VERIFY a QUIT) nebo

X=USR(2816) pro BASIC

USR(§B00) pro TURBOBASIC

Přechod do TURBO O.S. z monitorovacích programů (ATMAS II, DEBUG 2, ATADIS, ASSEMBLER, ...), které umožňují odstartování programu od dané adresy, se provede zadáním adresy §B00 (nebo 2816 decimálně).

POZOR !!! Po inicializaci TURBO O.S. stisknout RESET!

Program TURBO O.S. je uložen od adresy §B00 (2816) a je dlouhý §881 (2178) byte. Zavádí se běžným zavaděčem TURBO 2000. Ukládací adresa zavaděče je §05AC (1452), délka §01FF (511) a startovní adresa §05BA (1466).

Autor programu: Milan Říha

Návod zpracoval: Ing. Petr Válka

K Y A N P A S C A L 800 XL - TURBO 2000

TURBO 2000 - zaváděč	SO
TURBO 2000 Operační systém	T
PASCAL EDITOR V 1.1	TOS-ED.PAS
PASCAL COMPILER	TOS-PC.PAS
PASCAL LIBRARY	TOS-LIB.PAS

Návod na použití:

- 1) Nahrát zaváděč TURBO 2000
- 2) Nahrát TURBO 2000 Operační systém - Říha.
- 3) Volbou L vyvolat nahrávání
- 4) Specifikací ED.PAS nebo bez specifikace (samotné RETURN) nahrát Editor
- 5) Na dotaz zda startovat odpověď Y (yes)
- 6) Zvolit jméno souboru M: (jméno RAMDISKU)  
vymaže se obrazovka a je možno psát program v PASCALU
- 7) Pomocí ESC lze přejít do hlavního menu Editoru. Je-li potřeba zdrojový program nahrát na kazetu, změnit jméno na C: ne-

bo T: pomocí volby P a potom pomocí volby S nebo X nahrát. Je-li jméno M: nahraje se program na RAMDISK. Pozor ! původní obsah RAMDISKu se přepíše.

- 8) Použije-li se RESET, naskočí TURBO O.S. a zpět na posledně nahraný program (Editor nebo Compiler) je možno se dostat volbou G na adrese §2005.
- 9) Po editaci se nahraje Kompilátor z TURBO O.S. volbou L a specifikací PC.PAS nebo bez specifikace
- 10) Po natažení na dotaz zda startovat odpověď Y, objeví se nápis KYAN PASCAL COMPILER ... a znak >
- 11) Zapsat volbu

>M:-OC: M: - z RAMDISKU

C: - na kazetu ve standartu

nebo

>M:-OT: T: - na kazetu v TURBO blocích

- 12) Kompilátor vypíše seznam syntaktických chyb, je-li program v pořádku dvakrát zahouká a přeložený program se nahraje na kazetu.

Využívání programu:

- 13) Po zakončení nahrávání na kazetu se objeví opět znak >, zapsat C: nebo T: a nahrát přeložený program do počítače
- 14) Stisknout RESET a po přechodu do TURBO O.S. nahrát volbou L se specifikací LIB.PAS nebo bez specifikace Knihovnu.
- 15) Po nahrávce Knihovny na dotaz zda odstartovat odpověď N !!!
- 16) Když je v počítači strojový program i Knihovna, odstartuje se volbou G na adrese §2005
- 17) Opakování výpočtu RESET, G, §2005

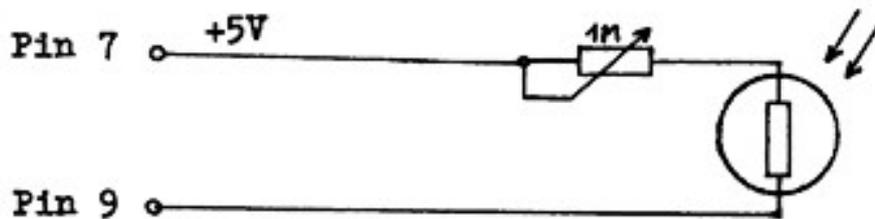
Spouštění dříve přeloženého programu:

Nahrát TURBO 2000, TURBO O.S., a např. Editor (nejkratší) a stisknout BREAK. Objeví se znak > . Dále viz. bod 13)

Pro TURBO 2000 upravil: Milan Říha (1988)

Návod zpracoval: Ing. Petr Válka (1988)

### Světelný sensor

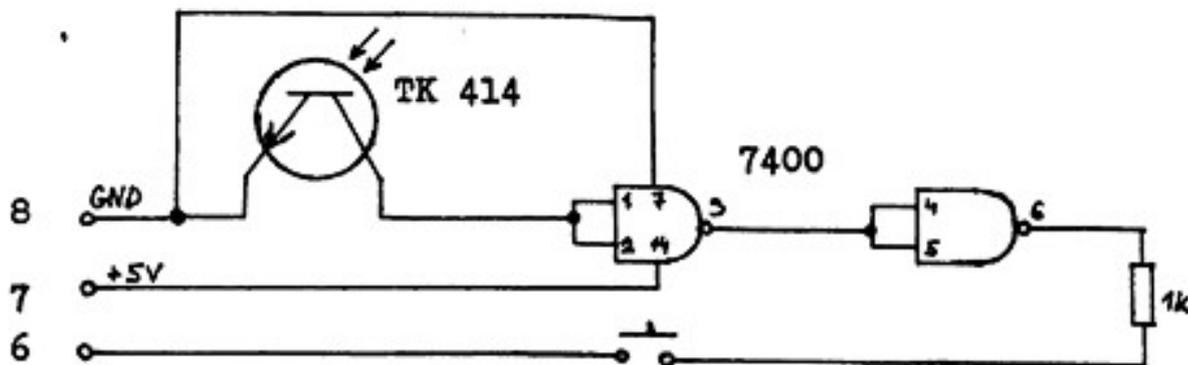


Program:

```
1Ø PRINT PADDLE(Ø)  
2Ø GOTO 1Ø
```

Při běhu programu se budou na obrazovce měnit čísla v rozmezí 1 až 228. Tato čísla jsou v přímém vztahu k množství světla dopadajícímu na fotočlánek. K vyzkoušení stačí množství dopadajícího světla regulovat např. dlaní. Při odkrývání se budou čísla zvětšovat a obráceně v rozmezí uvedeném výše.

### Světelné pero



Program:

```
05 A = 564:B = 565  
1Ø PRINT "X="; PEEK(A)"Y="; PEEK(B)  
2Ø GOTO 1Ø
```

Pobýhujte světelným perem po obrazovce a držte stisknutý mikrospínač. Čísla Vám budou udávat souřadnice bodu Vašeho světelného pera.

Upozornění: ke správné reakci světelného pera je nutné upravit kontrast monitoru.

Váš počítač využívá vývod 6 (pin 6) ke kontrole běhu paprsku televizního monitoru. Normálně je tento vývod na úrovni log. 1. Obvod světelného pera je navržen tak, aby ve vhodný okamžik uvedl tento vývod na log. 0. K tomu dojde pokud paprsek monitoru prochá-

zí právě pod světelným perem. Aby počítač tento jev zaznamenal je ovšem nutné stisknout mikrospínač. Uvedený program je zatížen touto chybou: ačkoliv je světelné pero umístěno do jednoho bodu, může se udávaná souřadnice měnit okolo střední hodnoty. Tomuto jevu lze zabránit opětovným mačkáním mikrospínače nebo následujícím programem.

Program Vám ukáže použitelnost Vašeho pera k uskutečnění výběru z určité nabídky. K funkci stačí přiložit světelné pero k hvězdičce u nápisu a stisknout mikrospínač. Monitor se smaže a objeví se Vámi zvolený text. Zakrátko se objeví POKRAČOVÁNÍ s hvězdičkou před slovem. Pokud opět k této hvězdičce přiložíte pero a zmáčknete mikrospínač - bude Vám opět nabídnut výběr.

