

ZO ZVÄZARMU ELEKTRONIKA MARTIN



**VÝBER Z PREDNÁŠOK
5. CELOSLOVENSKEHO SEMINÁRA
O POČÍTAČOCH ATARI**

MARTIN 25 – 27. NOVEMBER 1988

U V O D

ACTION! JE NOVÝ JAZYK ATARI, KTORÝ MOŽME ZARADIŤ MEDZI JAZYK PASCAL A C. PROGRAMY V ŇOM NAPISANE SU RYCHLE AKO V STROJOVOM KÓDE, A VĎAKA ŠTRUKTÚRE, EDITORU A BOHATEJ KNIZNICI PROCEDUR JE PROGRAMOVANIE OMNOHO JEDNODUCHSIE. KOMPILATOR ACTION! JE DOSTUPNÝ NA VŠETKYCH TROCH NOSICOCH POUŽÍVANÝCH V ATARI: NA KAZETE, NA DISKETE A NA CARTRIDGE. JE MOŽNE ZABUDOVÁVAŤ ACTION! NA STAĽO DO POCÍTACA.

SYSTEM ACTION! SA SKLÁDA Z EDITORA, KOMPILATORA A MONITORA, KTORÝ RIADI CELKOVÚ PRÁCU SYSTEMU. ZDROJOVÝ PROGRAM PÍŠEME POMOCOU EDITORA A NÁSLEDNE HO MOŽME KOMPILOVAŤ. JE MOŽNE VÝKONANIE NAPISANÉHO PROGRAMU HNED PO SKOMPILOVANÍ, AKO I V ZDROJOVEJ VERZII. V DRUHOM PRÍPADE MONITOR ČÍŤA ZDROJOVÝ PROGRAM, KOMPILUJE HO A OKAMŽITE VÝKONÁVA. NEVÝHODOU SYSTEMU ACTION! MOŽE BYŤ DLHŠIA DOBA PRI NACÍTAVANÍ PROGRAMU PRED JEHO VÝKONÁVANÍM (AJ SKOMPILOVANÉHO). VÝPLÝVA TO Z FAKTU, ŽE KOMPILATOR NEĽINKUJE DO VÝKONATEĽNEHO PROGRAMU KNIZNICU PROCEDUR. EXISTUJE AJ TAKÝ KOMPILATOR, OVSEM IBA U PRODUCENTA.

PROGRAMY V ACTION! SA SKLÁDAJU Z ČASŤI, Z KTORÝCH KAŽDA OBSAHUJE MNOŽINU PRÍSLUSNÝCH INSTRUKCII VÝKONÁVAJUCÍCH AKESI OPERÁCIE, VĎAKA TOMU JE MOŽNO LAHKO VLOŽIŤ DO EXISTUJÚCEHO PROGRAMU NESKORSIE NAPISANE BLOKY. JEDNOU Z PODMIENOK TAKTO ŠTRUKTÚRALNE NAPISANÉHO PROGRAMU JE TO, ŽE PROGRAM MUSI BYŤ ZLOŽENÝ Z FUNGUJÚCICH USEKOV. USEKY V ACTION! SU PROCEDURY A FUNKCIE. OBÝVKLE ICH PROGRAM OBSAHUJE VIAC, ALE POŽADOVANÝ V URČITOM CASE JE VZDY ASPON JEDEN. V PROGRAME OBSAHUJUCOM NIEKOĽKO PROCEDUR, VÝKONÁVANIE ZACÍNA OD POSLEDNEJ, KTORÁ POTOM MUSI RIADIŤ CELÝ PROGRAM.

E D I T O R

EDITOR SLUŽI NA VYTVORENIE NOVÝCH PROGRAMOV A REDIĎOVANIE STARYCH. EDITOR ACTION! JE JEDNÝM Z NAJLEPŠIE PREPRACOVANÝCH PROGRAMOVACÍCH EDITOROV. MOŽNO HO POUŽÍVAŤ AJ ZA INÝM ÚCELOM (PROGRAMOVANIE V INÝCH JAZYKOCH, OPRÁVA TEXTOV ATD).

PO NASTARTOVANÍ SYSTEMU V DOLNEJ ČASŤI OBRAZOVKY SA OBJAVÍ SVIETIACI NAPIS "ACTION! (C)1983 ACS". JE TO KOMUNIKACNÝ RIADOK, V KTOROM SA VYSVIEČUJU CHYBOVE OZNAMY, INE INFORMÁCIE A OZNAMY EDITORA. EDITOR MOŽE POUŽÍVAŤ Z OKNA A KOMUNIKACNÝ RIADOK JE UMIESTNENÝ MEDZI OKNAMI. EDITOR ACTION! JE NAPROGRAMOVANÝ TAK, ABY UMOŽNIL ČO NAJLAHŠIE PÍSANIE PROGRAMOV. UMOŽNUJE PÍSANIE RIADKOV O DLŽKE DO 240 ZNAKOV, ALE V OKNE EDITORA JE VIDIEŤ LEN 38 ZNAKOV. AK SA DOSIAHNE MAXIMÁLNE MOŽNEJ DLŽKY RIADKU, EDITOR TO SIGNALIZUJE ZVUKOVÝM TONOM; AK JE NAPISANÝ RIADOK DLHŠI AKO ŠÍRKA OKNA, POTOM ZNAK NA OKRAJI OKNA (PRAVEJ ALEBO LAVEJ) JE VYSVIEŤENÝ AKO NEGATÍVNY (INVERSE VIDEO), ABY TENTO FAKT BOĽO VIDIEŤ.

TEXT PROGRAMU SA PISE EDITOROM BEZ SPECIALNYCH CINNOSTI - TAK AKO PRI PISANI PISACIM STROJOM. PRI PISANI KONTROLNEHO ZNAKU STLACIME NAJPRV KLAVESU ESC, TYMTO HOVORIME EDITORU, ZE NASLEDOVNY ZNAK MA BYT INTERPRETOVANY AKO TEXT, A NIE AKO PRIKAZY PRE EDITOR. PRIKAZY EDITORA SU AJ NIEKTORE KOMBINACIE KLAVESY STLACANE SUCASNE, NAPR. SHIFT-CLEAR.

REZIMY PRACE EDITORA

EDITOR MOZE PRACOVAT V DVOCH REZIMOCCH (MODOCCH): VKLADANIA A PREPISOVANIA. V REZIME PREPISOVANIA PISANY TEXT NAHRADZA TEXT UZ EXISTUJUCI, ZNAK PO ZNAKU. V REZIME VKLADANIA JE NOVY TEXT ZAPISANY MEDZI STARY TEXT BEZ JEHO VYMAZANIA; PO NASTARTOVANI EDITOR PRACUJE V REZIME PREPISOVANIA. NA PRECHOD MEDZI REZIMAMI EDITORA SLUZI PRIKAZ SHIFT-CONTROL-I.

OBSLUHA EDITORA

NA ODSTRANENIE TEXTU Z EDITORA SA POUZIVA PRIKAZ SHIFT-CLEAR. VYMAZAVA NIENEN TEXT NA OBRAZOVKE, ALE AJ CELU PRACU EDITORA, KED SA POUZIVAJU DVE OKNA EDITORA, PRIKAZ SA TYKA TOHO OKNA, V KTOROM SA AKTUÁLNE NACHADZA KURZOR ABY SA PREDISLO NAHODNEMU ZNICENIU REDIGOVANEOHO PROGRAMU, EDITOR SA PYTA: "CLEAR?". TREBA ODPOVEDAT "Y" (ANO) ALEBO "N" (NIE). AK SME V REDIGOVANOM TEXTE UKONCILI ZMENY A TENTO NEBOL ZAPISANY, EDITOR SA PYTA ZNOVA: "NOT SAVED-DELETE?" (NEZAPISANY, VYMAZAT?), ABY SME SA PRESVEDCILI, ZE NECHCEME ZAPISAT OPRAVENU VERZIU.

EXISTUJE JEDEN SPOSOB PRACE UKONCENIA PRACE EDITORA (OKREM VYPNUTIA POCITACA) - SHIFT-CONTROL-M. TENTO PRIKAZ UMOZNUJE ODOVZDANIE RIADENIA SYSTEMU MONITORU, OKIAL MOZME VYVOLAT INE ELEMENTY SYSTEMU ALEBO MOZME PREJST DO DOS-U.

PROGRAM NAPISANY V EDITORE MOZE BYT ZAPISANY NA LUBOVOLNE VONKAJSIE ZARIADENIE. UMOZNUJE TO PRIKAZ SHIFT-CONTROL-W. PO STLACENI TEJTO KOMBINACIE KLAVES SA OBJAVI OZNAM: "WRITE?". PRI ZAPISOVANI SUBORU NA DISKETU JE NUTNE ZADAT CELY NAZOV SUBORU - D:NAZOV,EXT. PRI ZAPISANI NA MAGNEFOTON ALEBO NA TLACIAREN STACI UDAT NAZOV ZARIADENIA - ZODPOVEDAJUCE C: ALEBO P:

ANALOGICKY NACITAVANIE ZAPISANEOHO SUBORU SA UMOZNI PRIKAZOM SHIFT-CONTROL-R; PO KTOROM SA POCITAC PYTA "READ?". TREBA UDAT NAZOV SUBORU UVEDENY VYSSIE. DOPLNKOVE PRIKAZY UMOZNUJU NACITANIE ZAZNAMNIKA DISKETY (DIRECTORY). JE TO UMOZNENE PO ZADANI NAZVU SUBORU V TVARE: ??:*.*. AK CHCEME NACITAT DIRECTORY Z DRUHEJ MECHANIKY DISKOV, STACI NAHRADIT CISLO 1 CISLOM PRISLUSNEJ MECHANIKY.

POHYB KURZORA

NÍZŠIE UVEDENÉ PRIKAZY UMOZNUJU POHYB KURZORA V CELOM ROZSAHU EDITORA ACTION!:

CONTROL- ←	-	KURZOR SA PRESUNIE O JEDEN ZNAK VĽAVO
CONTROL- →	-	KURZOR SA PRESUNIE O JEDEN ZNAK VPRAVO
CONTROL- ↑	-	KURZOR SA PRESUNIE O JEDEN RIADOK HORE
CONTROL- ↓	-	KURZOR SA PRESUNIE O JEDEN RIADOK DOLE
SHIFT-CONTROL-←	-	KURZOR SA PRESUNIE NA ZACIATOK AKTUALNEHO RIADKU
SHIFT-CONTROL-→	-	KURZOR SA PRESUNIE NA KONIEC AKTUALNEHO RIADKU
SHIFT-CONTROL-H	-	PRESUNUTIE KURZORA NA ZACIATOK SUBORU
TAB	-	PRESUNUTIE KURZORA NA NASLEDUJUCU POZICIU TABULATORA
SHIFT-TAB	-	NASTAVENIE POZICIE TABULATORA
CONTROL-TAB	-	VYMAZANIE POZICIE TABULATORA

CASOM SA UKAZE POTREBA ROZDELIT RIADOK PROGRAMU NA DVA RIADKY. ZATYM CIELOM JE POTREBNE NASTAVIT KURZOR NA ZNAK, KTORÝ MA BYT PRVYM ZNAKOM V DRUHOM RIADKU A STĽACIT SHIFT-CONTROL-RETURN. OPACNA OPERACIA SA VYKONA PRI NASTAVENI KURZORA NA PRVY ZNAK DRUHEHO RIADKU A STĽACENI KLAVESNICE SHIFT-CONTROL-BACKSPACE.

REDIGOVANIE TEXTU

EDITOR ACTION! OBSAHUJE VACŠINU FUNKCIÍ EXISTUJUCICH V NORMALNÝCH EDITOROCH TEXTOV. JEDNOU Z NICH JE ZMENA PEVNÝCH FRAGMENTOV TEXTU ZA INE. UMOZNUJE TO PRIKAZ SHIFT-CONTROL-S. PO JEHO POUZITÍ SA VYSVIETI OZNAM "SUBSTITUTE?" A KĎD TATO FUNKCIA UZ BOLA POUZIVANA, VYSVIETI SA TAKISTO AJ NAPOSLEDY MENENÝ "NOVÝ" TEXT. JE TREBA NAPIŠAT TEXT, KTORÝ MA BYT UMIESTNENÝ V REDIGOVANOM PROGRAME A STĽACIT RETURN, AVŠAK SAMOTNE STĽACENIE RETURN ZACHOVA POUVODNÝ TEXT. TERAZ SA VYSVIETI OZNAM "FOR?", PO KTOROM TREBA NAPIŠAT TEXT, KTORÝ MA BYT NAHRADENÝ (KĎD UZ FUNKCIA BOLA POUZIVANA STĽACI POUZIT ANAĽOGICKY AKO S "NOVÝM" TEXTOM), PO STĽACENI RETURN EDITOR NAJDE PRVÝ VÝŠKYT ZADANEHO "STAREHO" TEXTU A ZAMENI HO ZA NOVÝ. NASLEDUJUCI PRIKAZ SHIFT-CONTROL-S UMOZNI ZMENU ZADANEHO TEXTU NA ĎALŠOM MIESTE BEZ ZADANIA OZNAMOV "SUBSTITUTE?" A "FOR?" (KĎD SA NEPOUZILA PREDTÝM INA FUNKCIA EDITORA), KĎD SA "STARY" TEXT NENACHADZA V REDIGOVANOM PROGRAME, VYSVIETI SA OZNAM "NOT FOUND" (NENAJDENÝ).

JE TAKTIEZ MOŽNE PREMIESTNENIE ALEBO KOPIROVANIE CELÝCH BĽOKOV TEXTU CEZ KOPIROVACI BAFER. ZAKAZDÝM, KĎD SA POUZIJE PRIKAZ SHIFT-DELETE, VYMAZANÝ RIADOK SA PRESUNIE DO KOPIROVACIEHO BAFRA. PRIKAZ SHIFT-CONTROL-P PRESUNIE VYMAZANÝ RIADOK NA AKTUALNU POZICIU KURZORA. KOPIROVACI BAFER SA VYMAZE PRI KAZDOM POUZITÍ SHIFT-DELETE, OVŠEM S JEDNOU VÝNIMKOU. AK SA SHIFT-DELETE POUZIJE VIACKRAT (BEZ INÝCH PRIKAZOV ALEBO PISANIA TEXTU), POTOM BAFER NIEJE VYMAZANÝ. TO UMOZNUJE UMIESTNIŤ DO BAFRA CELÝ BĽOK TEXTU A

PREMIESTNIT HO NA LUBOVOLNE MIESTO POMOCOU SHIFT-CONTROL-P, PRETOZE PRESUNUTIE TEXTU NEMAZE BAFER, MOZNO HO KOPIROVAT VIAC KRAT, AK SA POCAS REDIGOVANIA TEXTU RIADKU VYSKYTNE CHYBA, MOZME OBNOVIT POVODNY STAV TOHTO RIADKU PRIKAZOM SHIFT-CONTROL-U, TATO FUNKCIA PONECHAVA KURZOR NA POVODNOM MIESTE,

OKNA EDITORA

DOPLNKOVOU FUNKCIOU EDITORA ACTIONI JE MOZNOST VYUZIVANIA DRUHEHO OKNA A SUCASNEHO REDIGOVANIA DVOCH ROZNYCH PROGRAMOV, DRUHE OKNO SA VYVOLAVA PRIKAZOM SHIFT-CONTROL-2. TENTO PRIKAZ SLUZI TAKTIEZ NA PRENESENIE KURZORA Z PRVEHO DO DRUHEHO OKNA, PRENESENIE KURZORA Z DRUHEHO OKNA DO PRVEHO SA VYKONAVA PRIKAZOM SHIFT-CONTROL-1. AK CHCEME VYMAZAT OKNO MUSIME NAJPRV DON UMIESTNIT KURZOR A NASLEDNE STLACIT KLAVESY SHIFT-CONTROL-D. V KOMUNIKACNOM RIADKU SA OBJAVI OZNAM "DELETE WINDOW?". DALSI POSTUP JE TEN ISTY AKO V PREDCHADZAJUCICH PRIPADOCH,

OKNO MOZE BYT PRESUVANE HORE ALEBO DOLE V ZAVISLOSTI OD POHYBU KURZORA. DOPLNKOVE PRIKAZY UMOZNUJU POHYB CELEHO OKNA HORE ALEBO DOLE. PO PRIKAZE SHIFT-CONTROL-4 SA OKNO PREMIESTNI SMEROM HORE TAK, ZE NAJVYSSI RIADOK STAREHO OKNA JE TERAZ NAJNIZSIM RIADKOM, ANALOGICKY PRIKAZ SHIFT-CONTROL-4 PRESUVA OKNO SMEROM NADOL. JE TAKTIEZ MOZNE PRESUVANIE CELEHO OKNA O JEDEEN ZNAK VPRAVO (SHIFT-CONTROL-3) ALEBO VLAVO (SHIFT-CONTROL-2).

NAVESTIA

NAVESTIA UMOZNUJU OZNACIT LUBOVOLNE MIESTA V TEXTE. AK CHCEME NASTAVIT NAVESTIE NA AKTUALNEJ POZICII KURZORA, TREBA ZADAT PRIKAZ SHIFT-CONTROL-T O PO VYSVIETENI OZNAMU "TAG ID;" ZAPISAT JEDEEN ZNAK IDENTIFIKUJUCI NAVESTIE A STLACIT RETURN; AK UZ NAVESTIE OZNACENE ZADANYM ZNAKOM EXISTUJE, ONO SA PRESUNIE NA NOVO ZADANE MIESTO.

PRIKAZ SHIFT-CONTROL-G PRENESA KURZOR NA NAVESTIE, KTOREHO ZNAK SA ZAPISE AKO ODPOVED NA OZNAM "TAG ID;". KED SA UDANE NAVESTIE MENACHADZA V TEXTE, VYSVIETI SA KOMUNIKACNY RIADOK "TAG ID NOT SET" (NAVESTIE NEBOLO NASTAVENE).

UPOZORNENIE: KAZDA OPERACIA, KTORA MENI OBSAH RIADKU, MAZE AJ NAVESTIE V TOMTO RIADKU.

M O N I T O R

MONITOR ACTION! RIADI PRACU CELEHO SYSTEMU; PRIKAZOVY RIADOK MONITORA SA NACHADZA V HORNEJ CASTI OBRAZOVKY A ZOBRAZUJE ZNAK > ALEBO KURZOR. OSTATNA CAST OBRAZOVKY JE URCENA ROZSAHOM KOMUNIKACNYCH OZNAMOV, MA VIAC TVAROV, V CASE VYKONAVANIA PROGRAMU SLUZI NA VYSVIETENIE JEHO POZIADAVIEK, MOZE SA POUZIVAT NA SLEDOVANIE BEHU PROGRAMU. AK SA VYSKYTNE V PROGRAME CHYBA, POTOM SA NA MIESTE OZNAMOV VYSVIETI CISLO CHYBOVEHO HLASENIA A CAST PROGRAMU S RIADKOM, KDE SA VYSKYTLA CHYBA.

PRIKAZY MONITORA

MONITOR ROZPOZNAVÁ ZADANE MU PRIKAZY LEN PODLA PRVEHO ZNAKU. NIJE TREBA PISAT CELE SLOVA, LEN POCIATOCNE ZNAKY; VYKONAVANIE ZADANEHO PRIKAZU NASLEDUJE PO STLACENI KLAVESY RETURN.

BOOT(B) - UMOZNUJE RESTART SYSTEMU A NAVRAT SPAT DO EDITORA. MAZE VSETKY ZATVORENE PROGRAMY V PAMATI A PREMENNE.