Program:

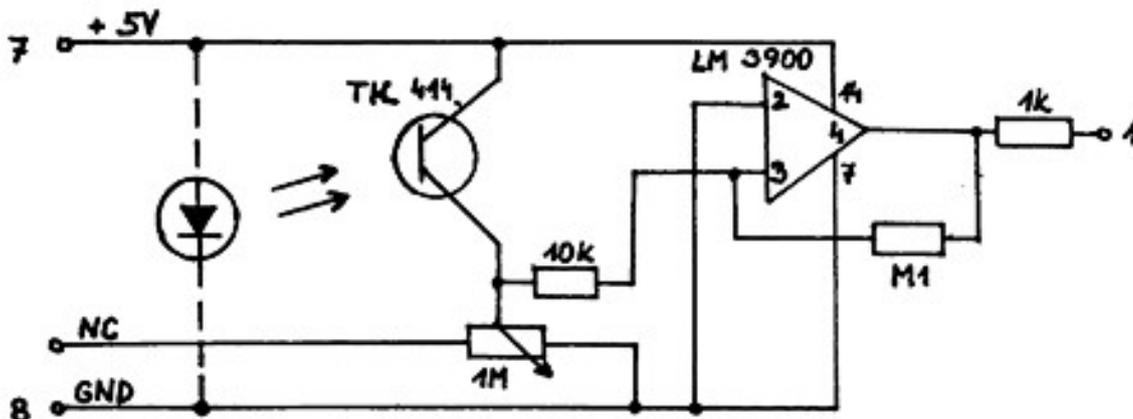
```
100 REM UKAZKA PRACE SVETELNEHO PERA
150 PRINT CHR$(125)
160 PRINT "TRI VOLBY"
170 PRINT : PRINT : PRINT
180 PRINT " * VOLBA PRVNI"
190 PRINT : PRINT : PRINT
200 PRINT " * VOLBA DRUHA"
210 PRINT : PRINT : PRINT
220 PRINT " * VOLBA Treti"
230 A=PEEK(564): B=PEEK(565)
240 IF A<90 THEN 230
250 IF (B>25) AND (B<35) THEN 300
260 IF (B>45) AND (B<55) THEN 330
270 IF (B>60) AND (B<70) THEN 360
290 GOTO 230
300 PRINT CHR$(125)
310 PRINT "VOLBA PRVNI"
320 GOTO 400
330 PRINT CHR$(125)
340 PRINT "VOLBA DRUHA"
350 GOTO 400
360 PRINT CHR$(125)
370 PRINT "VOLBA Treti"
400 FOR I=1 TO 15: PRINT : NEXT I
410 PRINT
```

```
42Ø PRINT " * POKRACOVANI"  
43Ø IF PEEK (564)<9Ø AND PEEK (565)>8Ø THEN 15Ø  
44Ø GOTO 43Ø
```

### Digitální světelný sensor

Světelné pero, které jste postavili indikuje světelný signál z monitoru počítače. Co ale v případě, kdy chcete vědět zda je např. venku tma nebo kdy Vám někdo otevřel dveře Vaší místnosti? Případně budete chtít postavit elektronické stopky, které vyhodnotí dobu přerušení světelného paprsku. K indikaci různých světelných veličin a jejich malých změn potřebujete dokonalejší sensor.

Dále následuje návod na stavbu univerzálního digitálního světelného sensoru. Tento sensor komunikuje s počítačem logickými úrovněmi napětí. Pokud je nastaven na určitou hodnotu osvětlení a tato - případně hodnota vyšší - trvá, je výstupní informace sensoru log. 0. Pokud úroveň osvětlení klesne, přejde výstupní úroveň na log. 1.



Program:

```
1Ø IF STICK(Ø)=15 THEN PRINT " OFF "  
2Ø IF STICK(Ø)=14 THEN PRINT " ON "  
3Ø GOTO 1Ø
```

Na obrazovce se objeví slova OFF (vypnuto), nebo ON (zapnuto). Zapnuto znamená, že sensor indikuje světla vyšší intenzity než odpovídá nastavení potenciometru a slovo vypnuto naopak. Toto nastavení je citlivé na nastavení potenciometru. Jistě lehce najdete bod nastavení při kterém se obě slova na monitoru neustále mění. Při indikaci např. osvětlení v místnosti nastavte potenciometr právě do oblasti tohoto rozhraní.

### Měřič času s využitím světelného paprsku

Předcházející obvod, doplněný o luminiscenční diodu (znázorněno zeleně), může být použit při závodě autíček či jiných aplikacích s měřením času pomocí světelného paprsku.

Upozornění: použijeme-li infračervenou diodu, není její záření okem viditelné a není tak umožněna vizuální kontrola funkce. Světlo infračervené diody musí při jejím zatemňování jednoznačně způsobovat změnu.

Velmi důležité je přesné nasměrování zdroje a čidla záření.

Obvod je možné využít všude tam, kde žádáme např. vyvolání poplachu při přerušení paprsku.

Program:

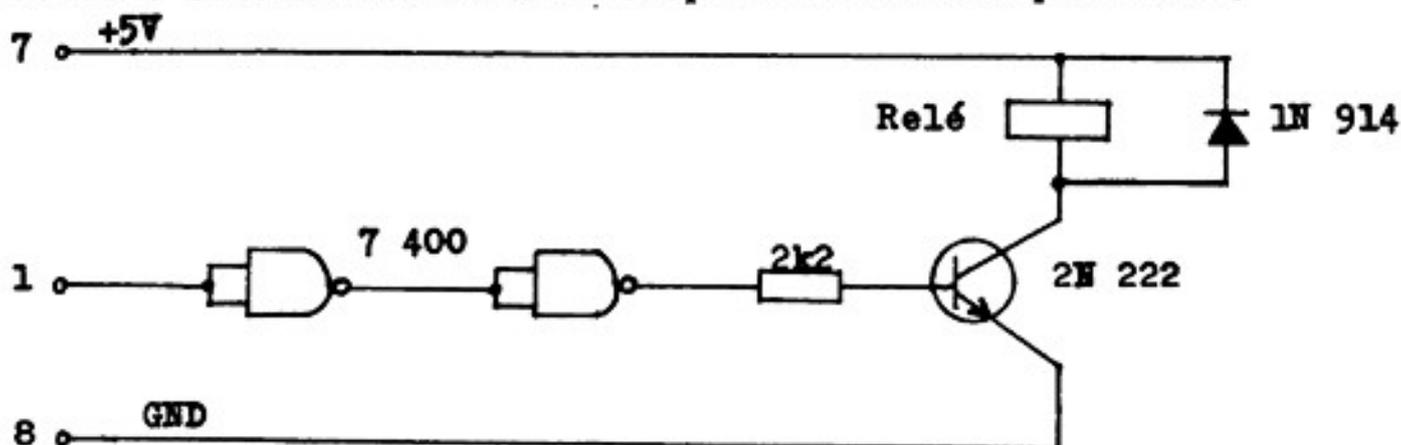
```
11Ø REM MERENI CASU
12Ø PRINT "ZASTINTE IC DIODU A NASTAVTE OBVOD"
13Ø PRINT "SENSORU DO STAVU, KDY SE MENI"
14Ø PRINT "MEZI ZAPNUTO A VYPNUTO"
15Ø PRINT "STISKNETE X CHCETE LI POKRACOVAT"
155 FOR J=1 TO 25ØØ : NEXT J
17Ø IF STICK(Ø)=15 THEN PRINT "VYPNUTO"
175 IF STICK(Ø)=14 THEN PRINT "ZAPNUTO"
18Ø REM KONTROLA STLACENI X
19Ø IF PEEK(764)=22 THEN 215
21Ø GOTO 17Ø
212 REM VYMAZANI PREDCHAZEJICIHO STISKU
215 POKE 764,255
22Ø PRINT "PRERUSENI PAPRSKU SPUSTI MERENI"
23Ø IF STICK(Ø)=14 THEN 23Ø
24Ø POKE 18,Ø: POKE 19,Ø: POKE 2Ø,Ø
25Ø IF STICK (Ø)=15 THEN 25Ø
26Ø IF STICK (Ø)=14 THEN 26Ø
27Ø PRINT "CAS PRERUSENI"; ((PEEK(18)*255*255)
+ (PEEK(19)*255) + (PEEK(2Ø)))/6Ø; "SEKUND"
```

od

### Elektronický spínač

Všechna doposud realizovaná zapojení využívala senzory k přenášení informace z okolního prostředí do počítače. Váš počítač

tač však může pomocí elektrických signálů sám ovládat vnější zařízení. Během využívání vstupních portů jste se dozvěděli, jak lze pomocí signálů v elektronické formě zapnuto/vypnuto ovládat datové I/O linky. Ty samé vodiče mohou však být využity pro výstup signálů z počítače. Jakým způsobem lze výstupní signál z počítače využít, ukáže následující obvod, který si můžete postavit. Vnější elektronické obvody musí zpracovat signál v digitální formě a vhodně jej využít. Pro příklad tohoto typu obvodů uvádíme schéma elektronického spínače řízeného počítačem.



Program:

```

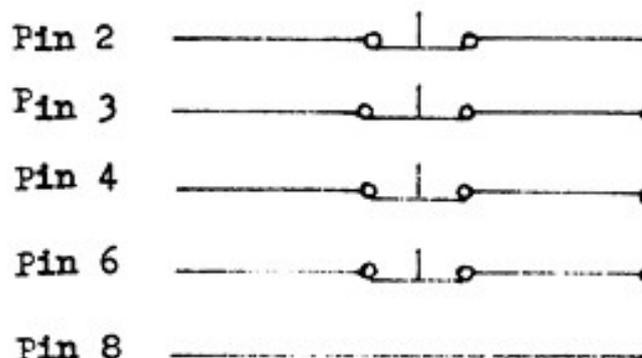
100 REM CASOVY ALARM
120 PRINT "ZADEJTE HODINY";
130 INPUT A
140 PRINT "ZADEJTE MINUTY";
150 INPUT B
160 PRINT "ZADEJTE SEKUNDY";
170 INPUT C
180 D=(A*216000) + (B*3600) + (C*60)
190 POKE 18,0: POKE 19,0: POKE 20,0
200 IF ((PEEK(18)*255*255) + (PEEK(19)*255) + (PEEK(20)))> D
    THEN 220
210 GOTO 200
215 REM NASTAVENI DATOVYCH LINEK NA VYSTUP
220 POKE 54018,48
225 POKE 54016, 255
230 POKE 54018,52
235 REM NASTAVENI DATOVE LINKY PRO SPUSTENI ALARMU
240 POKE 54016,0
    
```

```
245 POKE 18,0: POKE 19,0: POKE 20,0
250 IF ((PEEK(18)*255*255) + (PEEK(19)*255) + (PEEK(20))) > 300
    THEN 270
260 GOTO 250
265 REM NASTAVENI DATOVE LINKY PRO VYPNUTI ALARMU
270 POKE 54016,255
280 REM NASTAVENI DATOVYCH LINEK NA VSTUP
290 POKE 54018,48
300 POKE 54016,0
310 POKE 54018,52
```

Toto využití elektronického spínače je pouze jedním z příkladů použití. Jednoduché zapojení a možnost připojení spotřebičů s větším odběrem (díky použitému relé) dávají možnost dalších aplikací. Při použití síťového napětí je nutná dokonalá ochrana.

### Poplachová signalizace

S výhodou použijeme znalostí získaných v předcházející části. Snímací senzory mohou být různé, od napjatého kovového drátku, který se přetrhne v případě nechtěného pohybu, k magnetickému spínači, indikátoru vibrace k senzoru ultrazvukovému a infračervenému.



Program:

```
110 REM POPLACHOVA SIGNALIZACE
120 PRINT "PRI ZAKRYTEM ZDROJI SVETLA NASTAVTE
130 PRINT "POTENCJOMETREM OBVOD DO STAVU, KDY"
140 PRINT "SE MENI MEZI VYPNUTO A ZAPNUTO"
150 PRINT "STISKNETE X CHCETE-LI POKRACOVAT"
155 FOR J=1 TO 2500: NEXT J
160 IF STICK(1)=1 THEN PRINT "VYPNUTO"
170 IF STICK(1)=0 THEN PRINT "ZAPNUTO"
```

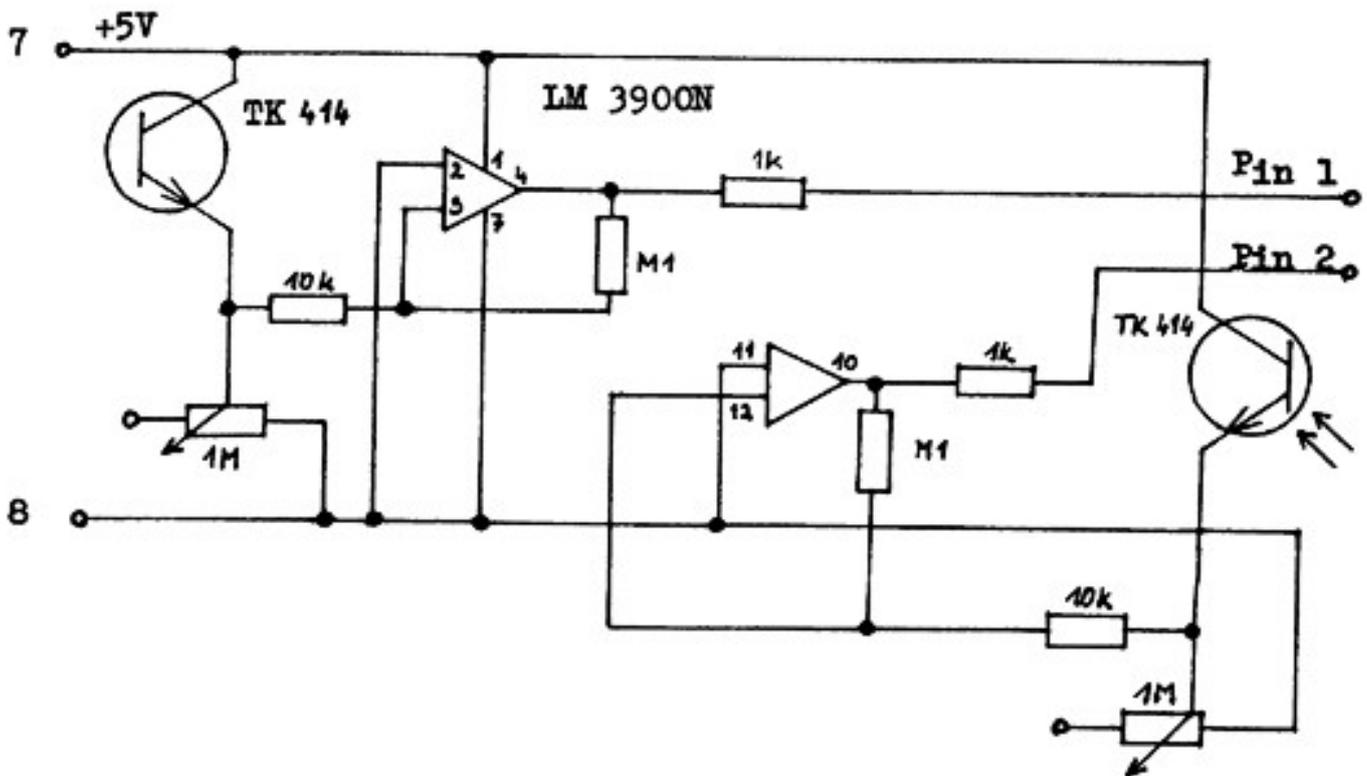
```
18Ø REM KONTROLA STISKU X
19Ø IF PEEK (764)=22 THEN 21Ø
2ØØ GOTO 16Ø
21Ø REM VYČISTENI PREDCHAZEJICIHO STISKU
22Ø POKE 764,255
23Ø PRINT "SEPNUTI SENZORU SPUSTI SIGNALIZACI"
24Ø A=STICK(1): B=STRIG(1)
25Ø IF (A=Ø AND B=Ø) THEN 24Ø
26Ø PRINT "POZOR - SIGNALIZACE SPUSTENA"
27Ø IF A=1 THEN PRINT "SENSOR 1"
28Ø IF A=2 THEN PRINT "SENSOR 2"
29Ø IF A=4 THEN PRINT "SENSOR 3"
292 IF A=8 THEN PRINT "SENSOR 4"
294 IF B=1 THEN PRINT "SENSOR 5"
3ØØ PRINT "DETEKCE NARUSENI"
3Ø5 REM INICIALIZACE HODIN
31Ø POKE 18,Ø: POKE 19,Ø: POKE 2Ø,Ø
32Ø REM NASTAVENI DATOVYCH LINEK NA VYSTUP
33Ø POKE 54Ø18,48
34Ø POKE
35Ø POKE 54Ø18,52
36Ø REM NASTAVENI DATOVYCH LINEK PRO SPUSTENI ALARMU
37Ø POKE 54Ø16,Ø
38Ø IF ((PEEK(18)*255*255) + (PEEK(19)*255) + (PEEK(2Ø)))>
    36ØØ/24 THEN 4ØØ
39Ø GOTO 38Ø
4ØØ REM NASTAVENI DATOVE LINKY PRO VYPNUTI ALARMU
41Ø POKE 54Ø16,255
42Ø REM NASTAVENI DATOVYCH LINEK NA VSTUP
43Ø POKE 54Ø18,48
44Ø POKE 54Ø16,Ø
45Ø POKE 54Ø18,52
```