COMPILE(C) - PROGRAM NAPISANY V ACTION! MUSI BYT SKOMPILOVANY PRED VYKONANIM. PRIKAZ COMPILE VYVOLA KOMPILATOR ACTION! A UMOZNI SKOMPILOVANIE PROGRAMU NACHADZAJUCEHO SA V PAMATI EDITORA. MOZME TAKTIEZ SKOMPILOVAT PROGRAM, KTORY SA NACHADZA NA VONKAJSEJ PERIFERII. V TOMTO PRIPADE TREBA UDAT NAZOV NAPR. C "C:" PRE MAGNETOFON ALEBO C "D:\NAZOV\EXT" PRE MECHANIKU DISKETY. KED MA BYT PROGRAM NACITANY Z MECHANIKY 1, MOZME VYNECHAT ONACENIE ZARIADENIA: C "NAZOV\EXT"; AK KOMPILATOR NARAZI V PROGRAME NA CHYBU, PRERUSI KOMPILACIU A ODVZDA RIADENIE OPAT DO MONITORA.

DOS(D) - OPUSTENIE SYSTEMU ACTION! A PREJDENIE DO DOS-U (PRACA EDITORA A MONITORA SA ZNICI).

EDITOR(E) - PREJDENIE DO EDITORA; AK V CASE KOMPILACIE PROGRAMU SA VYSKYTNE CHYBA, PO PREJDENI DO EDITORA SA PROGRAM NACHADZA V RIADKU, NASLEDUJUCOM PO CHYBE.

OPTIONS(O) - VYVOLANIE MENU VOLIEB, KTORE SLUZIA NA ZMENU PARAMETROV PRACE EDITORA, KOMPILATORA A MONITORA. KAZDA VOLBA JE VYSVIETENA V PRIKAZOVOM RIADKU. AK JU CHCEME ZMENIT, NAPISEME NOVY TVAR A STLACIME RETURN. SAMOTNE STLACENIE RETURN PONECHAVA DANU VOLBU BEZ ZMENY. MENU VOLIEB MOZNO UKONCIT STLACENIM ESC.

DALEJ SU VYMENOVANE VSETKY VOLBY; FORMAT ZAPISU; NAZOV VOLBY - PARAMETRE (STANDARTNE TVARY = CASTI SYSTEMU, NA KTORE MA VPLYV).

DISPLAY? = ZOBRAZENIE (Y = E,K,M)
ZOBRAZENIE MOZE BYT POTLACENE ZA UCELOM ZRYCHLENIA KOMPILACIE A OPERACII VSTUPU/VYSTUPU. NAPISANIE "N"(NIE) POTLACA ZOBRAZOVANIE, NAPISANIE "Y"(ANO) NECHAVA HO V CINNOSTI PO CELY CAS.

BELL? = SIGNALIZACIA (Y = E,K,M)
POCITAC SIGNALIZUJE VTEDY, KED NARAZI NA CHYBU. MOZNO JU POTLACIT PRI NAPISANI "N", AK NAPISEME "Y" SIGNALIZACIA SA OPAT NASTAVI.

CASE SENSITIVE? = ROZPOZNANIE PISMEN(N = K)

KED JE TATO VOLBA NASTAVENA NA "Y", POTOM SA ROZLISUJU MALE A VELKE PISMENA A INSTRUKCIE JAZYKA (NAPRIJOR, WHILE ATO) MUSIA BYT NAPISANE VELKYMI PISMENAMI. JEDNAKO PRE ULAHCENIE PRACE ZACIATOCNIKOM JE TATO VOLBA NEAKTIVNA (N) PO NASTARTOVANI SYSTEMU.

TRACE? - SLEDOVANIE (N - K)
 VDAKA TEJTO VOLBE MOZNO SLEDOVAT KOMPILACIU PROGRAMU. KED JE NASTAVENA (Y), KOMPILATOR VYSVIETI V PRIESTORE KOMUNIKACNEHO RIADKU VSETKY VYVOLAVANE PROCEDURY Z ICH PARAMETRAMI.

LIST? - LISTOVANIE (N - K)
 NASTAVENIE TEJTO VOLBY NA "Y" UMOZNUJE VYSVIETENIE V PRIESTORE KOMUNIKACNEHO RIADKU AKTUALNE KOMPILOVANEHO RIADKU.

WINDOW 1 SIZE? - ROZMER OKNA 1 (18 - E)
 ROZMER OKNA 1 V EDITORE ACTIONI SA URCI OKAMZITE, VELKOST OKNA 2 ZAVISI OD OKNA 1 - OBDVJE OKNA MUSIA MAT 23 RIADKOV. KED MAME DVE OKNA, KAZDY Z NICH MOZE OBSAHOVAT NIE MENEJ AKO 5 RIADKOV A NIE VIAC AKO 18 RIADKOV.

LINE SIZE? - VELKOST RIADKU (120 - E)
 ROZMER RIADKU JE POCET ZNAKOV V TOMTO RIADKU, MAXIMALNY ROZMER RIADKU PREDSTAVUJE 240 ZNAKOV, TATO VOLBA UMOZNUJE KONTROLOVAT DLZKU RIADKU PRI TLACI NA TLACIARNI.

LEFT MARGIN? - LAVY OKRAJ (2 - E, M)
 LAVY OKRAJ JE MIESTO, OD KTOREHO SA VYSVECUJU ZNAKY NA OBRAZOVKE. MOZE MAT ROZMER Z ROZSAHU OD 0 DO 39.

EOL CHARACTER? - ZNAK KONCA RIADKU (MEDZERA - E)
 ZNAK KONCA RIADKU SA VYSVIETI V EDITORE ACTIONI NA KONCI KAZDEHO RIADKU, NORMALNE NIEJE VIDIET (MEDZERA), JEHO VYSVIETENIE MOZE POMOC PRI REDIGOVANI PROGRAMU, MOZME TU ZAPISAT LUBOVOLNY ZNAK (AJ GRAFICKY) - NAHRADZA POUZITIE CONTROL-, ALEBO CONTROL-T.

PROCEED(P) - ZNOVUVYKONAVANIE PROGRAMU ZASTAVENEHO STLACENIM KLAVESY BREAK ALEBO POUZITIM PROCEDURY 'BREAK'. PROGRAM POKRACUJE TAK AKO KEBY NENASTALO ZIADNE PRERUSENIE.

RUN(R) - VYKONAVANIE SKOMPILOVANEHO PROGRAMU. MOZE VYSTUPOVAT V STYROCH ROZNYCH FORMATOCH:

RUN - VYKONANIE PROGRAMU NACHADZAJUCEHO SA V PAMATI EDITORA.
 RUN "NAZOV SUBORU" - NACITANIE A VYKONANIE PROGRAMU Z VONKAJSIEHO ZARIADENIA.
 RUN ADRESA - VYKONANIE PROGRAMU OD ZADANEJ ADRESY.
 RUN NAZOV PROCEDURY - VYKONANIE JEDNEJ PROCEDURY Z PROGRAMU ALEBO KNIZNICE PROCEDUR.

SET(S) - PRIKAZ UMOZNUJUCI BEZPROSTREDNY PRISTUP DO PAMATI RAM A ZMENY JEJ HODNOTY. ROBI TO ISTE AKO INSTRUKCIA SET V JAZYKU ACTIONI.

WRITE(W) - UMOZNUJE ZAPISANIE SKOMPILOVANEHO PROGRAMU NA VONKAJSIE ZARIADENIE AKO BINARNY SUBOR. PROGRAM ZAPISANY TYMTO SPOSOBOM NA DISKETU MOZE BYT VYVOLANY BEZPROSTREDNE Z DOS-U (PO NACITANI KOMPILATORA). PRI ZAPISE NA KAZETU TREBA UDAT NAZOV "C:", A NA DISKETU - "D1:NAZOV.EXT" (PRE MECHANIKU CISLO 1 - D1 MOZNO VYNECHAT).

EXECUTE(X) - PRIKAZ UMOZNUJE VYKONANIE LUBOVOLNEJ INSTRUKCIE JAZYKA ACTIONI ALEBO LUBOVOLNEJ DIREKTIVY KOMPILATORA (OKREM MODULE A SET) Z MONITORA. PO PRIKAZE TREBA NAPISAT INSTRUKCIU NA VYKONANIE,

NAPR. XECUTE PRINT("VITAJ V ACTION!").

? - VYSVIETENIE AKTUALNEJ VEKOSTI PAMATI O ZADANEJ ADRESE, ADRESOVAT MOZME KONSTANTU KOMPILATORA (POPIsANE DALEJ), V PRIESTORE KOMUNIKACNEHO RIADKU SA VYSVIETI ADRESA V SUSTAVE DESIATKOVEJ A SEsTNAsTKOVEJ; ZNAK ATASCII O KODE ZODPOVEDAJUCOM ROZMERU BYTU, VEKOST DVOCH BAJTOV STACI SEsTNAsTKOVEJ ALEBO DESIATKOVEJ SUSTAVE VO FORMATE BYTE A CARD.

NAPR.: ? OFFEE

65535,OFFEE = S DE6F3 243 59123

* - VYSVIETENIE AKTUALNEJ VEKOSTI PAMATI OD ZADANEJ ADRESY, ADRESA MOZE BYT LUBOVOLNA KONSTANTA KOMPILATORA, FORMAT VYSVIETENEHO VYSLEDKU JE TAKY AKO JE OPISANY VYSSIE, VYSVIETENIE SA PRERUSUJE sTLACENIM KLAVESY MEDZERA; sTLACENIE CONTROL-1 NA CHVILU POZDRZI VYSVIETENIE, A OPATOVNE sTLACENIE HO OBNOVUJE,

VYKONAVANIE PROGRAMU

PRI VYKONAVANI NAPISANEHO PROGRAMU CASTO VADI DLHA ODOZVA PRI VYSKYTE CHYBY, VDAKA MOZNOSTIAM MONITORA ACTIONI JE TATO CINNOST JEDNODUCHA A ZROZUMITELNA.

VOLBA TRACE - PO ZVOLENI TEJTO VOLBY MONITORA SKOMPILOVANY A VYKONAVANY PROGRAM JE SLEDOVANY NA OBRAZOVKE, NAZOV KAZDEJ VYVOLANEJ PROCEDURY JE VYSVIETENY NA OBRAZOVKE SPOLU S AKTUALNYMI PARAMETRAMI:

PRED ZASAHOM DO VYKONAVANEHO PROGRAMU SA POZDRZI VYKONAVANIE PROGRAMU, V ACTIONI JE TO MOZNE DVOMA SPOSOBMI: KLAVESOU BREAK A PROCEDUROU 'BREAK'.

KLAVESNICA BREAK - V ZASADE SA KLAVESA BREAK V ACTIONI NEPOUZIVA, JEDNAKO V DVOCH PRIPADOCH MOZME S POMOCOU KLAVESY BREAK PRERUSIT VYKONAVANIE PROGRAMU: POCAS OPERACIE VSTUPU/VYSTUPU A PRI VYVOLANI PROCEDUR, KTORE MAJU VIAC AKO TRI PARAMETRE.

KNIZNICNA PROCEDURA PROC BREAK() - AK CHCEME POZASTAVIT VYKONAVANIE PROGRAMU ZA UCELOM SLEDOVANIA SPRAVNOSTI JEHO CINNOSTI, MOZME NA NIEKOLKYCH MIESTACH UMIESTNIT VYVOLANIE PROCEDURY 'BREAK'. PRACUJE AKO INSTRUKCIA BASIC-A STOP. POKRACOVAT MOZNO VLOZENIM PRIKAZU PROCEED.

KLUCOVE SLOVA

BLIZSIE ZOZNAMENIE SA Z JAZYKOM ACTIONI SA ZACINA POZNANIM 'KLUCOVYCH SLOV'. SU TO SLOVA A SYMBOLY, KTORE MAJU SPECIALNY VYZNAM PRE KOMPILATOR- INSTRUKCIE, DEKLARACIE, OPERATORY A DIREKTIVY KOMPILATORA. KLUCOVE SLOVA SA MOZU POUZIVAT LEN SPOSOBOM AKYM SU ZADEFINOVANE V ACTIONI, NEMOZNO ICH POUZIVAT AKO NAZVY PREMENNÝCH. NASLEDUJE ZOZNAM KLUCOVYCH SLOV:

AND	FI	OR	UNTIL	=	(
ARRAY	FOR	POINTER	WHILE	<>)
BYTE	FUNC	PROC	XOR	#	,
CARD	IF	RETURN	+	>	[
CHAR	INCLUDE	RSH	-	>=]
DEFINE	INT	SET	*	<	"
DO	LSH	STEP	/	<=	'
ELSE	MOD	THEN	&	□	;
ELSEIF	MODULE	TO	%	^	
EXIT	OD	TYPE	!	@	

POZNAMKA: ZNAK □ OZNACUJE ZNAK DOLAR .

S Y M B O L Y

V DALŠOM BUDEME POPISOVAT PEVNE SYMBOLY SPOLU S ICH OBJASNENIM:

- ADRESA** - ADRESA JE ČÍSLO BYTU V PAMÄTI POCITACA, KED ZIADAME OD POCITACA, ABY UMIESTNIL COSI V PAMÄTI, MUSIME MU ZADAT ADRESU.
- IDENTIFIKATOR**- JE TO NAZOV POPISUJUCI PREMENNE, PROCEDURY ATD. NAZOV V ACTION! MUSI SPLNAT NASLEDUJUCE PODMIENKY:
1. MUSI SA LISIT OD ZNAKU;
 2. OSTATNE ZNAKY MUSIA BYT PISMENA, ČÍSĽA ALEBO ZNAKY PODCIARKNUTIA (_).
 3. NEMOZU TO BYT KLUCOVE SLOVA.
- PISMENO** - LUBOVOLNE PISMENO ABECEDY MALE AJ VELKE;
- ZNAK** - PISMENO ALEBO ČÍSLO OD 0 DO 9.
- MSB, LSB** - MSB OZNACUJE MOST SIGNIFICANT BYTE (NAJVYSSI VYZNAMOVY BYTE), LSB - LEAST SIGNIFICANT BYTE (NAJNIZSI VYZNAMOVY BYTE). V DESIATKOVEJ SUSTAVE SA NEOZNACUJU BYTE ALE ČÍSĽA. NAPRIKĽAD VO VYRAZE 54 JE NAJVYSSIE VYZNAMOVE ČÍSLO JE 5 A NAJNIZSIE JE 4. PODOBNE JE TO POCITACOVOM ZAPISE, ALE ZHODNE SO STANDARDOM MIKROPROCESORA 6502 SA VYRAZY ZAPISUJU A POUZIVAJU V PORADI LSB, MSB.
- - ZNAK DOLARA, KED JE POUZITY PRED VYRAZOM, OZNACUJE VYRAZ V SEŠTNAŠTOVOM ZAPISE (HEXADECEMÁLNOM).
- ;** - ROŠKOCIARKA JE SYMBOĽOM KOMENTARA; VSETKO CO PO NEJ NASLEDUJE JE V RIADKU IGNOROVANE KOMPILATOROM.
- <I>** - VYRAZ UZATVORENY TYMITO SYMBOLMI OZNACUJE, CO SA MA POVINNE NACHADZAT NA TOM MIESTE V INSTRUKCII ALEBO RIADKU PROGRAMU, NAPR. <IDENTIFIKATOR> OZNACUJE, ZE MUSI BYT POUZITY SPRAVNY IDENTIFIKATOR.
- [I]** - VYRAZ UZAVRETY V TYCHTO SYMBOLOCH MOZE BYT POUZITY, ALE NEMUSI, NAPR. [<IDENTIFIKATOR>] OZNACUJE, ZE SA MOZE POUZIT IDENTIFIKATOR, ALE NIJE TO POVINNE.
- I: I** - TIETO SYMBOLY OZNACUJU OPAKOVANIE (AKO V HUDBE). NAPR. I: <IDENTIFIKATOR>: I OZNACUJE, ZE JE MOZNE POUZIT LUBOVOLNY IDENTIFIKATOR (ALEBO ZIADNY).
- *** - TENTO SYMBOL OZNACUJE ALTERNATIVU, NAPR. VYRAZ <IDENTIFIKATOR>* <ADRESA> NAM HOVORII: POUZI IDENTIFIKATOR ALEBO ADRESU, ALE NIE OBE NARAZ.

T Y P Y D A T

PRED OPISOM TYPOV DAT POUZIVANYCH V ACTION! MUSIME SI TROCHU POVEDAT O PREMENNÝCH A KONSTANTACH, PRETOZE ONI SU ZAKLADNE OBJEKTY, NAD KTORÝMI OPERUJE POCITAC:

PREMENNE

LUBOVOLNY NAZOV PREMENEJ MUSI BYT POVOLENYM IDENTIFIKATOROM, NIE SU ZIADNE INE OHRANICENIA PRE POUZIVANIE MIEN, ABY SME ZISKALI PREHLADNY PROGRAM, JE DOBRE POUZIVAT OZNACUJUCE MENA PREMENNÝCH NAPR. 'CISLO' NAMIESTO 'C'. NEMA TO VPLYV NA DLZKU VYKONATELNEHO PROGRAMU (SKOMPILOVANEHO), A JEDINE SA ZVACSI O NIEKOLKO BAJTOV ZDROJOVY PROGRAM.

KONSTANTY

V ACTION! SA POUZIVAJU TRI TYPY KONSTANTY: CISELNE KONSTANTY, ZNAKOVE KONSTANTY A KONSTANTY KOMPILATORA,

CISELNE KONSTANTY SA MOZU ZAVIESŤ V TROCH ROZNYCH FORMACH: AKO CISLA DESIATKOVE, SEŠTNASTKOVE A ZNAKY.

DESIATKOVE CÍSLA NEPOTREBUJE ZIADNE SPECIÁLNE ZADAVANIE, NAPR, 27 46, 324, 65500, -4360. SEŠTNASTKOVE CÍSLA SA OZNACUJU ZNAKOM DOLAR (\$) PRED CÍSLOM, NAPR; \$00, \$300, \$400, -\$80. ZNAKOVE KONSTANTY SA OZNACUJU APOSTROFOM (') PRED ZNAKOM, NAPR. 'A', 'Q', 'R', 'F'; ZNAKY SU CISELNE KONSTANTY, PRETOZE SU VNUTORNE REPREZENTOVANE JEDNOBYTOVÝM CÍSLOM, KTORE JE ROVNE KODU ATASCII ZNAKU.