### Dvoupaprskový digitální časovač

Obvod začíná měřit s přerušením prvního světelného paprsku a končí s přerušením svitu druhého světelného paprsku. Je zvláště vhodný pro vyhodnocování závodů, popř. k měření času potřebného k přemístění libovolného předmětu z bodu A do bodu B.

Program:

```
110 REM DVOJITY SVETELNY CASOVAC
112 REM NASTAVENI OBOU SENZORU
115 FOR I=0 TO 1
120 PRINT "PRI ZAKRYTEM ZDROJI SVETLA NASTAVTE"
130 PRINT "POTENCIOMETREM"; I+1; "OBVOD DO STAVU KDY"
140 PRINT "SE MENI MEZI VYPNUTO A ZAPNUTO"
150 PRINT: PRINT "STISKNETE X CHCETE-LI POKRACOVAT"
160 FOR J=1 TO 1000: NEXT J
161 A=STICK(0)
162 IF I=1 THEN 174
164 IF A=15 THEN PRINT "VYPNUTO"
166 IF A=13 THEN PRINT "VYPNUTO"
168 IF A=14 THEN PRINT "ZAPNUTO"
170 IF A=12 THEN PRINT "ZAPNUTO"
172 GOTO 190
174 IF A=15 THEN PRINT "VYPNUTO"
176 IF A=14 THEN PRINT "VYPNUTO"
178 IF A=13 THEN PRINT "ZAPNUTO"
180 IF A=12 THEN PRINT "ZAPNUTO"
190 REM KONTROLA STISKU X
200 IF PEEK (764)=22 THEN 220
210 GOTO 161
220 REM VYMAZANI PREDCHAZEJICHO STISKU
230 POKE 764,255
240 NEXT I
250 PRINT "PRERUSENI 1. PAPRSKU SPUSTI CASOVAC"
260 REM OSETRENI PRVNIHO SENSORU
270 IF STICK(0)<>13 THEN 270
280 REM START CASU
290 POKE 18,0: POKE 19,0: POKE 20,0
300 REM OSETRENI DRUHEHO SENSORU
310 IF STICK(0)=14 THEN 320
315 GOTO 310
320 PRINT "CAS JE";((PEEK(18)*255*255) + (PEEK(19)*255) +
    (PEEK(20)))/60; "SEKUND"
```



### Světelný spínač

Tento obvod sdružuje světelný sensor a elektronický spínač. Světelný sensor může detekovat úroveň venkovního osvětlení a pomocí relé může reagovat na měnící se světelné podmínky. Světelný sensor s fotoodporem CdS se zapojí mezi vývody 9 a 5 (pin počítače). Použité relé musí odpovídat spínanému zařízení. Obvod dovoluje kontrolovat Vámi zvolené světelné úrovně. Např. při umístění v okně bude reagovat na venkovní světlo. Po prvotním nastavení vyčkejte, až se sensor bude nacházet v požadované hladině osvětlení a stiskněte X. Počítač si tuto hodnotu zapamatuje a bude ji porovnávat s okamžitou hodnotou osvětlení a reagovat na ni např. zapnutím světla. Pozor, aby ovládané světlo nedopádkovalo na sensor.

Program:

```

100 REM SVETELNY SPINAC
120 PRINT "NASTAVENI HODNOTY ZAPNUTO"
130 PRINT "STISKNETE X CHCETE-LI POKRACOVAT"
140 A=PADDLE(0)
150 REM KONTROLA STISKU X
160 IF PEEK (764)=22 THEN 180
170 GOTO 140
175 REM VYCISTENI PREDCHAZEJICHO
    
```

```
18Ø PRINT "NASTAVENI HODNOTY VYPNUTO"  
    POKE 764,255  
20Ø PRINT "STISKNETE X CHCETE-LI POKRACOVAT"  
21Ø B=PADDLE(Ø)  
22Ø REM KONTROLA STISKU X  
23Ø IF PEEK (764)=22 THEN 26Ø  
24Ø GOTO 21Ø  
25Ø REM VYCISTENI PREDCHAZEJICHO  
26Ø POKE 764,255  
262 PRINT "STISKEM X STARTUJE PROGRAM"  
264 IF PEEK (764)=22 THEN 268  
266 GOTO 264  
268 POKE 764,255  
27Ø REM TESTOVANI NIZSI UROVNE SVETLA - PRVNI NASTAVENI  
28Ø IF PADDLE(Ø)<A THEN 28Ø  
29Ø REM NASTAVENI DATOVEHO REGISTRU NA VYSTUP  
30Ø POKE 54Ø18,48  
31Ø POKE 54Ø16,255  
32Ø POKE 54Ø18,52  
33Ø REM NASTAVENI DATOVE LINKY NA NIZKOU UROVEN  
34Ø POKE 54Ø16,Ø  
35Ø REM TESTOVANI VYSSI UROVNE SVETLA - DRUHE NASTAVENI  
37Ø IF PADDLE(Ø)>B THEN 37Ø  
38Ø REM NASTAVENI DATOVE LINKY NA VYSOKOU UROVEN  
39Ø POKE 54Ø16,255  
40Ø REM VYNULOVANI DATOVE LINKY PRO VSTUP  
41Ø POKE 54Ø18,48  
42Ø POKE 54Ø16,Ø  
43Ø POKE 54Ø18,52
```

## TURBO BASIC - rozdíly v detailech

-----

(Podle Happy Computer zpracoval ing. Emil Belluš)  
AK Rožnov pod Radhoštěm

Neběží Váš program při použití TURBO BASIC XL? Některé změny umožní používat i Váš normální ATARI - BASIC program.

TURBO BASIC XL je podstatně rychlejší než zabudovaný ATARI BASIC. Vysoká rychlost je ve většině případů žádoucí. Ovšem vznikají při tom i určité problémy. Často jsou v ATARI BASICu napsané herní programy v TURBO - BASIC XL těžko přehrávatelné, protože enormní rychlost nedává hráči dostatek času pro reakci. Pomoc: Zpomalování ve formě čekacích smyček, použití speciálního příkazu PAUSE, nebo zabudování dodatečných zvukových efektů.

V ATARI - BASICu je realizovatelné krátké zpoždění též pomocí příkazu

$Q = 1 \text{ na } 1.$

V TURBO BASIC XL je ale tento výpočet podstatně rychlejší než v ATARI BASICu. Umocňování s celočíselnými exponenty menšími než sto je prováděno opakovaným násobením a ne jako v ATARI - BASICu podle rovnice

$a \text{ na } b = \exp(b + \log(a)).$

Nejjednodušší je příkaz  $Q = 1 \text{ na } 1$  nahradit příkazem PAUSE 9. Tím je dosažen stejný čas zpoždění a ještě se ušetří část paměti.

Zpoždovací smyčky s FOR NEXT jsou v ATARI BASICu velmi pomalé, obzvláště, když jsou na konci programu. Nejlépe je tyto smyčky taktéž nahradit повеlem PAUSE.