KONSTANTY KOMPILATORA SA LISIA OD TROCH PREDCHADZAJUCICH TYPOV, POUZIVAJU SA V CASE KOMPILÁCIE PROGRAMU PRI NASTAVENÍ PEVNÝCH ATRIBUTOV PREMENNÝCH, PROCEDUR ALEBO FUNKCII A NIE SU MENENE POCAS CINNOSTI PROGRAMU. KONSTANTY KOMPILATORA MOZU MAT NASLEDUJUCE FORMATY:

1. CISELNA KONSTANTA
2. IDENTIFIKATOR
3. SMERNIK
4. KOMBINACIA (DVOJICA) PREDCHADZAJUCICH TYPOV

PRVY FORMAT NIEJE TREBA OBJASNOVAT, ALE OSTATNE TREBA KRATKO OPISAT. KED SA POUZIVA IDENTIFIKATOR (NAZOV PREMENEJ, PROCEDURY ALEBO FUNKCIE) AKO KONSTANTA KOMPILATORA, POTOM UDANY TVAR JE ADRESA TOHTO IDENTIFIKATORA. TRETI FORMAT UMOZNUJE POUZIT AKO KONSTANTY KOMPILATORA SMERNIKOVE TABULKY, OSTANE SU OBYCAJNE KOMBINACIE DVOCH Z TROCH PRVÝCH TYPOV. KVOLI LEPSIEMU POROZUMENIU NIEKOLKO PRIKLADOV:

CISLICA	;ADRESA PREMENEJ "CISLICA"
\$800	;SEŠTNASTKOVA KONSTANTA
CISLO	;SMERNIK AKO KONSTANTA
\$+WSK	;5 PLUS HODNOTA SMERNIKA "WSK"
\$80+N	;80 PLUS ADRESA PREMENEJ "N"

TYPY PREMENNÝCH

V ACTIONI VYSTUPUJU TRI ZAKLADNE TYPY PREMENNÝCH BYTE, CARD A INT SPOLU S DODATKOVYMI, KTORE BUDU POPIsANE NESKORSIE. VSETKY ZAKLADNE TYPY PREMENNÝCH SU CISELNE:

BYTE - OSTATNE CELOCISELNE CIsLA SU MENSIE AKO 256. VNUTORNE SU REPREZENTOVANE AKO JEDNOBYTOVE CIsLICE BEZ ZNAMiENKA (V ROZSAHU OD 0 DO 255), VDAKA TAKEJ REPREZENTACII BYTE MOZE BYT POUZIVANY AKO ZNAKOVA PREMENNA, KTORA MOZE SLUZIŤ AKO ZAMENA KLUCOVÝCH SLOV BYTE I CHAR.

CARD - PODORNE AKO BYTE, LEN Z VACsIM ROZSAHOM, VNUTORNA REPREZENTACIA JE DVOJBYTOVE CIsLO, TAKTIEZ PREMENNE TYPU CARD MOZU MAT HODNOTY Z ROZSAHU OD 0 DO 65535, CIsLA TYPU CARD SA ZAPAMATAVAJU VO FORMATE LSB, MSB.

INT - CELOCISELNE VYRAZY SO ZNAMiENKOM, REPREZENTACIA I ZAPIS AKO CARD. NAJvYSSI BIT JE INTERPRETOVANY AKO ZNAMiENKO; ROZSAH CISEL INT JE OD -32768 DO 32767.

DEKLARACIE

PRED POUZITIM KAZDEHO IDENTIFIKATORA JE MO TREBA ZADEKLAROVAT, ABY POCITAC REZERVOVAL ZODPOVEDAJUCE MIESTO V PAMATI. FORMAT DEKLARACIE JE NASLEDOVNY:

<TYP><IDENTIF.>[=<INFOR.POCIAT.>]I; <IDENTIF.>[=<INFOR.POCIAT.>]I; KDE <INFOR.POCIAT.> JE POCIATOCNA HODNOTA ALEBO ADRESA PREMENEJ; <ADRESA>[<HODNOTA>].

PRIKLADY:

BYTE CIsLO, POCET - DEKLARACIA PREMENNÝCH "CIsLO" A "POCET" TYPU BYTE

INT NUM=[0] - PREMENNA "NUM" TYPU INT O HODNOTE 0

BYTE X=#8000 - PREMENNA "X" TYPU BYTE UMIESTNENA V PAMATI POD ADRESOU #8000

CARD PKT=[0],

WSK=#83D4 - PREMENNA CARD "PKT" O HODNOTE 0 A "WSK" NA ADRESE #83D4. PREMENNE NEMUSIA BYT V TOM ISTOM RIADKU.

DEKLARACIE PREMENNÝCH SA MUSIA NACHADZAT V PROGRAME BEZPROSTREDNE PO INSTRUKCII MODULE ALEBO NA ZACIATKU PROCEDURY CI FUNKCIE.

PREHLAD PRiKAZOV EDITORA

PREJDENIE DO MONITURA

<CTRL><SHIFT><N>

PRIKAZY I/O

NACITANIE SUBORU	<CTRL><SHIFT>R=NAZOV(READ)
NACITANIE DIRECTORY	<CTRL><SHIFT>R-?Ni*.*(READ)
ZAPIS SUBORU	<CTRL><SHIFT>W=NAZOV(WRITE)
TLAC	<CTRL><SHIFT>W-P:(WRITE)

POHYB KURZORA

KURZOR HORE	<CTRL>↑
KURZOR DOLE	<CTRL>↓
KURZOR VPRAVO	<CTRL> →
KURZOR VLAVO	<CTRL> ←
KURZOR NA ZACIATOK RIADKU	<CTRL><SHIFT> <
KURZOR NA KONIEC RIADKU	<CTRL><SHIFT> >
NASLEDUJUCI RIADOK	<RETURN>
TABULACIA	<TAB>
NASTAVENIE TABULACIE	<SHIFT><TAB>
VYMAZANIE TABULACIE	<CTRL><TAB>

POSUV OKNA OBRAZOVKY

ZACIATOK SUBORU	<CTRL><SHIFT>H(CHEAD)
1 OBRAZOVKA HORE	<CTRL><SHIFT>↑
1 OBRAZOVKE DOLU	<CTRL><SHIFT>↓
1 ZNAK VLAVO	<CTRL><SHIFT>]
1 ZNAK VPRAVO	<CTRL><SHIFT>[
VYTVORENIE DRUHEHO OKNA	<CTRL><SHIFT>2
PREJDENIE DO PRVEHO OKNA	<CTRL><SHIFT>1
PREJDENIE DO DRUHEHO OKNA	<CTRL><SHIFT>2
VYMAZANIE OKNA	<CTRL><SHIFT>D(DELETE)

EDITOVANIE TEXTU

VKLADANIE/VY MENA	<CTRL><SHIFT>I(INSERT)
VRATENIE ZMENENEO RIADKU	<CTRL><SHIFT>U(UNDONE)
VRATENIE VYMAZANEO RIADKU	<CTRL><SHIFT>P(PASTE)
ZAPAMATANIE BLOKU TEXTU	<SHIFT><DELETE>
VLOZENIE TEXTU Z BAFRA	<CTRL><SHIFT>P(PASTE)
HĽADANIE RETAZCA	<CTRL><SHIFT>F(FIND)
VY MENA RETAZCA	<CTRL><SHIFT>S(SUBSTITUTE)
ROZDELENIE RIADKA NA DVA	<CTRL><SHIFT><RETURN>
SPOJENIE DVOCH RIADKOV	<CTRL><SHIFT><DELETE>
NASTAVENIE ZNACKY	<CTRL><SHIFT>T(TAG SET)
HĽADANIE ZNACKY	<CTRL><SHIFT>G(GO TO TAG)

PREHLAD PRIKAZOV MONITORA

RESTART ACTIONI	B(BOOT)
PREJDENIE DO DOS=U	D(DOS)
PREJDENIE DO EDITORA	E(EDITOR)

MENU VARIANTOV	O(OPTIONS)
VYKONANIE INSTRUKCIE	X<INSTRUKCIA>(XECUTE)
KOMPILACIA PROGRAMU	C["NAZOV"](COMPILE)
ZAPIS PROGRAMU	W["NAZOV"](WRITE)
SPUSTENIE PROGRAMU	R["NAZOV"](RUN)
POKRACOVANIE PROGRAMU	P (PROCEED)
NASTAVENIE OBSAHU(POKE)	SET<ADRESA>=<OBSAH>
ZISTENIE OBSAHU (PEEK)	?<ADRESA>
PREHLAD PAMATI	*<ADRESA>

CHYBOVE OZNAMY ACTIONI

KOD	VYZNAM
0	MALY ROZSAH DOSTUPNEJ PAMATI
1	NEDEFINOVANY ZNAK (") V RETAZCI
2	ZAHNIEZDENA DIREKTIVA DEFINE
3	CHYBA UMIESTNENIA V TABULKE SYMBOLOV GLOBALNYCH PREMENNÝCH
4	CHYBA UMIESTNENIA V TABULKE SYMBOLOV LOKALNYCH PREMENNÝCH
5	CHYBA SKLADBY V DIREKTIVE SET
6	NESPRAVNY FORMAT DEKLARACIE
7	PRIVELA ARGUMENTOV V INSTRUKCII ALEBO PROCEDURE
8	NEDEKLAROVANA PREMENNA
9	PREMENNA POUZITA NA MIESTE KDE SA OCAKAVA KONSTANTA
10	NEPRIPUSTNA INSTRUKCIA
11	NEROZPOZNANA CHYBA
12	CHYBA INSTRUKCIE THEN
13	CHYBA INSTRUKCIE FI
14	MALO MIESTA PAMATI PRE VYKONATELNY MODUL
15	CHYBNA INSTRUKCIA DO
16	CHYBNA INSTRUKCIA TO
17	NESPRAVNY FORMAT VYRAZU
18	NEUZAMKNUTY NAWIAS
19	CHYBNA INSTRUKCIA OD
20	NIE JE MOZNO PRESUNUT VIAC PAMATI
21	NEPRIPUSTNA TABULKA
22	PRIDLHY VSTUPNY SUBOR
23	NEPRIPUSTNY FORMAT VYRAZU
24	CHYBNE ZLOZENIE V INSTRUKCII FOR
25	NEPRIPUSTNA INSTRUKCIA EXIT
26	PRIDLHE ZAHNIEZDENIE (DOVOLENE 16 UROVNI)
27	CHYBNE ZLOZENIE V INSTRUKCII TYPE
28	NEPRIPUSTNA INSTRUKCIA RETURN
69	CHYBNE UMIESTNENIE V TABULKE SYMBOLOV
128	ZASTAVENIE PROGRAMU KLAVESOU BREAK

PROGRAMOVANIE V JAZYKU ACTIONI

JAZYK ACTIONI SI MOZME PREDSTAVIT AKO ZMES Z VIACERYCH JAZYKOV; JEDNODUCHOSTOU VYUZIVANIA HO MOZME POROVNAT S INTERPRETEROM BASIC-A, STRUKTUROU PRIPOMINA PASCAL A MOZNOSTAMI, SKORO AKO STROJOVO ORIENTOVANYM PROGRAMOVANIM, K SPECIALIZACII JAZYKA C. V NEPOSLEDNOM RADE JE RYCHLOST VYTvoreNEHO PROGRAMU POROVNATELNA LEN S RYCHLOSTOU PROGRAMU VYTvoreNEHO V ASSEMBLERI. JAZYK ACTIONI JE STRUKTUROVANY JAZYK. NIEJE OVSEM NUTNE SA BAT TROCHU STRAPATEHO ZDROJOVEHO VYPISU. TATO PROGRAMOVACIA METODA PRINASA VLASTNE LEN SAME VYHODY; STRUKTUROVANY ZNAMENA, ZE PROGRAM JE STRIKTNE DELENY NA MODULY, KTORE MAJU SVOJ VLASTNY VSTUP A VYSTUP. STRUKTUROVANE PROGRAMY POZADUJU OKREM INEHO AJ DEFINOVANIE VSETKYCH PREMENNÝCH PRED ICH POUZITIM.

UKAZEME SI NIEKOLKO PRIKLADOV NA ROZDIELOCH S JAZYKOM BASIC. NAPRIKLAÐ NAPROGRAMOVANIE NEKONECNEJ SLUCKY:

```
10 PRINT "HALLO"
20 GOTO 10
```

```
V ACTIONI TO VYZERA TAKTO:
DO
PRINT("HALLO")
OD
```

VSETKY PRIKAZY, KTORE SU UZAVRETE MEDZI DO A OD SA OPAKUJU DONEKONECNA; NA UKONCENIE OPAKOVANYCH PRIKAZOV EXISTUJE YELA MOZNOSTI; JEDNOU Z NICH, KTORA JE PEDANTNE SPRACOVANA V BASIC-U JE FOR=SLUCKA:

```
10 FOR I=1 TO 10
20 PRINT I
30 NEXT I
```

```
V ACTIONI JE TO NASLEDOVNE:
BYTE I
FOR I=1 TO 10
DO
PRINTBE(I)
OD
```

AKO SME SA PRAVE PRESVEDCILI, MUSIME PREMENNÚ PRED POUZITIM DEFINOVAT. PRETO SME POUZILI BYTE-DEKLARACIU. ZAUJIMAVY JE TIEZ PRINT-PRIKAZ. PRINTBE JE FUNKCIA, KTOREJ VYSTUPOM JE BYTE-HODNOTA SPOLU S PREJDENIM NA NOVY RIADOK. PRI VYUZIVANI CÍSLA JE TEDA DOLEZITE PRESNE VEDIET, AKEHO TYPU JE VYUZIVANA PREMENNA. INYM POUZITIM DO-OD BLOKU JE NAPR.:

```
BYTE S
DO
S=STICK(0)
UNTIL S=14
OD
```

AK CHCETE TENTO PROGRAM VYUZIVAT V BASIC-U, POTOM MUSITE PRACOVAT S PRIKAZOM GOTO A TYM SA STRATILA JASNA STRUKTURA PROGRAMU:

```
10 S=STICK(0)
20 IF S=14 THEN 40
30 GOTO 10
40 ...
```

SAMOZREJME MOZME TO NAPISAT AJ INAC, ALE TU UZ IDE O PRINCIP, PROGRAMY V ACTIONI SU V DAKA PEVNEJ STRUKTURE JEDNODUCHO LEPSIE CITATELNE, PRIKAZ IF V BASIC-U JE OHRANICENY NA JEDEN RIADOK, AVSAK MEDZI IF=FI BLOKY V ACTIONI MOZME UMIESTNIT LUBOVOLNE MNOZSTVO PRIKAZOV, A KEDZE SU K DISPOZICII AJ PRIKAZY AKO ELSE A ELSEIF, ZOSTAVA V PROGRAMOVANI V ACTIONI PRIKAZ GOTO TAKMER NEPOTREBNY:

```
IF STICK(0)=14 THEN HORE()
ELSEIF STICK(0)=11 THEN DOLE()
ELSE BEZPOHYBU()
FI
```

TENTO JEDNODUCHY PRIKLAD NA RIADENIE JOYSTICKU BUDE V BASIC-U VYZERAT NASLEDOVNE:

```
10 IF STICK(0)=14 THEN GOSUB 1000:GOTO 40:REM HORE
20 IF STICK(0)=11 THEN GOSUB 2000:GOTO 40:REM DOLE
30 GOSUB 3000: REM ZIADEN POHYB
40 ...
```

AKO VIDITE, POUZITIM MENA PODPROGRAMOV SA PROGRAM SAM DOKUMENTUJE (POKIAL NEPOUZIEME NEMNEMONICKE NAZVY), ODPADA NAM POTOM POUZITIE AKEHOKOLVEK KOMENTARU,

MIMOTO ACTIONI DOVOLUJE PRIAMY PRISTUP K PAMATI, KDE MOZME PREMENNE AKO AJ POLIA PREMENNÝCH PRIAMO ULOZIT DO URCITEJ BUNKY PAMATE:

```
BYTE CHBAS=756
BYTE ARRAY COLOR(4)=0204
```

POMOCOU TEJTO DEFINICIE MOZME PRIAMO MENIT ZNAKOVU SADU KLAVESNICE ALEBO FARBU, S HODNOTOU CHBAS=204 MOZME NAPR. AKTIVOVAT MEDZINARODNU ZNAKOVU SADU (V BASIC-U:POKE 756,204). COLOR(0)=255 PREPNE FARBOVY REGISTER 1 NA BIELU FARBU, TAKETO KONSTRUKCIE SA VYKONAVAJU V ACTIONI ZA PRVE ELEGANTNE A ZA DRUHE TIEZ VELMI RYCHLO, AK POZNATE TROCHU ASSEMBLER A POZRIETE SI VYTVORENY KOD Z PRIKAZU CHBAS=204, POTOM UVIDITE NASLEDOVNE:

```
LDA #204
STA 02F4
```

RYCHLEJSIE SA TO UZ NAOZAJ NE DA USKUTOCNIT.

AKO UZ BOLO VYSSIE SPOMENUTE PROGRAMY V ACTIONI SA SKLADAJU Z CASTI, KTORE SA NAZYVAJU PROCEDURY A FUNKCIE. AJ HLAVNY PROGRAM JE SPRAVIDLA PROCEDURA, KTORA VYVOLAVA OSTATNE PROCEDURY A FUNKCIE A TAKTO RIADI BEH CELEHO VYPOCTU, VYVOLAVANE PROCEDURY A FUNKCIE

MUSIA BYT ZADefINOVANE PRED ICH VYVOLANIM Z HLAVNEJ PROCEDURY V PREDCHADZAJUCEJ CASTI PROGRAMU, AK TIETO PROCEDURY MAME ULOZENE NA VONKAJSEJ PAMATI, POTOM ICH NEMUSIME ZNOVA PISAT DO PROGRAMU, ALE STACI AK ICH VYVOLAME ZO ZARIADENIA DO PROGRAMU PRIKAZOMI:

```
INCLUDE "D1:MENO_PROGRAMU:EXT" - PRI VOLANI Z DISKETY
INCLUDE "C:" - PRI VOLANI Z KAZETY;
PRI VOLANI Z KAZETY SA MUSIME SAMOZREJME NASTAVIT NA ZACIATOK
VOLANEJ PROCEDURY
```

ROZDIEL MEDZI PROCEDUROU A FUNKCIOU JE TEN, ZE VYSLEDOK VOLANIA FUNKCIE JE HODNOTA ARGUMENTU FUNKCIE A PRETO SA VOLA TAM, KDE SA POZADUJE JEJ HODNOTA AKO VYSLEDOK NEJAKEJ OPERACIE. PROCEDURA MOZE DAVAT AKO VYSLEDOK VIAC PARAMETROV, KTORE SA ZADALI PRI VYVOLANI PROCEDURY. PRIKLAD FUNKCIE, KTOREJ VSTUPOM JE CELE CISLO A VYSTUPOM JE ABSOLUTNA HODNOTA TOHTO CISLA JE:

```
INT FUNC ABS(INT N) ; HLAVICKA FUNKCIE ABS
IF N<0 THEN ; ZACIATOK PODMIENKY IF
RETURN(-N)
FI ; KONIEC PODMIENKY BLOKU IF=FI
RETURN(N)
```

AKO VIDIME AJ SAMOTNA FUNKCIA MUSI BYT DEKLAROVANA A PRETO AKO PRVY JE V HLAVICKE DEFINICIE FUNKCIE URCENY TYP FUNKCIE INT, CARD ALEBO BYTE. VYSLEDKOM Z FUNKCIE JE POTOM HODNOTA ARGUMENTU UVEDENA V PRIKAZE RETURN(), ZA ZNAKOM ; PISEME V ACTIONI LUBOVOLNE POZNAMKY.

VYVOLANIE TEJTO FUNKCIE SI UVEDIEME V NASLEDOVNEJ PROCEDURE NA KRESLENIE KRuhu:

```
PROC KRuh(INT X,Y,R,C) ; HLAVICKA PROCEDURY KRuh S PARAMETRAMI
; X,Y,R,C TYPu INTEGER
INT PHI,PHIY,PHIXY, ; DEKLARACIE LOKALNYCH PREMENNÝCH
X1,Y1 ; ZACIATOK TELA PROCEDURY

PHI=0
X1=R
Y1=0
COLOR=C
DO ; ZACIATOK CYKLU DO
PHIY=PHI+Y1+Y1+1
PHIXY=PHIY-X1-X1+1
PLOT(X+X1,Y+Y1)
PLOT(X-X1,Y-Y1)
PLOT(X+X1,Y-Y1)
PLOT(X-X1,Y+Y1)
PLOT(X+Y1,Y+X1)
PLOT(X-Y1,Y+X1)
PLOT(X+Y1,Y-X1)
PLOT(X-Y1,Y-X1)
PHI=PHIY
Y1=Y1+1
IF ABS(PHIXY)+0<ABS(PHIY) THEN ;VYVOLANIE FUNKCIE ABS
PHI=PHIXY ;S PARAMETROM PHIXY A PHIY
X1=X1-1
FI
```

```

UNTIL Y1>X1          ; PODMIENKA NA OPUSTENIE CYKLU DO-OD
OD                  ; KONIEC CYKLU DO
RETURN              ; KONIEC PROCEDURY KRUH
MODULE              ; OZNACUJE HLAVNU PROCEDURU

```

TATO PROCEDURA ESTE SAMA V TAKOM TVARE AKO JE SA NEDA VYKONAT, PARAMETRE X,Y,R,C MUSIA VSTUPOVAT DO PROCEDURY Z INEJ, HLAVNEJ, PROCEDURY AKO SKUTOCNE PARAMETRE. V PRIKAZE IF SA VYKONAVAJU PRIKAZY, KTORE VYHOVUJU DANEJ PODMIENKE ZA THEN AZ PO KLUCOVE SLOVO FI UKONCUJE TENTO PRIKAZ; PODMIENKA UNTIL SLUZI NA VYBEHNUTIE Z CYKLU AK SA SPLNI PODMIENKA, KTORA JE V NEJ UVEDENA, CIZE POKIAL Y1 JE MENSIE ALEBO ROVNE AKO X1 CYKLUS ZACINA OD KLUCOVEHO SLOVA DO.

VYVOLANIE PROCEDURY KRUH Z HLAVNEJ PROCEDURY MOZE BYT NAPR.:

```

INCLUDE "DIKRUH.ACT" ; DOTIAHNUTIE PROCEDURY KRUH Z DISKU
                      ; POD NAZVOM KRUH,ACT
PROC KRESLIKRUH()    ; HLAVICKA HLAVNEJ PROCEDURY
  BYTE R             ; DEKLARACIA LOKALNEJ PREMENNEJ
  GRAPHICS(8)        ; NASTAVENIE GRAFICKEHO MODU
  SETCOLOR(2,0,0)    ; NASTAVENIE FAREBNEHO POZADIA
  FOR R=0 TO 75 STEP 5
    DO                ; VYVOLANIE PROCEDURY S PARAMETRAMI
      KRUH(160,80,75-R,1)
    OD
  DO                  ; ZASTAVENIE KRESLENIA NEKONECNYM
  OD                  ; CYKLOM DO-OD
RETURN               ; KONIEC HLAVNEJ PROCEDURY

```

TAKTO BY OVSEM ESTE PROGRAM NEFUNGOVAL, PRETOZE BY NEOBSAHOVAL FUNKCIU ABS, KTORU BY MAĽA OBSAHOVAT PROCEDURA KRUH ALEBO JU MOZME "INKLUDOVAT" Z DISKETY ALEBO PASKY PRIKAZOM INCLUDE, AK JU TAM RAZ MAJE ULOZENU. KLUCOVE SLOVO MODULE SA DOTIAHNE PRED HLAVNU PROCEDURU KRESLIKRUH SPOLU S PROCEDUROU KRUH A OZNACUJE ZVYCAJNE ZACIATOK HLAVNEJ PROCEDURY. ESTE RAZ PRIPOMINAM, ZE VSETKY PROCEDURY A FUNKCIE, KTORE SA VYVOLAVAJU Z HLAVNEJ PROCEDURY MUSIA BYT DEFINOVANE PRED ICH VOLANIM.

NA BLIZSI POPIS SYNTAXE JAZYKA ACTION! ODPORUCAM REFERENCNY MANUAL JAZYKA ACTION! ALEBO PRIRUCKU PROGRAMATORA, BEZ KTOREJ SA PRI ZLOZITEJSOM PROGRAMOVANI NEZAOBIDETE; TATO PRIRUCKA MAĽA SLUZIT IBA AKO POMOCKA PRI UVODNOM ZOZNAMENI SA S NEPOCHYBNE JEDNYM Z NAJLEPSICH JAZYKOV PRE POCITACE ATARI XL/XE.

```
INCLUDE "SORT.ACT"
```

```
PROC Test()
```

```
    BYTE LMargin=$52
```

```
    CARD i
```

```
    BYTE ARRAY d(500)
```

```
    LMargin=0
```

```
    FOR i=0 TO 499
```

```
        DO
```

```
            d(i)=Rand(0)
```

```
        OD
```

```
    PrintE("Unsorted")
```

```
    FOR i=0 TO 499 STEP 10
```

```
        DO
```

```
            Printf("%B %B %B %B %B ",d(i),d(i+1),d(i+2),d(i+3),d(i+4))
```

```
            Printf("%B %B %B %B %B%E",d(i+5),d(i+6),d(i+7),d(i+8),d(i+9))
```

```
        OD
```

```
    PrintE("Ascending")
```

```
    SortB(d,500,0)
```

```
    FOR i=0 TO 499 STEP 10
```

```
        DO
```

```
            Printf("%B %B %B %B %B ",d(i),d(i+1),d(i+2),d(i+3),d(i+4))
```

```
            Printf("%B %B %B %B %B%E",d(i+5),d(i+6),d(i+7),d(i+8),d(i+9))
```

```
        OD
```

```
    ;You can see how much slower this sorting
```

```
    ;pass is due to the fact the data is
```

```
    ;presorted.
```

```
    PrintE("Descending")
```

```
    SortB(d,500,1)
```

```
    FOR i=0 TO 499 STEP 10
```

```
        DO
```

```
            Printf("%B %B %B %B %B ",d(i),d(i+1),d(i+2),d(i+3),d(i+4))
```

```
            Printf("%B %B %B %B %B%E",d(i+5),d(i+6),d(i+7),d(i+8),d(i+9))
```

```
        OD
```

```
RETURN
```

```
;*****  
;SORT.ACT - a group of sorting routines  
;which allow you to sort BYTE, CARD,  
;INT, or string data. The algorithm  
;used is called a quicksort, and is  
;very fast (order  $N \ln N$ ) on unsorted  
;data but is almost as slow as bubble  
;or shell sort on presorted data  
;(worst case order  $N^2$ ).
```

```
; Copyright (c) 1984 by OSS, Inc.,  
; Mark Rose, and Mike Fitch  
;*****
```

MODULE

```
; Despite what your Toolkit manual says,  
; the definition of SortMax is ignored  
; by the sort routines does not need  
; to be changed to match your data size.  
; (This is due to an improvement in the  
; way the sort routines work)  
DEFINE SortMax="10000" ;change value to suit yourself  
  
CARD ListSize  
  
BYTE ARRAY BArray  
  
INT ARRAY IArray  
  
CARD ARRAY CArray,  
List(64) ; big enough for 64K sort elements  
  
;*****  
;BYTE Comparisions  
;*****  
  
BYTE FUNC BDescend(CARD i,j)  
IF BArray(i)>BArray(j) THEN  
RETURN (1)  
FI  
RETURN (0)  
  
;*****  
  
BYTE FUNC BAscend(CARD i,j)  
IF BArray(i)<BArray(j) THEN  
RETURN (1)  
FI  
RETURN (0)  
  
;*****  
;CARD Comparisions  
;*****  
  
BYTE FUNC CDescend(CARD i,j)  
IF CArray(i)>CArray(j) THEN  
RETURN (1)  
FI  
RETURN (0)
```

```

;*****
BYTE FUNC CAscend(CARD i,j)

  IF CArray(i)<CArray(j) THEN
    RETURN (1)
  FI
RETURN (0)

;*****
;INT Comparisons
;*****

BYTE FUNC IDescend(CARD i,j)

  IF IArray(i)>IArray(j) THEN
    RETURN (1)
  FI
RETURN (0)

;*****

BYTE FUNC IAscend(CARD i,j)

  IF IArray(i)<IArray(j) THEN
    RETURN (1)
  FI
RETURN (0)

;*****
;String Comparisons
;*****

BYTE FUNC SDescend(CARD i,j)

  IF SCompare(CArray(i),CArray(j))>0 THEN
    RETURN (1)
  FI
RETURN (0)

;*****

BYTE FUNC SAscend(CARD i,j)

  IF SCompare(CArray(i),CArray(j))<0 THEN
    RETURN (1)
  FI
RETURN (0)

;*****
;Interchange 2 BYTE sort elements
;*****

PROC BSwap(CARD i,j)

  BYTE temp

  temp=BArray(i)
  BArray(i)=BArray(j)
  BArray(j)=temp
RETURN

;*****
;Interchange 2 sort elements of CARD
;or string types
;*****

```

```

PROC CSwap(CARD i,j)

    CARD temp

    temp=CArray(i)
    CArray(i)=CArray(j)
    CArray(j)=temp
RETURN

;*****
;Interchange 2 INT sort elements
;*****

PROC ISwap(CARD i,j)

    INT temp

    temp=IArray(i)
    IArray(i)=IArray(j)
    IArray(j)=temp
RETURN

;*****
;The following two routines are set
;to one of the compare and swap routines
;above, depending on the type of data
;being sorted and the order of the sort.
;*****

BYTE FUNC Compare(CARD i,j)

PROC Swap(CARD i,j)

;*****
;Add a partition to the list
;*****

PROC AddList(CARD low,high)

    IF high+1>low+1 THEN
        List(ListSize)=low
        ListSize==+1
        List(ListSize)=high
        ListSize==+1
    FI
RETURN

;*****
;Retrieve last low,high pair from list
;of partitions
;*****

PROC GetFirst(CARD POINTER lowP,highP)

    ListSize==-1
    highP^=List(ListSize)
    ListSize==-1
    lowP^=List(ListSize)
RETURN

;*****
;Divide sort array into partitions
;*****

```

CARD FUNC Partition(CARD low,high)

```
CARD i,j,pivot,mid

; Find median of 1st,middle,and last
; elements to use as pivot for partitioning.
mid=(low+high) RSH 1
IF Compare(mid,low) THEN
  Swap(low,mid)
FI
IF Compare(high,low) THEN
  Swap(low,high)
FI
IF Compare(mid,high) THEN
  Swap(mid,high)
FI
pivot=high
i=low
j=high
WHILE i<j
  DO
    WHILE Compare(i,pivot)
      DO
        i==+1
      OD
    WHILE (Compare(j,pivot)=0),AND (j>i)
      DO
        j=-1
      OD
    IF i<j THEN
      Swap(i,j)
    FI
  OD
  Swap(i,high)
RETURN (i)
```

```
*****
;Sort using the QuickSort algorithm
*****
```

PROC QuickSort(CARD len)

```
CARD low,middle,high,i

ListSize=0
AddList(0, len-1)
WHILE ListSize>0
  DO
    GetFirst(@low,@high)
    middle=Partition(low,high)
    ; Put larger partition onto stack first
    ; in order to decrease maximum stack size.
    IF (middle-low) > (high-middle) THEN
      AddList(low,middle-1)
      AddList(middle+1,high)
    ELSE
      AddList(middle+1,high)
      AddList(low,middle-1)
    FI
  OD
RETURN
```

```
*****
;Sort a BYTE ARRAY
```

```
PROC SortB(BYTE ARRAY data CARD len BYTE order)
```

```
    IF order THEN  
        Compare=BDescend  
    ELSE  
        Compare=EAscend  
    FI  
    Swap=BSwap  
    BArray=data  
    QuickSort(len)  
RETURN
```

```
;*****  
;Sort a CARD ARRAY  
;*****
```

```
PROC SortC(CARD ARRAY data CARD len BYTE order)
```

```
    IF order THEN  
        Compare=CDescend  
    ELSE  
        Compare=CAscend  
    FI  
    Swap=CSwap  
    CArray=data  
    QuickSort(len)  
RETURN
```

```
;*****  
;Sort an INT ARRAY  
;*****
```

```
PROC SortI(INT ARRAY data CARD len BYTE order)
```

```
    IF order THEN  
        Compare=IDescend  
    ELSE  
        Compare=IAscend  
    FI  
    Swap=ISwap  
    IArray=data  
    QuickSort(len)  
RETURN
```

```
;*****  
;Sort a CARD ARRAY whose elements are  
;the addresses of strings  
;*****
```

```
PROC SortS(CARD ARRAY data CARD len BYTE order)
```

```
    IF order THEN  
        Compare=SDescend  
    ELSE  
        Compare=SAscend  
    FI  
    Swap=CSwap  
    CArray=data  
    QuickSort(len)  
RETURN
```

```
MODULE
```

SEQUENCER MIDI ATARI 130 XE (800 XL) REAL TIME

-UVOD-

SEQUENCER JE ZARIADENIE, KTORE SA POUZIVA PRI PRODUKCII HUDBY. UMOZNUJE ZAZNAM HUDOBNEJ SKLADBY, KTORÝ JE POTOM V LUBOVOLNOM CASE REPRODUKOVATELNY. NIEKTORE HUDOBNE ELEKTRONICKE NASTROJE "SYNTETIZATORY" MAJU TAKYTO SEQUENCER ZABUDOVANÝ PRIAMO V NASTROJI (POLLY 800).

ZAZNAM (PROGRAMOVANIE) DO SEQUENCERA JE MOZNE PO KROKU (STEP), ALEBO JE MOZNE VYTVOŘIT ZAZNAM V REALNOM CASE. VTEDY NIE JE NUTNE POCITAT DLZKU NOT ALE JE NUTNE HRAT NA NASTROJ PRESNE.

OVLADANIE NASTROJA (SYNTETIZATORA) JE MOZNE AJ EXTERNE AK JE VYBAVENÝ "MIDI" VSTUPOM. MIDI JE DOHODNUTÝ SERIOVÝ PRENOS DAT S PEVNE DANYM FORMATOM. TO UMOZNUJE NAPRIKĽAD OVLADANIE VIACERYCH NASTROJOV JEDNYM HRACOM CEZ JEDNU KLAVESNICU NASTROJA.

JE TEDA MOZNE PREPOJENIE VIACERYCH ELEKTRONICKYCH NASTROJOV POMOCOU MIDI PREPOJENIA. SAMOZREJME KAŽDY Z NICH MUSI BYT VYBAVENÝ MIDI VSTUPOM A VYSTUPOM. AK TERAZ NAHRADIME HRACA NA NASTROJ SEQUENCEROM, JE MOZNE PRODUKOVAT HUDBU BEZ HRACA. TATO MOZNOST SA VYUŽIVA HLAVNE PRI STUDIOVÝCH NAHRÁVKACH POPULARNEJ HUDBY. U NAS JE TO HLAVNE VASO PATEJDL, KTORÝ SEQUENCER CASTO VYUŽIVA (CASTO AJ NA KONCERTOCH).

SEQUENCER JE MOZNE VYTVOŘIT AJ S POCITACA. NAPRIKĽAD ATARI ST 520 JE PRIAMO VYBAVENÝ MIDI INTERFACE, KTORÝ SO SOFTWAREM NAHRADI KLASICKE SEQUENCERY. FRICOM PRACA S TAKYMTA VYBAVENIM JE VELKÝM KONFORTOM.

PRENOS DAT SYSTEMOM MIDI

SYSTEM MIDI UMOZNUJE PREPOJENIE 16 NEZAVISLYCH KANALOV (HUDOBNYCH NASTROJOV). TO ZNAMENA, ŽE JE MOZNE V PODSTATE VYTVOŘIT V DOMACICH PODMIENKACH DIGITALNE NAHRÁVACIE STUDIO.

PRENOS DAT MEDZI JEDNOTLIVYMI NASTROJMI JE SERIOVY S PRUDOVOU SLUCKOU. PRICOM NASTROJE SU NAVZAJOM ODDIENE OPTOCLENMI Z DOVODU VZNIKU NEPRIJEMNEHO "BRUMU".

SERIOVY PRENOS DAT

SERIOVY PRENOS DAT JE CHARAKTERIZOVANY PRENOSOVOU RYCHLOSTOU, KTORA SA VYJADRUJE POCOM PRENESENÝCH BITOV ZA SEKUNDU - BAUDOCH. PRICOM TENTO PRENOS MOZE BYT SYNCHRONNY, ALEBO ASYNCHRONNY. MIDI POUZIVA PRENOS ASYNCHRONNY S RYCHLOSTOU 31250 BAUDOV.

START	7	6	5	4	3	2	1	0	STOP
BIT									BIT

1 BYTE

FORMAT PRENOSU A SKLADA: START BIT, 8 DATOVÝCH BITOV A STOP BIT. DLZKA JEDNEHO BITU $T_0=1/31.25$ KHZ $\Rightarrow T_0=32$ US. CELKOVA DLZKA PRENOSU JEDNEJ INFORMACIE JE TEDA $T=10 \cdot T_0 \Rightarrow T=320$ US

KODOVANIE INFORMACII

KAZDY TON MA SVOJ ZACIATOK, KONIEC, VYSKU, DYNAMIKU A PATRI K NEJAKEMU ZDROJU ZVUKU (HUDOBNEMU NASTROJU). AK TEDA CHCEME PRODUKOVAT HUDBU NA VIACERYCH HUDOBNYCH NASTROJOCH MUSI BYT DANE KU KTOREMU HUDOBNEMU NASTROJU DANY TON PATRI. Z TOHTO VYPLYVA MNOZSTVO POTREBNÝCH INFORMACII: VOLBA HUDOBNEHO NASTROJA, VYSKA TONU, DYNAMIKA TONU, ZACIATOK (KONIEC) TONU.

MIDI POUZIVA DVA DRUHY KODOVÝCH BYTOV: TZV STATUS BYTE A DATA BYTE. STATUS BYTE SLUZI NA OSLOVENIE ZARIADENIE (VYBRATIE HUDOBNEHO NASTROJA) A DATA BYTE SLUZI NA DEFINOVANIE PARAMETROV: VYSKA TONU, DYNAMIKA TONU. KAZDY PRENOS SA TEDA MINIMAL-

NE SKLADA Z JEDNEHO STATUS BYTE A DVOCH DATA BYTE. STATUS BYTE MA UZDY 7 BIT NASTAVENY NA JEDNOTKU A DATA BYTE MA UZDY 7 BIT NASTAVEVENY NA NULU. TEDA 7 BIT URČUJE CI SA JEDNA O STATUS ALEBO DATA BYTE.

AK TEDA NA SYNTETIZATORE STLACIME JEDNU KLAVESU Z MIDI OUT SA VYSLU TRI BYTY: STATUS BYTE OBSAHUJE KOD KANALA 0-16, DATA BYTE - VYSKA TONU, DATA BYTE - DYNAMIKA TONU. AK POTOM KLAVESU PUSTIME SYNTETIZATOR VYSLE ZNOVA TRI BYTE, PRICOM STATUS BYTE A DATA BYTE-VYSKA TONU BYVAJU ZPRAVIDLA ZHODNE AKO PRI STLACENI KLAVESY (NIE PRI VSETKYCH DRUHOCH SYNTETIZATOROCH).