### Zakázané příkazy

TURBO - BASIC XL obsahuje podstatně víc příkazů, než ATARI BASIC, proto mohou vzniknout problémy s názvy proměnných. Názvy jako ERR, HEX\$, nebo DEL(ta) jsou v ATARI - BASICu přípustné, v TURBO - BASIC XL ne (jedná se o rezervovaná slova). Při nahrávání ATARI BASIC programu není tento rozdíl pozorovatelný. Až když se pokusíte takový program s těmito proměnnými editovat, objeví se chybové hlášení, a nebo špatně přeložené řádky: Při LET ERR = 100 bude použita proměnná ERR, ale při A = ERR funkce

"ERR". Potom zůstává jediná možnost, názvy proměnných změnit. Při tom pomáhá užití příkazu DUMP, kterým je možno provést listiny použitých proměnných.

Po nahrání postiženého programu příkazem LOAD zadáte příkaz DUMP a na obrazovce se objeví všechny proměnné. Pokud máte tiskárnu, lze je vytisknout повеlem DUMP "P:". Vypište si při tom všechny proměnné, které musíte změnit. Při tom si však neušetříte kontrolu každého programového řádku a jeho eventuální úpravu. Zaznamenejte potom program pomocí LIST "D: XXXXXX" na disketu nebo LIST "C:" na kazetu. Zadejte NEW a potom ENTER "D:XXXXXX" případně ENTER "C:". Při tomto pořadí příkazů bude tabulka přepsána. Nepoužité proměnné nezabírají tedy více místo v paměti. Příkazem DUMP můžete znovu zkontrolovat, zda jste nepřehlédli některé proměnné při opravách. Když jste provedli všechny korekce, zabezpečte program příkazem SAVE na disketu či kazetu a spusťte kontrolní běh programu.

#### Poznámka:

U verze TURBOBASIC 1.5X lze se stejným účinkem použít příkazy ENTER "T:" a LIST "T:" , které zapisují a čtou z kazety ve formátu ZX (Turbo 2000).

TURBO - BASIC XL není umístěn v ROM paměti, ale nahrává se z diskety nebo z kazety. Protože je ještě rozsáhlejší než ATARI BASIC (18 kByte místo 8 kByte), je většina interpreteru zabudovaná v OS-ROM (12 kByte). Zbývajících 6 kByte se nalézá na spodní hranici paměti přímo nad DOS.

Původně byl TURBO - BASIC XL určen pro ATARI 400 s 48 kByte RAM. V této verzi se nalézal celý interpreter nad DOS. Tak ovšem zůstalo jen 20 kByte pro BASIC programy. 800 XL obsahuje však celkem 64 kByte RAM, která TURBO - BASIC XL též plně využívá. Z přidanych 14 kByte jsou ještě 2 kByte obsazeny kopiemi obvodů ROM - znakových generátorů. Je to opatření, které zabraňuje poruchám obrazu při zobrazení textu, okamžitě po odpojení ROM. TURBO - BASIC XL je tím ale podstatně rozsáhlejší, než aby se dal zabudovat do zbývajících 12 kByte. Proto není možné se vyhnout tomu, aby jednotlivé programové části byly umístěny do oblastí, které jsou normálně rezervovány pro BASICové programy. Tím je ale umožněno použití podprogramů v ROM. Dále to umožňuje

ještě obsluhovat přerušeni.

Část TURBO - BASIC XL zasahuje do oblasti BASICu na adresách \$2080 až \$3629 (hexadecimálně). Jedná se tedy o oblast, která je bezprostředně nad DOS 2.5 při pěti instalovaných disketových přístupech (D1: až D4: a RAM - disku D8: na 130 KE) "Fillepufern". POKE v této oblasti vede často k zhroucení programu, ale co je horší, taky k špatnému výkladu interpreteru. což vede k vadným hodnotám výpočtu.

Po dobu provádění příkazů POKE, DPOKE, MOVE a -MOVE je po každé připojena ROM. Proto nemohou tyto příkazy v "ROM" části TURBO - BASIC XL způsobit žádné škody. Mimo to je ROM aktivována v příkazech PEEK, DPEEK, všech I/O příkazech, PRINT, INPUT, BGET, BPUT apod. Neplatí to však při PAINT a při volání aritmetických rutin (jsou použité některé hodnoty z rutiny pohyblivé čárky STR\$ a VAL).

### Pozor přerušeni !

Je-li připojena ROM, je přerušeni obslouženo obvyklým způsobem. Jestliže je ROM odpojena, odskočí obsluha přerušeni nejprve do spodní části oblasti RAM interpreteru. Tam je nejdříve zapojena ROM. Poté co je přerušeni obslouženo, je ROM znovu odpojena. Z tohoto důvodu trvají rutiny přerušeni několik  $\mu$ sec déle. To ovšem neplatí pro display-list-interrupt (DLI), protože ten musí proběhnout paralelně s tvorbou obrazu. Přerušeni je v tomto případě už rozeznané interpreterem, takže je okamžitě provedeno. Musí probíhat i při odpojené ROM.

Oblast paměti je pro programy v BASICu, data a pro zásobník v oblasti do adresy \$A000 (40960), v TURBO - BASIC XL ale až do \$C000 (49152). BASIC ROM je nepoužita a může být trvale odpojena. Čisté BASIC programy bez USB příkazů a POKE nejsou tímto postiženy; kritické jsou příkazy POKE a USB v oblasti adres 30000 až 40960.

Pokud použijete program s příkazem např. "POKE 39994,4" může program v TURBO - BASICu XL proběhnout jen po určitém způsobení. Tento povel mění display list. V tomto případě se jedná o strojový jazyk videoprocesoru zvaného ANTIC.

Protože existují ještě 600 XL se 16 kByte a ATARI 400 a 800 s 16,32 a 48 kByte RAM, existují též programy, které jsou přizpůsobené na různé rozsahy paměti. K tomu používá operační systém (OS) ATARI různé ukazovatele, o které se BASIC opírá. Nejdůležitější je RAMTOP, to je paměťová buňka 106 (\$6A). Tato buňka obsahuje MSB (Most Significant Byte) t.j. první buňka nepoužitelná pro program v paměti. To znamená, že po příkazu "PEEK (106)\*256" obdržíme hodnotu, která je o jedničku vyšší, než nejvyšší RAM adresa v ATARI BASIC daná "PEEK (106)" tedy 160 (\$A0) v TURBO - BASIC XL 192 (\$C0). Příkazem "POKE 106, PEEK (106)-16" lze 4096 bytů (16\*256) chránit před zásahem BASICu nebo OS.

Nakonec je potřebný ještě jeden GRAPHICS příkaz k přizpůsobení ostatních ukazovatelů (pointerů), aby došlo k posuvu paměti obrazu. Příkazem "POKE 106, 160: GRAPHICS 0" na začátku programu je simulováno uspořádání paměti v ATARI BASIC. Zůstane ovšem nevyužitých 8 kByte paměti pro BASIC program. Zato začíná obrazová paměť už na adrese 40000 místo 48192. Pro krátké programy to plně vyhovuje a je to nejkratší způsob přizpůsobení.

Pro dlouhé programy, jako je např. "Magic Painter", jsou možné dvě možnosti:

1) Všechny POKE a USR adresy zvýšit o 8192. To se dá provést jen u přesouvateľných dat a strojových programů. Bohužel jsou to velmi zřídkaivé případy. Když nastanou, zapisuje je programátor do textových řetězců (příznak toho je "USR(ADR(X+),...)"). Toto volání nezpůsobuje žádné těžkosti. Můžete přirozeně strojové programy přemístit, tedy přizpůsobit jiným adresám. To ale vyžaduje znalosti assembleru a hodně času a pracnosti. Jednodušší je, když je k dispozici zdrojový text programu.

### Zakázané adresy

Na problémy narazíme také v datech pro grafiku hráč - střela, aneb při změněných grafických znacích. V tomto případě je nutné POKE příkazy určené pro OS anebo hardware přizpůsobit. Jedná se nejčastěji o POKE na adresách 756, 54279, 561, 89 (a 513 atd.). Hodnoty v POKE je nutno zvýšit o 32. Někdy jsou tyto adresy použity ve vnitřních strojových programech.