POSTUPNOST VYSIELANIA MIDI:

1. BYTE	2. BYTE	3. BYTE
STATUS BYTE	DATA -VYSKA TONU BYTE	DATA -DYNAMIKA BYTE

AK SUCASNE STLACIME VIAC KLAVES NA SYNTETIZATORE POTOM SA ZPRAVIDLA VYSIELA JEDEN STATUS BYTE (URČENIE KANALA MIDI) A PRISLUSNY POCET DATA BYTE : PRE KAZDY TON 2 DATA BYTE (DATA BYTE-VYSKA TONU A DATA BYTE-DYNAMIKA). AK TEDA ZAHRAJE AKORD (3 TONY) VYSLE SYNTETIZATOR SPOLU 7 BYTOV. PODOBNA SITUACIA NASTANE AK PRESTANEME AKORD HRAT. NIEKTORE SYNTETIZATORY (ROLAND ALFA 1) VYSIELAJU PO PUSTENI VSETKYCH KLAVES ESTE KOD, ZE NIE STLACENA ZIADNA KLAVESA -NIE VSETKY (KORG POLLY 800).

POUZITIE POCITACA AKO SEQUECER.

ULOHOU SEQUENCERA (POCITACA) JE ZAZNAMENAT SPOMINANE INFORMACIE, KTORE VYSIELA SYNTETIZATOR A SPETNE ICH AK JE ZIADUCE REPRODUKOVAT. V PODSTATE SU DVE MOZNOSTI (SPOSOBY) ZAZNAMU INFORMACII: TZV. STEP A REAL TIME. ZAZNAM STEP - PO KROKU SA

REALIZUJE POSTUPNYM VKLADANIM INFORMACII, PRICOM VOBEC NEZALEZI NA PRESNOSTI HRANIA NA HUDOBNY NASTROJ Z HLADISKA DLZKY TRVANIA NOT. JE VSAK NUTNE PRI TOMTO SPOSOBE VKLADANIA INFORMACII PRAKTICKY VZDY POUZIVAT NOTOVY ZAPIS Z DOVODU ORIENTACIE. TENTO SPOSOB PRI ZNALOSTI NOTOVEHO ZAPISU, ORIENTACII NA KLAVESNICI HUDOBNEHO NASTROJA A TRPEZLIVOSTI DOVOLUJE PRAKTICKY AJ NIE CELKOM MUZIKANTOM HRAT PRAKTICKY "VSETKO". AKO SOLIDNELSI SA JAVI SYSTEM REAL TIME, KTORÝ UMOZNUJE PRESNE REPRODUKOVAT HRU NA NASTROJ. SAMOZREJME JE TU NUTNE HRAT NA NASTROJ POMERNE PRESNE. AJ KED SAMOZREJME SA DA TENTO ZAZNAM DODATOCNE UPRAVOVAT (MENIT TEMPO, TRANSPONOVAT ATD.).

AK TEDA POUZIJEME SPOSOB ZAZNAMU REAL TIME (REALNY CAS) JE NUTNE OKREM INFORMACII, KTORE BOLI UZ SPOMENUTE ZAZNAMENAT AJ CAS V KTOROM BOLI TIETO INFORMACIE VYSLANE SYNTETIZATOROM. PRAKTICKY SA VYUZIVAJU PRE TENTO UCEL DALSIE DVA BYTY. TO ZNAMENA, ZE POTREBUJEME PRAKTICKY 5 BYTOV NA ZAZNAM JEDNEHO STLACENIA (PUSTENIA) JEDNEJ KLAVESY. SAMOZREJME AK ZAHRAJE SUCASNE 3 TONY NEPOTREBUJEME $3.5=15$ BYTOV, ALE PRAKTICKY 3.2 BYTOV (VYSKA, DYNAMIKA) + 1 BYTE (STATUS) + 2 BYTY (TZV. TIMER) = 9 BYTOV. Z TYCHTO POCTOV SA DA VYPOCITAT AKE JE PRIBLIZNE MNOZSTO HUDOBNEHO ZAZNAMU POCITACA. AK PREDPOKLADAME ZE, PRIBLIZNA DLZKA PROGRAMU BUDE ASI 5 KBYTE POTOM PRE ATARI 800 XL VOLNA KAPACITA BUDE 37 KBYTE (VOLNA KAPACITA V BASICU) + 22 KBYTE (RAMDISK POD OS A BASIC) - 5 KBYTE (PROGRAM) = 54 KBYTE = $54 \cdot 1024 = 55\ 296$ BYTOV. Z TOHO $55296:5=11\ 060$ MOZNYCH NOT. SAMOZREJME ,ZE TENTO POCET JE LEN PRIBLIZNY (TAKTO SA VSAK UDAVA AJ PRI FIREMNYCH SEQUENCEROCH). PRE ATARI 130 XE JE KAPACITA PAMETE ZOZSIRENA O DALSICH 64 KBYTE (65535 BYTOV), PRETO CELKOVA VOLNA KAPACITA JE PRIBLIZNE 118 KBYTE = $120\ 832$ BYTOV => $120832/5=24\ 166$ MOZNYCH NOT. SAMOZREJME PRI POUZITI DISKETOVEJ MECHANIKE JE TATO KAPACITA PRAKTICKY NEOBMEDZNENA. TAKTIEZ TO PLATI PRI POUZITI MAGNETICKEJ PASKOVEJ MECHANIKE, AVSAK Z PRAKTICKEHO HLADISKA PRI PRODUKCII HUDBY JE VYUZITIE MAGNETOFONU POMERNE TAZKOPADNE. PRE PRAKTICKE POUZITIE SA JAVI VYHODNEJSIE POUZIT POCITAC 130 XE Z DOVODU KAPACITY PAMETE , LEBO PRISTUP K DATAM JE DALEKO RYCHLEJSI AKO PRI

POUZITI DISKETOVEJ MECHANIKY. SAMOZREJME PO ROZSI-
RENI PAMETE JE AJ ATARI 800 XL MOZNE POUZIT, TO
VSAK ZAVISI NA KONKRETNÝCH POZIADAVKACH. AKO PERS-
PEKTIVNE SA JAVI POUZITIE EXTERNEHO RAMDISKU.

RIESENIE SOFTWARE MIDI PREPOJENIA

MIDI AKO BOLO UZ UVEDENE PRACUJE SO SERIOVYM
PRENOSOM DAT. TO ZNAMENA, ZE VSETKY DATA MUSIME
PRIJIMAT A VYSIELAT POCITACOM SERIOVE. ATARI RADY
XL A XE MA V SEBE ZABUDOVANY ELEGANTNY SERIOVY
PORT "POKEY", KTORÝ MA POMERNE VELKE MOZNOSTI A
DA SA PROGRAMOVAT. URCITE OBMEDZENIE NASTAVA PRI
SUCASNOM PRIJIMANI A VYSIELANI DAT. PRAKTICKY TO
ZNAMENA, ZE MOZEME LEN PRIJIMAT ALEBO VYSIELAT.
PRE POUZITIE TO ZNAMENA NAHRAT LEN JEDEN HUDOBNY
NASTROJ ALEBO VIAC HUDOBNÝCH NASTROJOV A POTOM
ICH REPRODUKOVAT PRIPADNE UPRAVOVAT. NIE JE TEDA
MOZNE V TOMTO JEDNODUCHOM POUZITI LEN "POKEY"
ROBIT TZV. POSTUPNY PLAYBACK. SAMOZREJME PRI
POUZITI DVOCH POCITACOV JE AJ TOTO MOZNE. TEDA SU
TU TROCHA PROBLEMY PRI NAHRAVANI. PRI PRODUKCII
SAMOZREJME STACI JEDEN POCITAC, KTORÝ JE SCHOPNY
OVLADAT AZ 16 HUDOBNÝCH NASTROJOV. UVEDENY
NEDOSTATOK BY SA DAL ODSTRANIT POUZITIM EXTERNE-
HO SERIOVEHO PORTU S TYM, ZE POTOM BY SA DATA
ZPRACOVAVALI V POCITACI CEZ PARALELNY PORT (PIA
) .

K TOMU ABY SME MOHLI VYTVORIT SEQUENCER REAL
TIME POTREBUJEME GENERATOR CASOVEHO INTERVALU.
TENTO SA DA POMERNE JEDNODUCHO REALIZOVAT POMOCOU
VERTIKALNEHO ZATEMNOVANIA (VBLANK), ALEBO
POMOCOU OBVODU "POKEY" . AK POUZIJEME VERTIKALNE
ZATEMNENIE ZISKAME GENERATOR Z PEVNOU FREKVENCIOU
50 HZ. PRI NAHRAVANI JE TENTO SPOSOB POSTACUJUCI.
PRI REPRODUKCII AK CHCEME MENIT TEMPO JE NUTNE
POUZIT PRE TAKTOVANIE OBVOD POKEY, CIM MOZEME VO
VELKOM ROZSAHU TEMPO MENIT.

DATA AK VYUZIJEME LEN OBVOD "POKEY" PRIJIMAME
TEDA TAK, ZE HO NAPROGRAMUJEME NA PRIJEM (31.25
KHZ) A CEZ VEKTOR PRERUSENIA PRE PRIJEM DAT PRI

SUCASNOM TESTOVANI CASOVEHO INTERVALU ZAPISUJEME DO PAMETE. SAMOZREJME, ZE PODPROGRAM NA KTORÝ SA PREJDE PO PRERUŠENÍ MUSÍ BYŤ V STROJOVOM JAZYKU A MUSÍ BYŤ NAFISANÝ POMERNE EFEKTÍVNE - MÁME NA TO MAXIMALNE 320 ÚS.

PRI VYIELANI NAOPAK PODLA DAT, KTORE PREDSTAVUJU TZV. TIMER INFORMACIE VYSIELAME TAK AKO SME ICH PRIJALI. OBVOD "POKEY" NAPROGRAMUJEME NA VYSIELANIE.

SIMULOVANIE DISKETOVEJ MECHANIKY NA ATARI 130 XE

VLASTNIKOV ATARI 130 XE BEZ DISKETOVEJ MECHANIKY URCITE ZACNE ZAUJIMAT PRIDAVNA PAMET, KTORA MA 64 KBYTE. TATO PAMET SA DA VYUZIT PRE ULOZENIE ROZNYCH DAT, STROJOVE PROGRAMY, OBRAZOVA PAMET A.T.D. PRIDAVNA PAMET JE SAMOZREJME PRISTUFNA AJ Z BASICU. AVSAK VYUZITIE PAMETE JE POMERNE TAZKOPADNE - MUSIME SI PRESNE EVIDOVAT KDE MÁME VSETKO ULOZENE. OBJAVILI SA PROGRAMY, KTORE SA NAZYVAJU POMERNE CASTO AKO "RAMDISKY". NAPRIKLAD PUBLIKOVANY - MOS V TURBO 4000. TENTO RAMDISK UMOZNUJE UKLADANIE PROGRAMOV DO TZV. ZARIADENIA DEFINOVANYM V TABULKE HATABS POD NAZVOM M: . TENTO RAMDISK KTORÝ MA Priblizne 22 KBYTE JE MOZNY VYUZIT AJ 800 XE (XL). SAMOZREJME TEN KTORÝ SA TROCHU VIAC ZAUJIMA O PROGRAMOVANIE URCITE ZISTIL, ZE PRACA S TAKYMTO RAMDISKOM JE DALEKO EFEKTIVNEJSIA AKO "CVICIT" S MAGNETOFONOVOU PASKOU "HORE-DOLE".

NEVYHODOU TOHTO RAMDISKU JE VSAK SKUTOCNOST, ZE TENTO RAMDISK MA LEN 22KBYTE A PRI "STUDENOM" STARTE (NAPRIKLAD POKE 580,1 + STLACENIE RESET) SA TENTO RAMDISK NENAVRATNE ZMAZE. TATO NEVYHODA JE ODSTRANENA PRI POUZITÍ RAMDISKU VYTVORENEHO Z PRIPAVNEJ PAMETE KTORA SA PRAKTICKY ZMAZE AZ PO VYPNUTÍ NAPETIA DO POCITACA.

POSTUPNYM STUDOVANIM DOS 2.5 KTORÝ MA SOFTWAREVOU PODPORU PRE VYUZITIE PRIDAVNEJ PAMETE AKO RAMDISKU POD NAZVOM D8: BOLA VYTVORENA VERZIA

KTORA SA DA NAHRAT Z MAGNETOFONU A VYUZIVA VSET-SLUZBY DOS 2.5 A DUP.SYS. SYSTEM BOL PREPRACOVANY NA POUZITIE TURBO 4000. TO ZNAMENA, ZE PROGRAMI SA DAJU NAHRAVAT ZVYSENOU RYCHLOSTOU. PRACA S TYMTO RAMDISKOM JE SKUTOCNE VYSOKO EFEKTIVNA. PRACUJEME SO SUBORMI PODOBNE AKO NA DISKETE. PRITOM S OPERACNOU PAMETOU MOZEME ROBIT CO CHCEME. AK SA NAM POCITAC "ZADRHNE" STACI POUZIT TLACITKO KTORE SI ZAUJEMCA MOZE PRIPOJIT NA KONKREKTOR CARTRIDGE (PUBLIKOVANE V PRAZSKOM ZPRAVODAJI). POTOM ZNOVA NAHRAT POPISOVANY PROGRAM Z PASKY KTORY, PRI DALSOM ZAVEDENI ZISTI CI JE NIECO V RAMDISKU - AK ANO PROGRAM SA ZASTAVI A VYPISE ADRESAR RAMDISKU. SUCASTOU PROGRAMU NA INICIALIZACIU RAMDISKU JE AJ UNIVERZALNY KOPIROVACI PROGRAM S KAPACITOU 30 KBYTE.

PRI VYUZITI NOVEHO OPERACNEHO SYSTEMU, KTORY SA NEDAVNO OBJAVIL POD NAZVOM *** Q-MEG *** JE POUZITIE TOHTO RAMDISKU SKUTOCNYM PREVRATOM: MOZEME PRACOVAT S KLASICKYMI DISKETOVYMI SUBORMI PRESNE AKO NA DISKETE. SU TO ZNAME PROGRAMY S VYKRICKNIKOM "!". DOKONCA JE MOZNE USKUTOCNIT STUDENY START S NAVOLENYM RAMDISKOM - VTEDY DOCHADZA K "SIMULOVANIU" DISKETOVEJ MECHANIKY. SAMOZREJME, ZE PROGRAMY KTORE CHCEME POUZIVAT MUSIA BYT UZ PREDTYM NAHRATE V RAMDISKU. K TOMU JE MOZNE VYUZIT KOPIROVACI PROGRAM, KTORY BOL SPOMINANY.

PRAKTICKY BOLO OVERENE POUZITIE ATMASU II, KDE PO VYVOLANI MONITORA A VOLBY SAVE: D:SUBOR SA PROGRAM NAHRAL DO RAMDISKU. SAMOZREJME VTEDY JE DISKETOVA MECHANIKA UPLNE ODSTAVENA. PODOBNE Z BASICU POTOM UZ PRACUJEME SAVE D:SUBOR LOAD D:SUBOR A NIE SAVE D8:SUBOR ATD.

BLIZSIE INFORMACIE PRIPADNE ZAUJEM O POPISOVANY PROGRAM POSKYTNE AUTOR:

NEDBALEC JOZEF
ATARI KLUB MARTIN

 TEST PRIDAVNEJ PAMETE ATARI 130 XE

TENTO PROGRAM UMOZNUJE PREVERIT STAV PRIDAVNEJ PAMETE. DOBA TESTOVANIA JE ASI 4 SEKUNDY. PRINCIP SPOCIVA V ZAPISANI URCITEHO CISLA A JEHO SPETNEHO PRECITANIA. V PRIPADE, ZE PRECITANE CISLO JE INE AKO SA ZAPISALO PROGRAM OHLASI CHYBU A BANKU V KTOREJ SA CHYBA VYSKYTLA. V PRIPADE, ZE POCET CHYB JE NULOVI MOZNO POVAZOVAT PAMET ZA BEZCHYBNU. K TESTOVANIU SA ODPORUCA POUZIT CISLO 255 A 0 (DECIMALNE). V TYCHTO DVOCH PRIPADOCH OTESTUJEME VSETKYCH 8 PAMETI RAM 4164, KTORE TVORIA 64 KBYTE.

PROGRAM UMOZNUJE V PRIPADE CHYBY SPECIFIKOVAT, KTORA Z PAMETI JE VADNA (PRAKTICKY OVERENE): POSTUPNE TESTUJEME JEDNOTLIVE BITY V BYTOCH. TO ZNAMENA, ZE POSTUPNE ZAPISUJEME CISLA 1,2,4,8, 16,32,64,128 (DECIMALNE).

V PRIPADE ZISTENIA ZAVADY V PAMETI ODPORUCAM ZASLAT POCITAC NA ADRESU KOVOSLUZBY PRAHA, DOSTA-LOVA 10 (TEL.C. 353082).

```

10 FOR X=1536 TO 1577: READ DAT: POKE X,DAT:NEXT X
15 PRINT " TEST PRIDAVNEJ PAMETE 130 XE"
20 PRINT: PRINT "TESTOVANE CISLO",: INPUT CISLO
25 PRINT: PRINT "PAMET","CHYBA"
30 POKE 559,0: SUC=0
35 FOR BANKA=0 TO 3
40 POKE 54017,193+4*BANKA
45 POKE 203,141: POKE 204,0: POKE 205,64:
   POKE 206,96: U=USR(1536,CISLO)
50 POKE 203,173: POKE 204,0: POKE 205,64:
   POKE 207,0: POKE 1560,CISLO: U=USR(1555)
55 PRINT BANKA, PEEK(207): NEXT BANKA
60 POKE 559,34: PRINT "-----":
   GOTO 20
70 DATA 104,104,104,32,203,0,230,204,208,2,
   230,205,162,128,228,205,208,241,96
80 DATA 104,32,203,0,201,255,240,2,230,207,
   230,204,208,2,230,205,162,128,228,205,208,
   235,96

```


VYPIS STROJOVYCH PODPROGRAMOV:

ZAPIS:

```
          PLA
          PLA
          PLA
B        JSR ZAPIS; ADRESA 203 (DECIMAL)
          INC 204
          BNE A
          INC 205
A        LDX #128
          CPX 205
          BNE B
          RTS
ZAPIS    STA 16384; RUTINA PRE ZAPIS
          RTS
```

CITANIE

```
          PLA
E        JSR CITANIE; ADRESA 203 (DECIMAL)
          CMP #CISLO; TESTOVANE CISLO
          BEQ C
          INC 207; REGISTER POCTU CHYB
C        INC 204
          BNE D
          INC 205
D        LDX #128
          CPX 205
          BNE E
          RTS
CITANIE  LDA 16384
          RTS
```

RUTINY PRE ZAPIS/CITANIE MAJU FIXNU ADRESU 203.

PROGRAM JE MOZNE SKRATIT PREPISANIM STROJOVYCH PODPROGRAMOV DO RETAZCOV. V RIADKU 45 POTOM ZAMENIT: U=USR(ADR("RETAZEC"), CISLO)) A RIADOK 50 U=USR(ADR(RETAZEC1)). RETAZEC JE TVORENY DATAMI V RIADKU 70. RETAZEC1 JE TVORENY DATAMI V RIADKU 80

NEDBALEC JOZEF
ATARI KLUB MARTIN

JACKET LISTER FOR ATARI XL/XE

Obslužný program "Jacket Lister" tlačí abecedný adresár programov na disku priamo na obal disku. Originálna verzia programu bola vytvorená pre počítač Commodore 64 a 128, nové verzie sú určené pre IBM PC, Apple II. Program vyžaduje tlačiareň a minimálne 32 kB pamäti.

Klasický spôsob hľadania programu na disku - vloženie disku a vylistovanie zoznamu na obrazovku - je pri hľadaní špecifického programu pomalý a únavný. Jacket Lister je časovo veľmi úsporné riešenie. Neumožňuje len tvorenie diskových zoznamov na obyčajný papier, ale tiež listing adresára priamo na obal. Takto môžete okamžite vidieť zoznam programov na disku. Program zahrňuje aj tlač dátumu, takže vidieť, či sa nejedná o zastaralý záznam. Jacket Lister môže zahrňovať až 88 mien úsekov použitím čela a rubu obalu. Niektoré počítače umožňujú ukladanie viac úsekov na disk, obal však poskytuje miesto len pre 88 údajov/.

Meno pre disk je uložené v premennej NAME\$ /riadok 15/ a môže byť ľubovoľne menená. Dĺžka reťazca je ohraničená 26 znakmi.

Jacket Lister je program komentovaný, takže nie je potrebný žiaden manuál. Po spustení programu vlož disk, ktorý chceš katalogizovať a sleduj informácie k vytváraniu obalu na obrazovke. Po vytlačení obalu tento postrihaj na potrebný formát, zlož pozdĺž natlačenej skladačej hrany a zlep.

Atari verzia programu pracuje na každom počítači ATARI 400, 800, 130 XL, XR s minimálne 32 kB pamäti. Môže pracovať s ľubovoľnou, štandardne priradenou tlačiarňou.

Podľa G. Jackmond: Jacket Lister, COMPUTE! September 1986
upravil Ing. Ján Čelko, Žilina

JACKET LISTER PRE ATARI XL/XE

```

2 REM *           JACKET LISTER           *
3 REM *           PRE ATARI UPRAVIL       *
4 REM *           ING FUCHSBERGER         *
5 REM *           ZILINA                   *
7 REM
10 DIM DATE$(12),UPPER$(1),K$(1),DN$(10),DIR$(17*88),T$(20),NAME$(26),SPC$(80),N
AZOV$(22),NR$(3)
15 FOR A=1 TO 80:SPC$(A,A)=" ":NEXT A:NAME$=" REFERENCNY DISK ":REM THIS M
UST BE 26 CHARACTERS
20 OPEN #4,4,0,"K:"
100 GRAPHICS 0:POKE 710,15:POKE 709,0:POKE 712,55
102 POSITION 4,6:PRINT "NAPIS NAZOV DISKU MAX.22 ZNAKOV"
105 POSITION 15,13:INPUT NAZOV$
106 PRINT CHR$(125)
107 POSITION 4,6:PRINT " NAPIS CISLO DISKU - TRI ZNAKY "
108 POSITION 15,13:INPUT NR$
109 PRINT CHR$(125)
110 POSITION 10,6:PRINT " AKY JE DNES DATUM "
120 POSITION 6,8:PRINT " VSTUP DD/MM/RR POTOM <RETURN> "
130 POSITION 15,13:INPUT DATE$:PRINT "(CLEAR)":POKE 752,1
160 PRINT CHR$(125):POSITION 2,5:PRINT "Z KTOREJ DISKETOVEJ JEDNOTKY CITAT ?":PO
SITION 17,7:PRINT "(1-9)"
170 GOSUB 1000:IF K$<"1" OR K$>"9" THEN 170
180 DN$="D1:*,*":DN$(2,2)=K$
190 PRINT CHR$(125):POSITION 3,8:PRINT " CITAM DATA : PROSIM CAKAJ ":POKE 712,
136
200 TRAP 220:FILE=1:OPEN #1,6,0,DN$
210 INPUT #1,T$:DIR$((FILE-1)*17+1,FILE*17)=T$:FILE=FILE+1:IF FILE<90 THEN 210
220 FILE=FILE-2:TRAP 65000:CLOSE #1:IF PEEK(195)=136 OR FILE=88 THEN 250
230 PRINT CHR$(125):POSITION 12,11:PRINT "DISK ERROR #":PEEK(195)
240 POSITION 8,13:PRINT "PRESS ANY KEY TO RETRY":GOSUB 1000:GOTO 190
250 PRINT CHR$(125):POSITION 5,12:PRINT " TRIEDIM ":POKE 712,200
260 G=INT(FILE/2)
265 N=0:FOR I=1 TO FILE-G
270 IF DIR$((I-1)*17+1,I*17)<=DIR$((I+G-1)*17+1,(I+G)*17) THEN 290
280 T$=DIR$((I-1)*17+1,I*17):DIR$((I-1)*17+1,I*17)=DIR$((I+G-1)*17+1,(I+G)*17):D
IR$((I+G-1)*17+1,(I+G)*17)=T$:N=N+1
290 NEXT I:IF N=1 THEN 265
300 G=INT(G/2):IF G=1 THEN 265
310 PRINT CHR$(125):POSITION 2,8:PRINT " TLACIM OBALKU : PROSIM CAKAJ ":POKE 7
12,104:OPEN #1,4,4,"E:"
320 FOR A=1 TO 3:PRINT #1;CHR$(13):NEXT A
330 GOSUB 3070:PRINT #1;"REZ"
340 GOSUB 3000:GOSUB 3010:GOSUB 3020
350 GOSUB 3000:PRINT #1;CHR$(27);CHR$(14);NAME$;NR$;CHR$(27);CHR$(15);:GOSUB 30
20
360 FOR A=1 TO 2:GOSUB 3000:GOSUB 3010:GOSUB 3020:NEXT A
370 GOSUB 3000:PRINT #1;SPC$(1,4);NAZOV$;SPC$(1,10);DATE$;SPC$(1,38)-LEN(DATE$)-L
EN(NAZOV$);:GOSUB 3020
375 GOSUB 3000:GOSUB 3010:GOSUB 3020
380 CD=INT(FILE/2):IF FILE>32 THEN 400
390 FOR DD=1 TO CD:GOSUB 3000
400 PRINT #1;SPC$(1,8);DIR$((DD-1)*17+1,DD*17-3);SPC$(1,9);DIR$((CD+DD-1)*17+1,(
CD+DD)*17-3);SPC$(1,7);

```

```

410 GOSUB 3020:NEXT DD
420 GOSUB 3000:GOSUB 3010:GOSUB 3020:DD=DD+1
430 IF DD>17 THEN 450
440 GOTO 420
450 GOSUB 3070:PRINT #1,"OHYB"
460 FOR SL=1 TO 29:GOSUB 3050:GOSUB 3060:GOSUB 3065
470 NEXT SL:GOSUB 3090:PRINT #1,"REZ":GOTO 2000
480 FOR DD=1 TO 16:GOSUB 3000
490 PRINT #1;SPC$(1,8);DIR$(DD-1)*17+1,DD*17-3);SPC$(1,9);DIR$(16+DD-1)*17+1,(
16+DD)*17-3);SPC$(1,7);
500 GOSUB 3020:NEXT DD
510 GOSUB 3000:GOSUB 3010:GOSUB 3020
520 GOSUB 3070:PRINT #1,"OHYB":GOSUB 3050:GOSUB 3060:GOSUB 3065
530 CX=INT((FILE-33)/2):CZ=CX+32
540 FOR DD=33 TO CZ:GOSUB 3050
550 PRINT #1;SPC$(1,9);DIR$(DD-1)*17+1,DD*17-3);SPC$(1,6);DIR$(DD+CX-1)*17+1,(
DD+CX)*17-3);SPC$(1,11);"!"
560 NEXT DD
570 GOSUB 3050:GOSUB 3060:GOSUB 3065:DD=DD+1
580 IF DD<61 THEN 570
590 GOSUB 3090:PRINT #1,"REZ":GOTO 2000
1000 POKE 752,1:GET #4,K:K#=CHR$(K):RETURN
2000 FOR CR=1 TO 3:PRINT #1;CHR$(13):NEXT CR:CLOSE #1
2010 PRINT CHR$(125):POSITION 3,10:PRINT "ABECEDNE DISK VYTRIEDENY "
2020 POSITION 3,13:PRINT "PRAJETE SI DALSI DISK TRIEDIT ?"
2030 GOSUB 1000:IF K#="Y" THEN RUN
2040 IF K#<>"N" THEN 2030
2050 PRINT "(CLEAR) (2 DOWN)BYE":POKE 752,0:END
3000 PRINT #1;" !";!*";RETURN
3010 PRINT #1;SPC$(1,52);RETURN
3020 PRINT #1;"*!";!*";RETURN
3050 PRINT #1;"!";!*";RETURN
3060 PRINT #1;SPC$(1,54);RETURN
3065 PRINT #1;"!";RETURN
3070 PRINT #1;" ";FOR A=1 TO 72:PRINT #1;"-";NEXT A:RETURN
3090 PRINT #1;" ";FOR A=1 TO 54:PRINT #1;"-";NEXT A:RETURN

```

VZOR JACKET LISTERU

REFERENCHY	DISK	015
SYSTEMOVE PROGRAMY	19/09/1988	
* ADRESAR BAS	* GRAFY2D BAS	
* ATARI1 BAS	* KURZIVA BAS	
* AUTO3D BAS	* NEMCINA1BAS	
* AUTOPROFBAS	* OXYGENE1BAS	
* BASIC BAS	* OXYGENE2BAS	
* BLC02 UTL	* SUKOLIE BAS	
* CHANGER BAS	* TELESR BAS	
* CORNERS BAS	* VARLIST BAS	
* CORNERS2BAS	* VIDEO1 BAS	
* DOS SYS	* VIDEO2 BAS	
* DUP SYS	* VYPTELESBAS	
* GRAFY1 BAS	* ZEMEPIS BAS	

REZ

OHYB

REZ

RAMdisk 256 kB pre rozšírenie pamäte na 320 kB.

O RAMdiskoch bolo už dosť popísané, ale ani jedno prevedenie mi nevyhovovalo. Rôzne informácie zrejme ani neboli odskúšané, ale len uverejnené. Rozšírenie s obvodom FREDDIE vyžaduje zvláštny prístup pri realizácii pamäte. Vychádzal som zo zapojenia uverejneného v /A/ , ale s obvodom FREDDIE to jednoducho nechodilo, tak ako to bolo uverejnené. Ďalej som bral do úvahy zapojenie ATARI 130, kde je použitý FREDDIE a obvod PAL. Žiaľ žiadne bližšie informácie o týchto obvodoch mi nie sú známe. Zapojenie podľa /A/ síce je rozšírenie na 320 kB, ale neviem, ktoré programové vybavenie to vie použiť vyššie ako + 64 kB. Mať pamäť len pre použitie vo vlastných programoch, asi nemá význam.

Firma Compy Shop vyrobila DOS / Bibo - DOS / , ktorý využíva RAMdisk 256 kB. Sú aj ďalšie programy, ktoré toto rozšírenie používajú, ako FCOPY, SCOPY. Toto rozšírenie pozná aj operačný systém QMEG - OS. Zvláštnosťou rozšírenia od Compy Shop je to, že na prepínanie stránok pamäti nepoužíva port PB 5, ale port PB 7, ktorý normálne zapína SELF TEST. Tu sa testuje PB 5 a PB 7, či sa jedná o RAMdisk, alebo o SELF TEST. Keď sa jedná o RAMdisk, tak je pre MMU vstup MAP blokované.

V zapojení podľa /A/ je o jeden obvod menej, než v danej schéme. Dôvod je ten, že som musel použiť dva rôzne selecty, a nie je multiplexer s dva na jedna v jednom púzdre s dvomi select vstupmi. Preto bol pridaný ďalší obvod 742 S 157. Odpor RX - 1K som zaradil z dôvodu, že po zahriatí počítača vysadzoval RAMdisk. Takto sa choval môj počítač /ATARI 800 XL/ i po-dľa zapojenia v /A/. Navyše som musel blokovať ešte kondenzátorom M 1 vývod 40 u

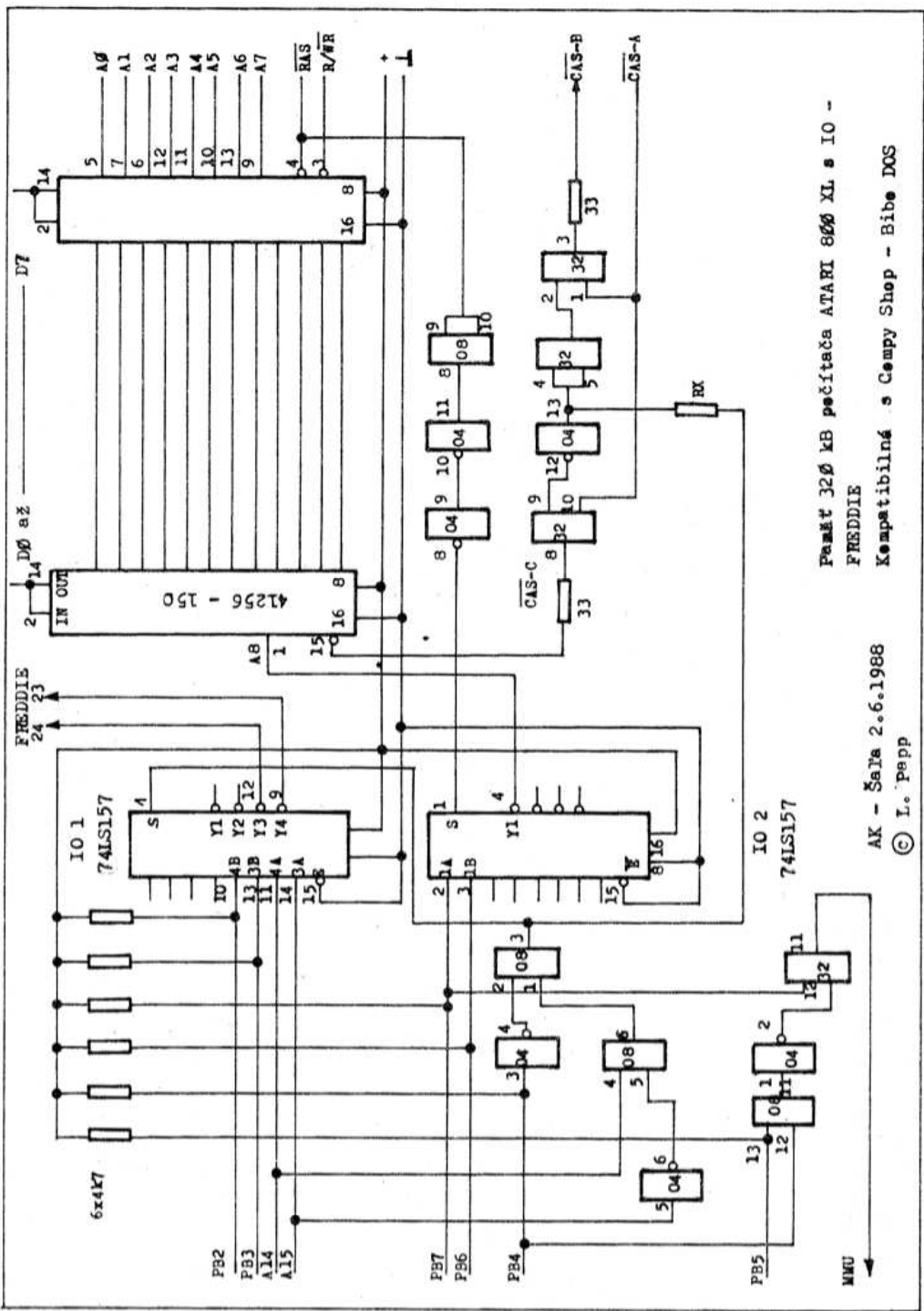
FREDDIE. Zrejme sa jedná o čiastočnú vadu môjho obvodu FREDDIE. Do vydania tejto publikácie nebolo času odskúšať toto zapojenie v ďalších počítačoch. Zrejme odpor RX vo väčšine prípadoch nebude potrebný. O ďalších skúsenostiach sa zmienim v niektorom spravodaji ATARI Klubu Tlmače.

Použitá literatúra

/A/ - ATARI Magazín 2,3 / 87

Ladislav PAPP

ATARI Klub ŠALA

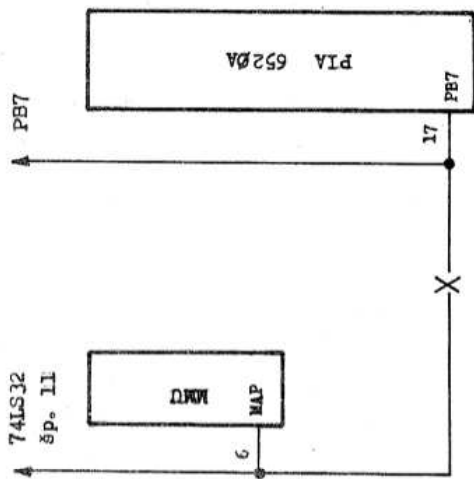
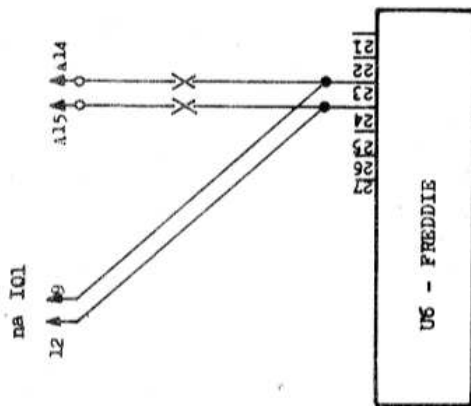


FREDDIE 24 23

Pamät 320 kB počítača ATARI 800 XL s IO -
 FREDDIE
 Kompetibilná s Compy Shop - Bibe DOS

AK - Šafa 2.6.1988
 © L. Pepp

Úprava na základovej doske



X - prerušenie plošného spoja

OPERAČNÝ SYSTÉM " MOS "

V oblasti vonkajších pamätí domácich počítačov Atari 800 XL/XE dominujú kazetopáskové pamäte s dostupnými magnetofónmi. Disketová mechanika je ešte stále pre väčšinu majiteľov počítačov nedostupná. Aj keď magnetofón nikdy nemôže nahradiť prácu disketovej mechaniky, môže sa k nej svojimi vlastnosťami aspoň čiastočne priblížiť. Na základe týchto úvah vznikol operačný systém TURBO MOS.

Program TURBO MOS je rozšírením pôvodného operačného systému, ktorý pri ovládaní datového magnetofónu nevyniká rýchlosťou a ani spoľahlivosťou záznamu. MOS umožňuje podstatné zrýchlenia zefektívnenia programátorských prác s cieľom čo najviac potlačiť nedostatky datového magnetofónu.

V systéme turbo mos nie je potrebné programy prerábať, stačí nahrať program v štandardnej rýchlosti 600 BD kopírovacím programom a späť zaznamenať v zrýchlenom zázname 4 000 BD a prípadne i viac. Systém sa skladá z troch častí :

- RAMDISK
- TURBO DNM
- Monitor

Vo všetkých nahratých programoch možno využívať ramdisk a zrýchlený záznam pre prácu s dátami. Výhodou je vysoká spoľahlivosť záznamu, prekonávajúca všetky známe spôsoby záznamu na ATARI. Pod MOS -om by mali pracovať všetky užívateľské kazetové a disketové programy, ktoré nemajú softwareovú ochranu a ktoré priamo nespolupracujú s disketovou jednotkou.

RAMDISK

RAMDISK využíva pamäť RAM schovanú pod ROM, možno ho výhodne využívať ako rýchle zariadenie pre ukladanie pracovných súborov, najčastejšie spúšťaných programov a pod.