Vidíte, že tato přizpůsobení bez dokonalé znalosti programů

a hardware jsou prakticky neproveditelná.

2) Program o 8 kByte zkrátit, to je při velkoryse komentovaných programech dosažitelné často přímým vymazáním REM řádků. Často pomůže vyčlenit DATA do podprogramu, v němž je nahrajeme zvlášť smyčkou READ, POKE a potom hlavní program startujeme `RUN"D:XXX.XXX"`. Rychlejší je ale příslušný BLOAD nebo BGET příkaz, pokud jsou data v odpovídajícím formátu na disketě. To navíc šetří i cenné místo na disketě. Dále je možné také šikovným nasazením rozšířených TURBO - BASIC XL příkazů ušetřit paměť.

Kombinovaným nasazením těchto prostředků je možné přizpůsobit i program "Magic Painter". Zvlášť užitečné jsou při tom povely BLOAD, CIRCLE a TEXT. Ještě jeden typ: je-li "Magic Painter" napsán v TURBO - BASIC XL, může být bez dalšího používáno grafiky 15 + 16 s vysokou rozlišovací schopností místo grafiky 7 + 16. Navíc je v TURBO - BASIC XL k dispozici o 8 kByte RAM paměti více.

adresa	ATARI BASIC	adresa	TURBO - BASIC XL
FFFF	OS - ROM	FFFF	OS-ROM / INTERPRETER
D800	I/O - CHIPS	D800	I/O - CHIPS
D000	OS - ROM	D000	OS-ROM / INTERPRETER
C000	BASIC - ROM		GRAFIKA - RAM
A000	GRAFIKA - RAM		
			BASIC - ZÁSOBNÍK
	BASIC-ZÁSOBNÍK		ŘETĚZCE
	ŘETĚZCE	3629	PROGRAM
2080	PROGRAM	2080	INTERPRETER
	DOS 2.5		DOS 2.5
0700		0700	
0000	OS - PROMĚNNÉ	0000	OS - PROMĚNNÉ

Porovnání uspořádání paměti v TURBO - BASIC XL a ATARI BASIC.

## Uspořádání sektorů na disku operačního systému DOS 2.5

(Z ANALOG COMPUTING č. 56 vybral a přeložil Ing. Outlý Martin)

Informace na jedné straně diskety jsou uloženy na 40 soustředných stopách.

Jednoduchá hustota záznamu rozděluje každou stopu na 18 sektorů a na každý z nich ukládá 128 bytů. Poslední sektor není využit, máme tedy k dispozici 719 využitelných sektorů. Dvojitá hustota záznamu rozděluje disketu stejným způsobem, avšak na každý sektor ukládá 256 bytů. Rozšířená hustota rozděluje každou stopu na 26 menších sektorů a na každý ukládá 128 bytů. Protože DOS může počítat jen do 1023, posledních 16 sektorů je nevyužito. Sektor je blok dat, které jsou psány nebo čteny z diskety najednou.

Většina z těchto sektorů je využita na data, avšak při formátování diskety jsou některé sektory rezervovány pro zvláštní účely: Sektory 1, 2 a 3 jsou tzv. BOOT sektory. Když zapisujete DOS - soubory na disketu, jsou tyto sektory také popsány, ačkoliv ve skutečnosti nejsou částí těchto souborů. Jsou použity pro to, aby sdělily počítači jak natáhnout DOS - soubory do paměti při zapnutí počítače.

Tyto 3 sektory jsou vždy psány v jednoduché hustotě a to i v případě, že zbývající sektory jsou v hustotě dvojitě.

Sektor 360 je rezervován pro "tabulku obsahu" (VTOC), která obsahuje informace, které sektory jsou již použité a které jsou ještě volné. Byty 10 až 99 tohoto sektoru indikují stav sektorů 0-719 (1bit pro každý sektor). Pokud je bit nastaven na 1, odpovídající sektor je volný, pokud je nastaven na 0, sektor je použit. Sektor 0 neexistuje, jemu odpovídající bit je vždy nastaven na 0.

Byte 0 sektoru 360 indikuje typ DOSu, byty 1 a 2 maximální počet volných sektorů, byty 3 a 4 počet běžně volných sektorů, byte 5 je návěst požadavku čtení (WRITE REQUIRED FLAG) a byty 6,7,8 a 9 nejsou použity.

Sektor 1024 používá DOS 2.5 jako rozšíření VTOC pro označení dostupnosti sektorů 720 až 1023. Sektory 361 až 368 jsou rezervovány pro informace o adresáři všech souborů na disketě. Zápis každého souboru do adresáře obsazuje v těchto adresářových sektorech 16 bytů a proto každý ze sektorů 361 až 368 může obsahovat maxi-

málně 8 zápisů. U dvojité hustoty je tomu také tak, proto druhá polovina každého takového sektoru je prázdná. 8 zápisů do adresáře krát 8 adresářových sektorů dává dohromady maximální limit 64 souborů na disketě. Každý zápis do adresáře je organizován následovně:

byte 0	.....	stavový byte
byty 1 a 2	.....	délka souboru (v počtu sektorů)
byty 3 a 4	.....	počáteční sektor souboru
byty 5 - 15	.....	jméno souboru a extender

Stavový byte se dále člení na jednotlivé bity s následujícím významem:

bit 7	.....	soubor zrušen
bit 6	.....	aktivní soubor
bit 5	.....	soubor uzamčen
bity 4 - 2	.....	jsou nepoužity
bit 1	.....	soubor byl vytvořen DOS 2
bit 0	.....	soubor je otevřen pro výstup

Proto např. status byte s hodnotou 128 (§ 80) indikuje, že soubor byl zrušen, hodnota 98 (§ 62) indikuje, že aktivní soubor byl vytvořen DOS 2 a je uzamčen (64 + 2 + 32). Zápis v adresáři 66, 8, 0, 3, 1, 77, 89, 70, 73, 76, 69, 32, 32, 68, 80, 32 patří aktivnímu souboru se jménem MYFILE.

DP, který začíná na sektoru 259 (3 + 1 \* 256) a je 8 sektorů dlouhý (8 + 0 \* 256).

Zbývající sektory (707 při jednoduché a dvojité nebo 1010 při rozšířené hustotě zápisu) jsou souborové sektory. Mohou zde být BASIC programové soubory, datové soubory nebo soubory ve strojovém kódu.

Poslední 3 byty každého takového sektoru jsou rezervovány pro spojování souborových sektorů dohromady.

Nižších 6 bitů bytu 125 obsahuje číslo souboru (0 - 63) jako kontrolu úplnosti souboru. Zbývající 2 bity společně s následujícími 8-mi bity bytu 126 obsahují číslo následujícího sektoru. Těchto 10 bitů reprezentuje každé číslo mezi 0 - 1023, což vysvětluje proč DOS 2.5 používá pouze tento počet sektorů. Poslední-byte 127 informuje kolik bytů obsahuje příslušný sektor. Při jednoduché a rozšířené hustotě obsahuje každý souborový sektor 125 bytů, při

dvojité potom 253 pokud se však nejedná o poslední sektor souboru, který může být naplněn jen částečně. Pokud se jedná o poslední sektor souboru, číslo indikující následující sektor je rovnou nule.

Výše popsané poslední 3 byty jsou obsaženy pouze v souborových sektorech, v rezervovaných, systémových sektorech se nenachází.

### S T R E A M L I N E R

(Analog COMPUTING č. 56 - přeložil Ing. M. Outlý)

Zvyšuje výkonnost nahrávání strojových souborů.

Binární soubory. Pokud je tu věc, které mají Atari majitelé dost potom jsou to určitě binární soubory.

Máte velkou zásobu binárních souborů. Co však pravděpodobně nevíte je to, že mnoho těchto souborů je delších než by bylo nutné a o to se déle čtou. Podívovali jste se někdy nad tím, že čtení některých souborů je doprovázeno nepřerušným proudem zvuku, zatímco některé mají pauzy mezi každým blokem? Toto není vada vašeho d rivu, ale spíše těch souborů, které však mohou být opraveny!

Streamliner je binární soubor, který toto umožňuje, výsledek práce spočívá v tom, že soubory jsou kratší a co je důležité mohou být čteny až 3x rychleji než dříve. Streamliner však není "kompresor" v tom smyslu, že komprimované soubory musí být před použitím dekomprimovány.

Výstup ze strealineru je pouze více "stlačená" verze bin. souboru, která může být načtena obvyklým způsobem (volbou L z DOSu).

#### Mechanismus stlačování:

Před vysvětlením jak použít program samotný se podívejme, který typ souborů bude užitečné "stlačovat".

Existují 3 případy, kdy soubory obsahují nadbytečné informace, které mohou být odstraněny streamlinerem.

#### Případ č. 1:

Binární soubor byl vytvořen mnoha populárními assemblyery nebo

compilátory. Výsledek toho je, že mnoho těchto systémů rozbíjí soubor na mnoho segmentů.

Segment je část bin. souboru, který sestává z bloku dat a 4 bytu, které říkají kam tento blok v paměti vložit.

Co mnoho assemblerů a kompilátorů dělá je to, že vytvářejí množství malých segmentů místo jednoho velkého. Například assembler může vytvořit skupinu 100 bytových segmentů přičemž první se ukládá v paměti od adresy 400, druhý od 500, třetí od 600 atd... Pokud se spojí množství těchto segmentů v jeden velký, ušetříme několikrát ty 4 byty na začátku. Co je důležitější, takovýto velký soubor bude přečten rychleji, protože DOS se nebude zastavovat a "přemýšlet" před každým segmentem.

Snadné, že?

Mezi assembly a kompilátory, které rozbíjejí soubory na mnoho malých patří:

MAC/65 (který vždycky tvoří 251 - bytové segmenty)

ATARI MACRO ASSEMBLER, DEEP BLUE C, a ACE-C.

Atari assembler editor soubory nerozděluje.

### Příklad č.2:

Bin. soubory, které byly spojeny. Toto nastává u AUTORUN. SYS souborů.

Řekněme, že máte 3 velké programy ve stroj. jazyku, které rozšiřují výkonnost Atari Basicu. Protože nemůžete použít 3 AUTORUM; soubory, spojíte je dohromady pomocí DOS - volby COPY/APPEND (přidáním /A).

Soubory budou spojeny v jeden, ale ke každému budou přidány 2 zvláštní byty.

Nuže to jsou důvody proč soubory obsahují nadbytečné informace, které přidávají drahocenné vteřiny k době čtení.

### Do práce!

Po natažení streamlineru uvidíte obrazovku se dvěma okénky. To nahoře je větší a slouží jako informační. Obsahuje všechny druhy informací co je prováděno se souborem.

Pod ním je komunikační okénko.

Vložte disketu do drivu č. 1 a potom napište jméno souboru se

kterým budete pracovat.

Nezáleží na tom, zda je soubor uzamčen, protože toto bude odhaleno a soubor se odemkne, když streamliner dokončí svoji práci, soubor bude opět uzamčen.

Pokud se disketa točí, informační okénko stále ukazuje co se právě provádí.

Je zobrazen počet ze čtených a psaných bytů. Také počet zvláštních segmentů a zvláštních bytů nalezených v souboru je zobrazen, abyste byli informováni jak málo nebo mnoho má být soubor upraven.

Pokud soubor nepotřebuje "stlačení" (tzn. neobsahuje žádné zvláštní segmenty nebo byty) objeví se hlášení "NO NEED TO REWRITE, PRESS A KEY".

Jediné omezení streamlineru je ve velikosti souboru se kterým pracujeme. To proto, že celý soubor je uložen v RAM.

Se 48K paměti by to neměl být problém. V nejhorším se objeví hlášení "FILE TOO LARGE MESSAGE".

## D R A W I T

Popis grafického editoru na ATARI

Autor popisu: RNDr. Zdeněk Ambros

Jedná se o grafický editor pracující s rozlišovací schopností 159 x 95 bodů.

Pracuje v grafickém a textovém modu. Má možnost výstupu obrázku na kazetu, disketu nebo tiskárnu. Po nahrání programu do paměti ATARI 800 XL se objeví hlavička DRAWIT, Copyright 1983, James W Burton. Potom se objeví v horní části obrazovky řádek pro komunikaci s uživatelem a uprostřed obrazovky blikající kurzor ve tvaru kříže o rozměru 7 x 10 bodů. Někdy se po nahrání programu stane, že je obrazovka vyplněna zmetí nesmyslných znaků. Je to proto, že program neinicIALIZUJE pracovní oblast paměti. Stačí obrazovku vyčistit stiskem kláves SHIFT-CLEAR. Program zabírá v paměti ATARI necelé čtyři kByte. Zato je schopen adresovat asi 36 kByte obrazové informace.

V komunikačním řádku jsou tyto údaje:

- X,Y..... aktuální souřadnice kurzoru ( $X=\emptyset-159$ ;  $Y=\emptyset-95$ )
- x,y..... souřadnice naposledy vykresleného bodu ( $x=X$ ;  $y=Y$ )
- ..... hodnota  $\emptyset-3$  udává číslo barvy kreslené čáry podle obsahu barvového registru v C. Hodnota  $\emptyset$  mění barvu pozadí.
- C ..... barva právě používaná pro zobrazení (16 barev k dispozici uživateli)
- I ..... intenzita všech čar v obrázku, které byly nakresleny stejnou barvou (v rozmezí  $\emptyset-8$ )
- P ..... číslo právě zobrazované strany (v rozmezí 1-9)

Na každé straně je možné kreslit obrázek ve třech zvolených barvách. Přitom je možné v průběhu práce měnit obsahy barvových registrů stiskem C. Tím se ovšem změní barva všech čar, které byly nakresleny v modu daném hodnotou, takže počet použitelných barev je stále 3.

Význam jednotlivých funkčních kláves v grafickém modu:

Kurzorem lze po obrazovce pohybovat stiskem klávesy CONTROL a současným stiskem klávesy se šipkou jako ve standartním editoru ATARI s tím rozdílem, že klávesu CONTROL není třeba stále držet. Je-li stisknuta klávesa šipka vpravo bez CONTROL, objeví se místo kříže kurzor jako blikající bod. Při stisku klávesy; zmizí kurzor i blikající bod obrázku. Kurzor lze v obou případech vyvolat stiskem klávesy šipka vlevo bez CONTROL.

- U ..... posune koncový bod do bodu kurzoru
- L ..... vykreslí čáru mezi koncovým bodem a kurzorem
- B ..... vykreslí pravoúhlý rovnoběžník o stranách  $a=\text{abs}(X-x)$ ;  $b=\text{abs}(Y-y)$  nebo čáru o délce a nebo b.
- O ..... vykreslí kružnici o poloměru daném vzdáleností koncového bodu od kurzoru se středem v souřadnicích kurzoru.
- Z ..... "přiblíží" vykreslované místo při vykreslení detailů ve dvou úrovních a současně vykreslí body o transformovaných souřadnicích X,Y a x,y. Jednotlivé úrovně se přepínají opakovaným stiskem klávesy Z.
- R ..... transformuje souřadnice všech bodů obrázku a posune celý obrázek tak, že současně posune koncový bod do bodu kurzoru.