V Basicu RAMDISK poskytuje možnosť segmentovať programy. Toto môžeme využiť napríklad pri riešení rozsiahleho problému, keď nestačí operačná pamäť, program sa rozdelí na viac častí a tieto sa môžu postupne podľa potreby spúšťať. Tiež sa tu otvára možnosť v programe dynamicky meniť s niektorými obmedzeniami, samotný program príkazom ENTER.

RAMDISK pozná súbory vstupné, výstupné a rozšírované, poskytuje možnosť vypísať adresár RAMDISKu, ktorého kapacita je až osem súborov.

TURBO DNM

Turbo DNM je rozšírením už existujúceho operačného systému Turbo D. Navyiac má možnosti nastavovať rýchlosť záznamu od 4 000 BD vyššie. Pri čítaní sám zistí akou rýchlosťou bol záznam zapísaný. Toto prispieva k zvýšeniu spoľahlivosti a prenositeľnosti údajov. Ďalej si užívateľ môže jednoduchým spôsobom určiť či chce používať záznam jednoduchými blokmi voľbou zariadenia "D" alebo "N". Rutina OPEN je rozšírená o výpi názvu súboru štandardným príkazom OPEN X, 6, 0, "D:" a INPUT X, A použiteľnými v každom programe.

MONITOR

Monitor slúži užívateľovi ako účinný nástroj pri organizovaní práce s mikropočítačom. Poskytuje najpotrebnejšie služby: -

- zavádzanie a spúšťanie strojových programov, zavádzač sám rozlíši kazetový formát od disketového a zavedie ho.
- prechod zo strojového programu do Basicu bez vypínania počítača, každé vypnutie a zapnutie počítača škodí. MOS šetrí čas užívateľa a predlžuje životnosť mikropočítača.
- presun súboru z pásky do RAMdisku a naopak, tieto služby možno

tiež využiť na kopírovanie súborov do rozsahu 22 KB.

- výpis adresára RAMdisku a názov súborov na páske.
- ovládanie kurzoru pomocou joysticku, výhodné pri prezeraní textov v textových editoroch.
- nastavenie 600/800 BD pre zariadenie "C", nastavenie rýchlosti záznamu 4000 BD a viac pre TURBO DNM.

Služby monitora sa môžu využívať aj v zavedených strojových programoch. Monitor možno kedykoľvek zavolať a vrátiť sa späť do programu stlačením RESET.

Pod systémom TURBO MOS pracujú užívateľské programy, pre hry je príliš rozsiahly. Mos simuluje do určitej miery disketovú mechaniku a preto pod ním pracuje veľká časť programov, výlučne disketových. MOS obsahuje časť disketových operácií, pretože všetky operácie na magnetofóne nie je možné vytvoriť, práca s magnetofónom bude náročnejšia na obsluhovanie ako s disketovou jednotkou. No aj tak MOS bude užitočný m pomocníkom pre majiteľov mikropočítačov ATARI, ktorí nevlastnia disketovú jednotku, lebo sa im sprístupnia ďalšie programy, disketové programy ako disketová verzia KYAN PASCAL, DESIGN MASTER, ATARI WRITER, PAPER CLIP, STARTEXTER a mnohé iné.

TURBO MOS jestvuje v dvoch verziách - jedna pracuje s hardware TURBO D a druhá s hardware TURBO 2 000. Podrobné informácie o popísanom systéme sú v užívateľskej príručke MOS.

Popis a práce s HARDCOPY

V nasledujúcej časti si popíšeme niekoľko HARDCOPY programov s ktorými je pomerne jednoduchá práca, nevyžadujú žiadny zvláštny manuál. Stačí len trochu trpezlivosti a možno objavíte aj niektoré ďalšie možnosti uvedených programov.

Nasledujúce popisy nie sú prekladom firemných manuálov ale len skúsenosťami, ktoré som získal pri práci s týmito programami.

1/ HARDCOPY PRINT 1029 V.2.0.

Program spracováva obrázky vytvorené programami:

1. HI RES ANTIC MODE /8/
2. KOALA MICROILLUSTRÁTOR
3. PRINT SHOW /MACIC SHOW/
4. PAGE DESIGNER
5. B/GRAPH

a tlačí na tlačiarni ATARI 1029 a s touto kompatibilnými.

Program je spracovaný v BASICu. Po nahratí do počítača a odštartovaní nám ponúkne menu. Po voľbe sa objaví krátka informácia. Po stlačení (RETURN) vytlačí directory programov zvoleného módu. Obraz, ktorý chceme vytlačiť sa volí dvoma číslami /podľa obrazovky/. Po zvolení je položená otázka o veľkosti tlače

1 - 80 x 136 mm

2 - 165 x 258 mm

Po voľbe rozmeru tlače sme upozornení na funkcie

(START) - tlač obrazu

(SELECT) - voľba inverzie

(OPTION) - návrat do menu

(RETURN) - nahrať zvoleného programu

Program je veľmi dobrý so širokou škálou pôsobnosti.

2/ HARDCOPY PRE ATARI TLAČIARENĚ

Program je v basicu a uložený pod file "HARDCOP 2.BAS".

Po odštartovaní nás upozorní, že obraz musí byť uložený pod m.PIC - file a musí mať 62 sectorové file.

Ďalej pýta obraz, ktorý chceme nahráť. Po stlačení (RETURN) vytlačí na obrazovke všetky file s príveskou PEC. Po zadaní zvoleného file je nám ponúknutá tlač

1. celej obrazovky
2. výrezu

Tlač pomocou tohto programu je veľmi pomalá a vytlačenie obratovky trvá cca 7 min.

3/ HARDCOPY PRE GR.9

Program je taktiež spracovaný v basiou a uložený pod file /HARDCOP 9.BAS/. Po odštartovaní si pýta názov obrazu. Ak nevieme pomocou (RETURN) vyvoláme directory. Názov programu /obrazu/ zadávame

napr. D: DONALD.PIC

Po nahratí obrazu stlačíme (D) a príde otázka tlačiť inverzne /J/N/, upozorní nás na prípravu tlačiarne a pýta (RETURN).

4/ PICCON V 2.0

Je program v strojovom kóde uložený na disku pod file PICCONV.COM.

Po nahratí programu je nám poskytnuté menu:

- 1 - LOAD MICROPAINTER
- 2 - LOAD KOALAPAD
- 3 - SAVE MICROPAINTER
- 4 - SAVE KOALAPAD
- P - PICTURE
- X - X MIRROR /otáčanie obrazu okolo x-ovej osi/
- Y - Y MIRROR / " " " y-ovej " /
- I - INVERSE
- H - HARDCOPY (1029)
- D - DIRECTORY
- L - DELETE
- R - RENAME
- + - LOCK
- - UNLOCK
- F - FORMAT (DRIVE 1)
- W - WRITE DCS.SYS
- S - SAVE SCREENLOADER

Voľbou (D) vyčítame file, ktoré máme na disku. Po zistení fi-

le volíme druh /MICROPAINTER - 1, KOALA - 2/ názov obrazu, ktorý mienime vytlačiť. Obraz sa nahrá a objaví na obrazovke, klávesami (X) a (Y) môžeme obraz obracať okolo osi X a Y, a tým vytvárať rôzne variácie. Klávesou (I) volíme inverziu obrazu. Klávesou (H) - - HARDCOPY odštartujeme tlač. Tlačítkom ESCAPE sa z obrazu vrátíme do menu a (P) zas do obrazu.

Program má v sebe zabudovaný mini DOS.

5/ PRINT SHOP MAGE CONVERTER

Kedže program print shop nie je kompatibilný s Atari tlačiarňami, uvedený program umožňuje tlačiť časť "SCREEN MAGIC", a to tým spôsobom, že si vytvoríme požadovaný obraz a tento uložíme na náš data disk, z ktorého potom čítame údaje do hardcopy programu a vytlačíme.

6/ PRINT 1029

Vytvára hardcopy z obrazov vytvorených:

- HI - RES GRAPHICS 8
- KOALA MICRO - ILUSTRÁTOR

na tlačiarne ATARI 1029.

Program je v BASICU a uložený pod file "PRNT 1029.BAS". Po nahratí sa objaví menu. Po zadaní názvu obrazu sa nás pýta v akom móde je obraz vytvorený.

1. HI-RES ANTIC MODE /8/
2. KOALA MICRO - ILUSTRÁTOR

Ďalej si program pýta informácie pre tlač

- 1 - uložiť obraz vľavo
- 2 - v strede
- 3 - v pravo

Po nahratí obrazu cez (RETURN) volíme

- (START) - tlač obrázku
- (SELECT) - voľba inverzie
- (OPTION) - návrat do menu

7/ SCREEN DUMP II

Je program určený pre zobrazovanie a tlač obrázkov vytvorených koalou a micropainterom.

Po zavedení programu dostaneme menu:

- K) KOALA/MALTAFEL
- M) MICROPainter
- D) BILO DRUCKEN
- P) PARAMETER
- ESC) BILE ZEIGEN

Najskôr volíme (P) pre nastavenie parametrov našej tlačiarne /ak táto už nie je nahratá na programovom disku/. Zvolené parametre uložíme cez (S) na disk, ktoré sú potom automaticky pri nahrávaní programu natiiahnuté. Ďalej volím režim v ktorom je obraz nahratý /KOALA, MICROPainter/ klávesou (K) alebo (M). Na obrazovku sú načítané file obrazov, z ktorých volíme kurzorovými tlačítkami a potvrdíme (RETURN). Po nahratí prejdeme nazad do menu (ESC) a cez (D) vyvoláme tlač. Na obrazovke dostaneme ponuku

NORMAL	INVERS
MITTE	LINKS

Tlačítkom (I) volíme normálnu alebo inverznú tlač, tlačítkom (A) volíme polohu obrazu na papieri.

Číslom 1-5 (voľba veľkosti obrazu/ odštartujeme tlač.

8/ DESIGN MASTER + HARDCOPY

Uvedený program pozostáva z dvoch častí. V prvej - DESIGN MASTER - tvoríme /obraz, schému, popis,/ ktorú uložíme na data disk, alebo prechodom do druhej časti - HARDCOPY - vytlačíme /vytvorené dielo ostáva v pamäti počítača/.

Druhá časť programu - HARDCOPY - sa volí z hlavného menu stlačením klávesy (SELECT) a používa sa pre tlač na mozaikových tlačiarnach kompatibilných s ATARI. Dáta pre prispôsobenie tlačiarne je možné nahrat' na disk s EXT.SET. obraz je možné natiahnúť priamo z -HARDCOPY- /nemusíme nahrávať design master/. Je možné nahrat' obraz 1 a 2, tieto pri tlači spojiť a tak vytvorit' plochu 320x384 bodov. Program je komentovaný - vedie nás, tak popíšem len najdôležitejšie povelý.

BILD ANSEHEN - zobrazí na obrazovke obraz 1 alebo 2. Zo zobrazených obrazov potom volíme ktorý chceme tlačiť. Do menu sa môžeme vrátiť cez (RETURN), (ESC) alebo tlačítkom joysticku.

BILD LADEN - natiahne do programu zvolený obraz /schemu/ z diskety. Pri nahrávaní volíme číslo disketovej jednotky, názov súboru a číslo obrazu /1 alebo 2/ pod ktorým bude uložený. Ak budeme neskôr robiť tlač obrazov musíme si ktorý nahráme ako 1 a ktorý 2, nakoľko najskôr je tlačený obraz 1.

DISK INHALT - zobrazí directory na disku.

PROGRAM BEENDEN - vracia program do hlavného menu, maže pamäť a sme na naše rozhodnutie opýtaný cez otázku áno-nie.

BILD AUSDRUCKEN - obraz vytlačiť. V tejto časti volíme číslo tlačeného obrazu 1, 2 alebo obidva /beide/. Normálnu alebo inverznú tlač. Pri normálnej tlači je biely bod z obrazovky na papieri ako čierny.

Pre tlač je veľmi dôležitá časť DRUCKER ANPASSUNG nakoľko v tejto časti pripravíte svoju tlačiareň pre prácu.

Ak máte kompatibilnú tlačiareň s EPSON, SEIKOSNA, ATARI 1029 môžete použiť prislúchajúci SETUP, ktorý je nahratý na programovom disku. K ostatným tlačiarnam si musíte vytvoriť svoj vlastný kód, ktorý potom zapíšete na disk pre ďalšie použitie. Súbor je dlhý 1 sektor a s EXT.SET.

V tejto časti sú nasledovné povely:

SETUP LADEN - natiahne SETUP súbor z diskety pre Vašu tlačiareň. Zadáme číslo disket. jednotky a názov súboru s EXT.SET. Na disku sú tieto súbory:

PRE ATARI - AT 1029 KL - malý formát
 AT 1029 GR - veľký formát
 PRE EPSON - FX 80 KL - malý formát
 FX 80 GR - veľký formát

Ak chceme automaticky naťahovať niektorý SETUP premenujeme ho na HC.SET.

SETUP SPEICHERN - ukladá nami vytvorený setup na disk.

HOEHE - BREITE - touto voľbou volíme rozmer tlačeného obrazu. Výška môže byť 1 až 4 násobná šírka môže byť 1 až 3 násobná. Šírka ŠPECIAL sa používa pri tlačiarni Atari 1029 k rozšíreniu riadku z 320 na 480 bodov. Ak nechceme obraz deformovať musíme pre šírku a výšku voliť rovnaký stupeň zväčšenia. Pozor - pri zväčšovaní nesmie obraz "pretiecť" cez obrazovku, lebo tlačiareň v takom prípade nevie tlačiť. Pri zväčšení 2x a tlačí na šírku A⁴ môžeme kresliť na 240 bodov.

Pri tvorbe vlastného setup použijeme nasledovné povely:

MATRIX - tu zadáme počet ihličiek tlačiarni, ďalej bitovú váhu hornej ihličky pre

ATARI 1029	64
SEIKOSHIA	1
EPSON	1

potom označíme či je najvyšší BIT /bit 128/ obsadený. Pre ATARI 1029 nie je obsadený a volíme NEIN.

CODE 1 INIT - tlačiarni je potrebné pred zahájením graf. tlače vyslať riadiaci kód, aby bola pre graf. tlač pripravená.

CODE 2 BIT-MAP - slúži k prepnutiu tlačiarni do gr. modu
Např.: EPSON a jednoduchá šírka je kód 27, 75, 64,
1. Udáva počet bodov tlač. riadku a počet bodov na riadok.

CODE 1 VORSCHUB - za BIT-MAP kodom na konci riadku musí nasledovať kód pre posun 10 "CR" alebo 155 "EOF".

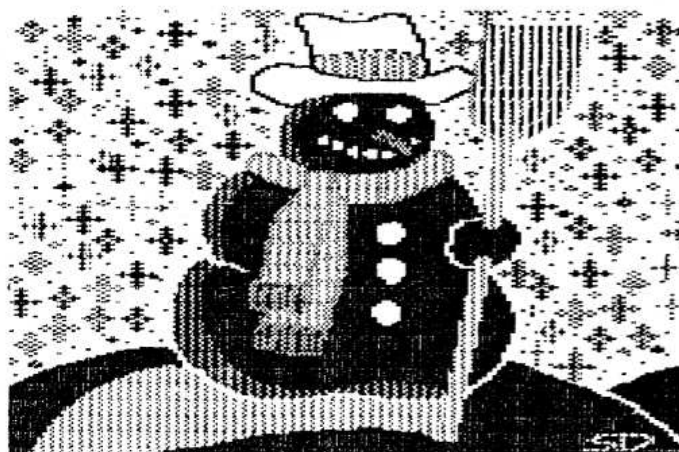
CODE 4 ENDE - týmto kódom sa tlačiareň vracia do normálneho modu. Tento kód je vyslaný raz na konci tlače.

Po overení takto vytvoreného kódu tento uložíme na disk pre ďalšie použitie.

UKAZKA TLACE SCREEN DUMPLU



INVERZNA TLAC



Doplnky pre ATARI SPEEDCALC

Článok popisuje dve rozšírenia Atari verzie Speedcalcu, populárneho tabuľkového programu, publikovaného v marci 1986. Tieto nové vlastnosti umožnia zvýšenú kontrolu nad výstupnou tlačou a dovoľia kopírovanie alebo presun pamäťových blokov bez prepočítania celej tabuľky. Doplnky požadujú disketovú jednotku.

Atari Speedcalc je výborný tabuľkový program, ale aj dobrý program môže byť zdokonalovaný. Doplnky vytvárajú niekoľko modifikácií ku zvýšeniu výkonu a vymoženosti programu. Doplnkový program po spustení vyzve k vloženiu disku obsahujúceho Speedcalc. Pozor, presvedč sa, že máš náhradnú kópiu Speedcalcu na ďalšom disku pre prípadný výskyt chyby v rozšírenej verzii. Po vložení disku stlač RETURN. Rozširujúci program automaticky pripíše na koniec súboru potrebné kódy pre AUTORUN.SYS úsek Speedcalcu. Po krátkom čase počítač vytlačí DONE. Pri zavedení Speedcalcu deaktivuj BASIC a zaveď systém.

Výberové kopírovanie.

Pri tlači na rôzne zariadenie /tlačiareň, disk, obrazovku/ originálny Speedcalc vždy začína tlačiť v ľavej hornej bunke /AA1/. Táto vlastnosť limituje šírku každej tlače 7- alebo 8- bunkových stĺpcov na 80 stĺpcovej tlačiarňi. Rozšírená verzia má schopnosť poslať obsah ktoréhokoľvek bloku buniek na zariadenie podľa výberu. Pre tlač vybraných blokov buniek presuň kurzor na spodnú pravú bunku bloku, ktorý chceš tlačiť a stlač CTRL-P. Keď udávaš výstupné zariadenie, vstup P: vyberá tlačiareň, E: obrazovku, D: tlačí úsek na disk. Potom presuň kurzor na ľavý vrchol bloku, ktorý chceš tlačiť /ľavú hornú bunku/ a stlač RETURN. Speedcalc vytlačí vybraný blok.

Zdokonalený presun a kopírovanie.

Nová verzia Speedcalcu má tiež schopnosť kopírovať alebo presúvať bloky buniek bez prepočítavania. Toto dovoľuje vytvárať spoločné sekcie tabuľky pre tlač bez chýb spôsobených výpočtom. /Napríklad môžeš presunúť stĺpec ľavých buniek na tlač/. Prepočítavanie behom kopírovania a presun operácií je spojené s automatickým prepočtovým módom. Ak je automatické prepočítavanie v činnosti, kopírovanie a presun príkazov zapríčini prepočet celej tabuľky.

Ak je automatické prepočítavanie vypnuté, kópia a presun jednoduchého presúva obsah vybraných blokov z jedného miesta tabuľky na druhé. Rovnako ako v originálnej verzii je možné prepínať automatické prepočítavanie pomocou CTRL-R.