- F ..... vyplní ohraničenou oblast (např. kruh) barvou = .  
FILL je možné přerušit stiskem libovolné klávesy kromě kláves BREAK, CONTROL a SHIFT.  
Při provádění příkazu záleží na poloze kurzoru.
- W ..... zruší komunikační okénko. To se objeví při dalším stisku W.
- C ..... mění barvu v barvovém registru = .
- I ..... mění intenzitu čar, kreslených barvou = . V případě, že ==Ø, mění I intenzitu barvy pozadí.
- P ..... přepíná devět graficky nezávislých stran.
- K ..... v komunikačním okénku se objeví nápis KEEP FILENAME C: očekává zadání jména obrázku (nepovinné) a RETURN. Po stisku RETURN se ozve dvojitý tón a při stisku libovolné klávesy je obrázek vyslán na vybrané periferní zařízení. C: lze přepsat podle toho, na které zařízení má být výstup realizován. Pozor na zadání písmene S. V tomto případě dostane editor kolaps. Nahrávání lze přerušit a do normálního modu se lze vrátit stiskem BREAK.
- G ..... v komunikačním okénku se objeví nápis GET FILENAME C: po stisku RETURN se ozve tón a editor je připraven přijmout grafickou informaci.
- SHIFT-CLEAR, CONTROL-CLEAR: funkce obou dvojic je stejná a způsobí smazání celého obrázku.
- SHIFT-INSERT: Tato funkce poskytuje dvě možnosti.
1. Prioritní postavení má v editoru první strana (P=1). Jestliže na této straně nakreslíme barvou = obrázek, je možné tento obrázek překreslit stiskem těchto kláves na libovolnou stranu 2-9, pokud je při překreslování ==Ø. Obrázek se vykreslí v čísle barvy, pod kterým byl kreslen na straně P=1 a ve stejném místě, kde byl na straně P=1 nakreslen. Pokud jej potřebujeme zase vymazat, nezbyvá než smazat celou 1. stranu a použít opět tuto funkci. Omezení vyplývá z následujícího.
  2. Pokud jsme na straně 2-9 a ==1,2,3, potom SHIFT-INSERT vymaže všechny čáry kreslené stejnou barvou kromě těch, které patří obrázku z 1. strany, pokud na této straně ještě je. Proto opatrně s použitím této funkce !
- K mazání jednotlivých čar obrázku je vhodný jiný postup: Přepneme na ==Ø a od tohoto momentu kreslí editor čáry v barvě pozadí. Proto lze "vymazat" libovolnou část obrázku jeho

opětovným vykreslením. Při troše zručnosti lze takto "gumovat" např. kružnici jejím vykreslením v modu ==Ø.

### Textový mod

Klávesa OPTION je určena k přepínání grafického a textového modu editoru. Pro nastavení barvy a intenzity písma platí stejná pravidla jako v grafickém modu. Tyto parametry spolu s polohou kurzoru je třeba nastavit před stiskem OPTION. Pokud jsme kurzor a koncový bod posunuli do souřadnice  $X=Ø$ ,  $x=Ø$  a stiskli OPTION, je nastaven nejmenší rozměr písma tj. 1Ø x 1Ø obrazových bodů. Na celý řádek se vejde dvacet znaků. Souřadnice levého horního rohu prvního znaku je  $X+1$ ,  $Y+1$ . Všechny klávesy v textovém modu mají funkci standartního editoru ATARI. Po napsání dvacátého znaku se editor automaticky přepne do grafického modu. Do grafického modu se přepne rovněž po stisku OPTION. Výšku i šířku písma lze nastavit zvětšením vzdálenosti kurzoru a koncového bodu. Pokud chceme umazat část textu na řádku, přesuneme kurzor i koncový bod do správných souřadnic, nastavíme ==Ø a po stisku OPTION napíšeme písmeno, které chceme vymazat. Chce to trochu cviku !

Na závěr zbývá popřát příjemnou práci s programem DRAWIT !

### J A Z Y K C

- strukturovaný vyšší programovací jazyk vytvořený k optimalizaci doby výpočtu, kapacity paměti a výkonnosti procesoru.
- byl zhotoven jako systémový programovací jazyk pro operační systém UNIX.

Computer Dictionary  
(Charles Sippl, USA 1982)

Chcete i Vy proniknout do světa 16-bitových počítačů? Vaše 8-bitové ATARI XL/XE Vám teď nabízí tuto možnost. Atari Klub Olomouc nabízí všem zájemcům o jazyk C překlad manuálu verze jazyka C - DEEP BLUE C 1.2 od firmy ANTIC APX Classics. Překlad zahrnuje i další vybavení jazyka C - popis a použití programu MATHLIB.

Stručná charakteristika těchto programů:

## DEEP BLUE C COMPILER verze 1.2

John Palewitch

(program byl vytvořen v jazyku C)

Jazyk C je jedním z nejlépe přenositelných jazyků - program se dá použít i u jiných počítačů pouze s malými změnami! Eliminuje zdlouhavé hodiny odstraňování chyb ze strojového jazyka. Je rychlejší a mnohem výkonější než BASIC. Tento jazyk se v současné době stává standartním vybavením pro novou generaci víceuživatelských počítačových pracovišť nebo počítačů založených na procesoru 68000 (jako je např. ATARI ST). Ukazatele, rekurzivní funkce a vysoká úroveň řízení strukturovaného programování Vám dává kompletní systém k tvoření profesionálních programů na ATARI XL/XE. Potřebné vybavení: - minimálně 48 kB RAM

- disketová jednotka

- textový editor (SpeedScript)

Optimální vybavení: přidání rychlé pohyblivé řádové čárky a grafiky programu:

## MATHLIB for DEEP BLUE C

Frank Paris

(program byl vytvořen v jazyku DEEP BLUE C a ML)

Mathlib - knihovna matematických funkcí, která byla vytvořena k rozšíření možností jazyka DEEP BLUE C v oblasti výpočtů s pohyblivou řádovou čárkou a využití grafických schopností 8-bitových počítačů ATARI XL/XE. Program poskytuje přístup k funkcím operačního systému (OS) ATARI, který program DEEP BLUE C COMPILER nemá. Mezi 32 novými matematickými funkcemi, které Mathlib obsahuje jsou:

- převod celočíselných proměnných v pohyblivé řádové čárce

- převod stupně - radiány

- převod ATASCII proměnných v pohyblivé řádové čárce

- operace s pohyblivou řádovou čárkou: sčítání, odečítání, násobení, dělení, druhá odmocnina, přirozený logaritmus, desítkový logaritmus, exponenciály a více ...

Demonstrační programy Vám ukáží příklady použití těchto nových funkcí, mimo jiné také použití tzv. Turtle grafiky (grafika podobná grafice jazyka LOGO).

ANTIC Catalog

Jiří Hrdlička, Olomouc

(c) 1988 by GIA Software

O b s a h :	str.:
Supermon V 2.0 - 2.1 ... Petr Vičar	193
ZXL 3.1 - ZX loader se zvýšeným komfortem obsluhy	208
Modul cartridge ... J. Halamík, P. Vičar	212
Turbo 2000 operační systém ... M. Říha	227
Světelný sensor	232
Turbo Basic - rozdíly v detailech ... E. Belluš	242
Uspořádání sektorů na disku operačního systému DOS 2.5 ... M. Outlý	248
Streamliner ... M. Outlý	250
Drawit ... Z. Ambros	252
Jazyk C ... J. Hrdlička	255

#### Redakční oznámení:

Publikaci Jazyk C připravujeme do tisku v rozsahu 240 str. popisu jazyka a 60 stran jeho aplikací pro Atari. Bude vydána v omezeném nákladu ve IV. čtvrtletí 1988 a objednávky možno zaslat za celé kluby přes krajské vedoucí.

Výzva k založení dalších specializovaných skupin:

2. Hledám zájemce, kteří se zabývají statikou, dynamikou a zakládáním stavebních konstrukcí: Ing. Zdeněk Kroča  
Za mlýnem 7, 750 00 Přerov

3. Hledám zájemce o rozšíření naší skupiny v Nových Zámkách, která využívá Atari v medicíně. Již jsme vytvořili program na "faktorovou analýzu rizikových faktorů isch. chorob srdce" a chceme se pustit do dalších programů: MUDr. Tibor Důriš  
Nábřežná 4  
940 73 Nové Zámky

Zpravodaj ATARI KLUB č. 7-8 1988

Vydává ATARI klub Olomouc

NEPRODEJNÉ - odběr vázán na příspěvek

Odpovědný redaktor : Ing. Pavel Kopečný

Odborný redaktor: Ing. Dobromil Pavlík

Neprošlo jazykovou úpravou

Přetisk pouze se souhlasem redakce

Povoleno OK ONV Olomouc, č.j. 0380500387

Tisk: Moravské tiskařské závody Olomouc

© ATARI klub Olomouc 1988

Předáno do tisku: srpen 1988

30 - j