Podľa F. Chapman: Enhancements For Atari Speedcalc, COMPUTERI
September 1986 upravil Ing. Ján Čelko, Žilina

DODATOK PRE ATARI SPEEDCALC

```

10 REM RUTINA ROZSIRENIA PRE SPEEDCALC
20 REM TENTO PROGRAM PRIPISUJE NIEKOLKO VSUVIEK DO ORIGINALNEHO SPEEDCALCU
30 TRAP 430
40 CHECKSUM=0:NBYTES=80
50 FOR BYTE=1 TO NBYTES:READ ABYTE:CHECKSUM=CHECKSUM+ABYTE:NEXT BYTE
60 IF CHECKSUM<>7369 THEN PRINT "CHYBA V DATACH":GOTO 440
70 DIM A$(1)
80 PRINT "(CLEAR)VLOZ DISK SPEEDCALC A STLAC RETURN":INPUT A$
90 CLOSE #1
100 OPEN #1,9,0,"D:AUTORUN.SYS":REM PRIPISANIE VSUVIEK NA KONIEC ORIGINALNEHO SU
BORU
110 RESTORE 170
120 PRINT "CAKAJ.. "
130 FOR BYTE=1 TO NBYTES:READ ABYTE:PUT #1,ABYTE:NEXT BYTE
140 CLOSE #1
150 PRINT "DONE":END
160 REM $1F00+$1F2B,PRVY DODATOK
170 DATA 0,31
180 DATA 43,31
190 DATA 162,0,32,199,58,32
200 DATA 88,46,173,17,66,205
210 DATA 1,66,144,240,173,1
220 DATA 66,133,205,173,19,66
230 DATA 205,2,66,144,227,169
240 DATA 65,160,79,162,0,32
250 DATA 199,58,32,89,33,162
260 DATA 4,96
270 REM
280 REM $1F40-$1F4A,DRUHY DODATOK
290 DATA 64,31
300 DATA 74,31
310 DATA 173,143,62,240,3
320 DATA 76,150,51,76,152,33
330 REM
340 REM $SCDB-$2CE0,NAHRADA 6 BYTOV V KODE SPEEDCALCU
350 DATA 219,44
360 DATA 224,44
370 DATA 32,0,31,32,199,58
380 REM
390 REM $31C9-$31CB,DODATOK KOPIA/PRESUN
400 DATA 201,49
410 DATA 203,49
420 DATA 76,64,31
430 ERR=PEEK(195):PRINT "ERROR-";ERR
440 PRINT "PROGRAM ZRUSENY!"
450 CLOSE #1

```

SCREEN SCROLLER

preložil: Milan Soka Zilina

Program je v Basicu s použitím strojovej rutiny na vertikálne rolovanie. Umožňuje posun textu po obrazovke zdola nahor v grafickom mode 2, ktorý je na jemné rolovanie najefektívnejší. SCREEN SCROLLER môže byť využitý aj ako podprogram na posuvanie správ vo vašich programoch. V tom prípade zmeňte riadok 9440 na: 9440 RETURN. Volanie potom prevádzkame GOSUB 9000.

Text, ktorý chceme posúvať sa zapisuje do DATA príkazov, ktoré začínajú na riadku 9500. Každý riadok obrazovky značí práve jeden DATA riadok a môže mať najviac 20 znakov. Prázdny riadok na obrazovke odpovedá DATA príkazu bez reťazca.

Ukončenie posunu správ možno realizovať niekoľkými spôsobmi, ale ešte predtým je vhodné na koniec textu dať niekoľko prázdnych DATA príkazov. Ak na posledný DATA riadok dáme END, program sa zastaví, alebo v prípade podprogramu vráti sa do hlavného programu.

Neustále vypisovanie textu dokola zaistíme tak, na posledný riadok napíšeme REPEAT namiesto END. Toto možno ukončiť stlačením <SPACE BAR>= medzerovníkom

RYCHLOST POSUVU:-možno ju regulovať pomocou premennej DEAY na riadku 9250. Hodnota 0 je najvyššia rýchlosť, minimálna je až 255.

AUTOMATICKÉ CENTROVANIE:-umožňuje organizovať text od stredu obrazovky. Vyradenie centrovacieho, a tým zobrazenie textu od ľaveho okraja obrazovky, sa nastaví premennou CENTER na 0:

```
9250 CENTER=0:DELAY=3
```

Opätovne sa centrovanie nastaví premennou CENTER na 1

FARBY:-Máme k dispozícii 4 farby pre text a jednu pre pozadie. Na riadku 9290 sú 4 príkazy SETCOLOR pre text a na riadku 9300 je príkaz SETCOLOR pre pozadie. Nastavenie je zrejme:

```
SETCOLOR register, farba, jas
```

Verím, že predložený program využívajúci hrubé i jemné rolovanie bude dobrou pomôckou pri vašej práci.

Jeff Brenner/ANALOG 1/87

HOT POKER

preložil: Milan Soka, Zilina

Tento kratky program, ktorý možno uložiť na začiatok alebo koniec programu, sa z pasky nahráva príkazom ENTER"C:". V pôvodnej verzii sa nachádza na riadkoch číslo 1-...., takže pozor pri nahrávaní aby ste si nepremazali svoj program, ktorý máte uložený v pamäti počítača. Pri vyvolaní HOT POKER-a príkazom USR, bude tento transformovať dekadické číslice 0-255 z dát príkazov do pamäte od určenej adresy, alebo do BASIC-ovského reťazca.

Pre porovnanie s klasickými POKE - príkazmi uvediem len nasledovné porovnanie: HOT POKER dokáže uložiť za sekundu 1675 čísel, kdežto BASIC-ovskými príkazmi je to len 16 čísel. Odpadá tým zdĺhavé pozeranie na obrazovku s nápisom " PLEASE WAIT...!"

V riadkoch 8-9 programu je stručne pripomenuté, ako volat strojový podprogram vo vašom programe. Pred uvedením 2 príkladov využitia este by som chcel upozorniť, že v DATA-riadkoch musí byť zapísaných práve 5 dekadických čísel v uz spomenutom rozmedzí 0-255.

Príklad 1:

```
10 U=USR (ADR(PKR$),1000,2,1536)
1000 DATA 216,104,104,133,209
1010 DATA 104,133,208,104,104
```

vykona sa presun 10 čísel v riadkoch 1000,1010 na adresy 1536 - 1545. Z toho je už asi zrejmé, ako sa podprogram uložený v reťazci PKR\$ využíva. Prvý parameter určuje počiatočný DATA riadok, druhý počet DATA riadkov a posledný adresu, od ktorej sa čísla majú ukladať v pamäti.

Príklad 2:

```
30 DIM ML$(15):ML$="*":ML$(15)="*":
40 U=USR(ADR(PKR$),2000,3,ADR(ML$))
2000 DATA 104,104,133,204,104
2010 DATA 133,203,169,0,197
2020 DATA 205,240,20,162,0
```

vyplní sa reťazec ML\$ znakmi, ktorých ASCII hodnoty sú v DATA príkazoch. Upozorňujem, že je nutné najprv reťazec vyplniť a až potom ho zmeniť pomocou opísaného strojového podprogramu.

Skúste v nejakom programe, kde sa ukládajú čísla z DATA-riadkov do pamäte či reťazca, použiť uvedené podprogram a budete prekvapení efektom!

XL - Basic , Atari Basic a Ased priamo v počítači.

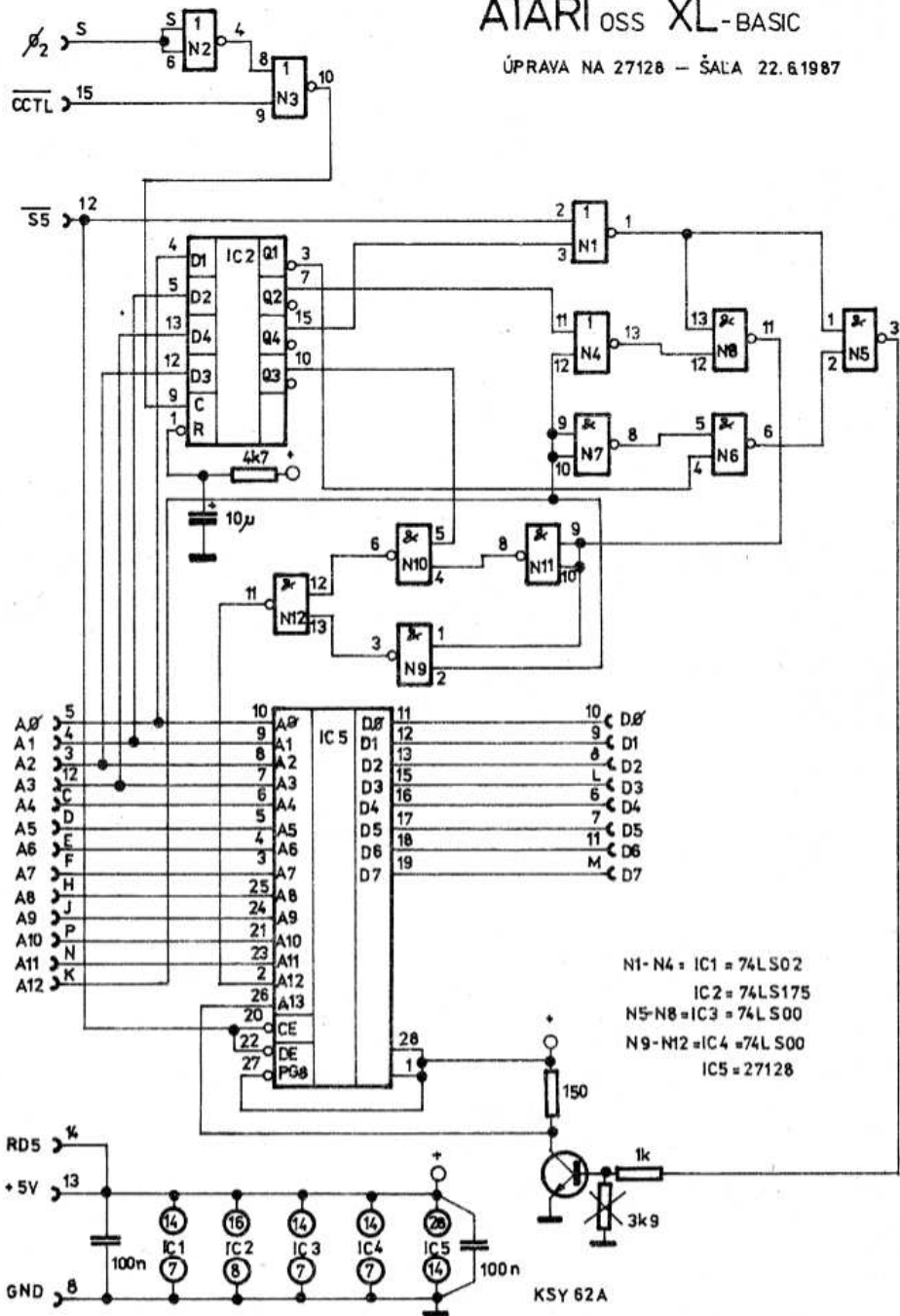
Mnohí, ktorí pracujú v Basicu vedia, aký je dobrý Basic XL. Tento zásuvný modul bol prerobený z dvoch pamätí EPROM typu 2764 na jeden typu 27128. Dôvod bol cenový rozdiel 20.-DM za dva rôzne typy EPROM, čiže ušetrených asi 60.-DM. Pre tých, ktorí majú možnosť doniesť si obvody zo zahraničia, to nie je jedno, koľko za ne zaplatia. Zhruba ďalšie 2.-DM je rozdiel medzi EPROM 27128 a 27256. Pamäť 27256 má kapacitu 32kB. Tak sa naskytuje možnosť využitia ďalších 2 x 8 kB na rôzne programové vybavenie. Preto som sa rozhodol upraviť XL-Basic modul na obvod 27256 a vmontovať ho priamo do počítača. Do voľných 2 x 8 kB pamäte som dal originál Atari Basic a assembler ASED. Miesto týchto programov sa samozrejme dajú použiť aj iné. To záleží na záujme užívateľa. S montážou do počítača nám zostane voľná zásuvka pre modul. Zasunutím modulu, ktorý pracuje od adresy A0000 sa vnútorný program vypne.

So vstavanými programami sa pracuje tak, ako s Basicom, čiže cez OPTION sa vypínajú. Na prepínanie jednotlivých funkcií slúži trojpolohový prepínač.

Ladislav PAPP ATARI KLUB ŠALA

ATARI OSS XL-BASIC

ÚPRAVA NA 27128 - ŠALA 22.6.1987



Úprava zásuvného modulu pre montáž do počítača s programom

XL BASIC

AT BASIC

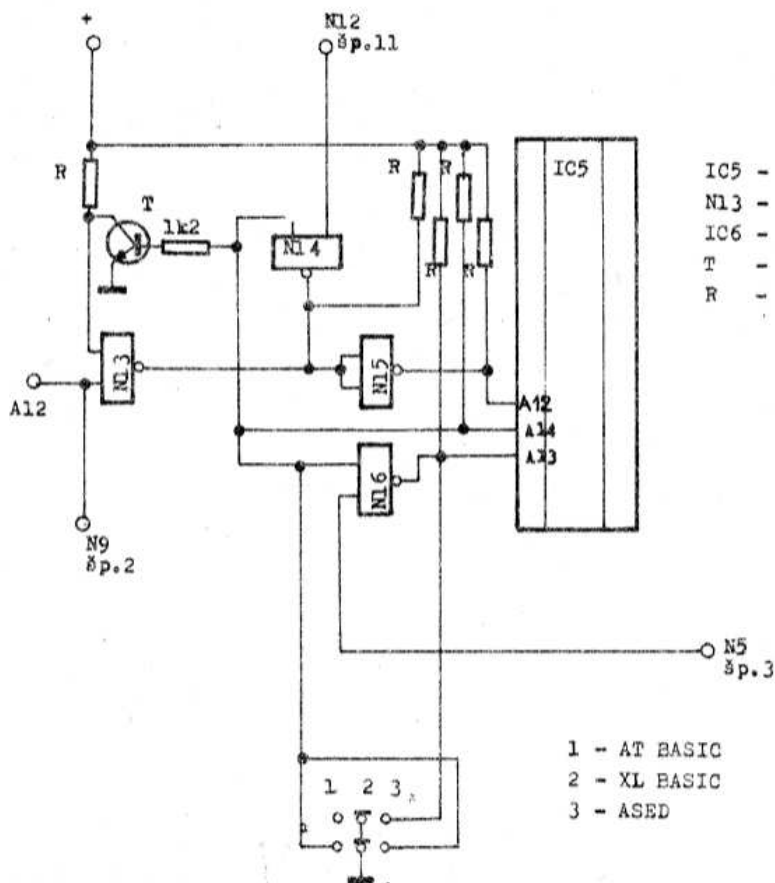
ASED

ASED	AT-BASIC	XL-BASIC
------	----------	----------

8k

8k

16k



IC5 - 27256

N13 - N16; - 74LS03

IC6 - 74LS03

T - KSY 62

R - 680Ω

1 - AT BASIC

2 - XL BASIC

3 - ASED

š - šierny

r - červený

i - biely

V XL Basicu sa všetky príkazy môžu písať jak malým tak inverzným písmom.

Inštrukcia INPUT dovoľuje jak pýtanie sa na viac položiek, tak vy-
nisať úvodný text. /input " Udať hodnotu",a,b,c/

V XL Basicu netreba dimenzovať stringové premenné AUTO DIM

Umožňuje tiež indexovanie string. premenných. Dimenzovanie -

Dim a\$(b,c) /b= počet string. premenných od 1 c= max.dĺžka
string.premennej

a\$(1;)="qwerty" ? b\$(#;) ? c\$(d;)

Fri index.premenných je dimenzovanie nutné - AUTO DIM nepracuje

V XL Basicu môžeme zadávať priamo hexadecimálne čísla /\$ D01A/

Obsadenie pamäte ako ATA-I BASIC. Navyše zaberá pamäťovú časť od
1536 po 1664 /input and floating point buffers/

BGET - BG.

Bget#a,b,c - uloží skupinu bytov /a = číslo kanálu
b = začiatková adresa c = počet bytov

BPUT - BP.

Bput#a,b,c - vyšle skupinu bytov /a,b,c ako pri Bget

BUMP

Bump(a,b) - kolízny register PMG grafiky
ak Bump (a,b)=0 - nie je kolízia medzi a,b

1 - kolízia medzi a,b

hráč/hráč a = 0-3, b = 0-3

strela/hráč a = 4-7, b = 0-3

hráč/pole a = 0-3, b = 8-11

strela/pole a = 4-7, b = 8-11

COM

Com je identický s Dim

CP

Cp je identický s Dos

DEL

Del(a,b) - maže Basic riadky od a po b

DIR

Dir"D2:*.BAS" - vypíše obsah disku na display

DPEEK

Dpeek a,b - vyčíta dvojbytové číslo /a= nižšia adr.
a+1=vyššia adr.

DPOKE - DP:

Dpoke a,b - vloží dvojbytové číslo /a ako u Dpeek
b môže byť 0-65535

ERASE - ER.

Erase"D:TEST.BAS" - maže súbor daného mena

ERR

Err(a) - chybové návěstí ak a=0 obsahuje číslo Error chyby
ak a=1 číslo Basic riadku kde bol Error

FAST - FA.

Zaznamená si adresy všetkých riadkov na ktoré ukazujú skokové príkazy. Pri chode programu skáče priamo na zazn. adresy - program bude rýchlejší.

FIND

Find(a\$,b\$,c) - hľadá b\$ v a\$ od pozície c
Ak najde = číslo pozície v a\$
Ak nie = 0

HEX\$

Hex\$(a) = hexadecimálne slo s a /a = decimálne číslo

HSTICK

Hstick(a) - udáva horizontálnu polohu joysticku. Vpravo =+1, vľavo =-1, stred =0 /a=číslo joysticku

IF...ELSE...ENDIF

If a:b:Else:c:Endif - ak a<>0 prevedie sa b, ak a=0 prevedie sa c. If a:b:Endif

LEFT\$

Left\$(a\$,b) - skráti a\$ na dĺžku b, ostane ľavá časť

LOMEM - LOM.

Lomem a - rezervuje si prázdný byt v pamäti z adresy a

LVAR - LV.

Lvar a: - vypíše všetky premenné a riadky, kde sa nachádzajú na zariadenie a
Lvar - vypíše na display

MID\$

Mid\$(a\$,b,c) - skráti a\$ na dĺžku c od pozície b

MISSILE - MIS.

Missile a,b,c - vytvorí strelu v PMG grafike /a=číslo strely 4-7 b=Ypozícia c=výška strely

MOVE - M.

Move a,b,c - presúva pamäťové bloky od adr. a na adr, b
c = dĺžka bloku v bytoch

NUM - NU.

Num a,b - automatické číslovanie Basic riadkov od a s k krokom b. Num - od 10 riadku s krokom 10.

PEN

Pen(0) -X pozícia svetelného pera
Pen(1) -Y pozícia

PMADR

Pmadr(a) = počítačová adresa PMG plochy / hráči a=0-3
strely a=4-7

PMCLR - PMCL.

Pmclr(a) - nulovanie PMG Plochy /a= ako u Pmadr

PMCOLOR - PMC.

Pmcolor a,b,c - nastavenie farieb PMG /hráči a=0-3
b=farba c=jas

PMGRAPHICS - PMG.

Pmgraphics a - nastavenie zobrazovania PMG /a=0 ruší
PMG a=1 jednoriadkové PMG
a=2 dvojriadkové PMG

PMMOVE - PMM.

Pmmove a,b;c - nastavenie pozície hráčov a striel
a= ako u Pmadr b=X pozícia c=pripočítava sa k Y pozícií /Pmovea,b Pmmove a;c

PMWIDTH - PMW.

Pmwidth a,b - určenie šírky /a=ako u Pmadr b=1 normálna
b=2 dvojnásobná b=4 štvornásobná šírka

PRINT USING - ? USING

Print using a\$,b - Print s možnosťou formátovania vypisovanej informácie b /a\$=tip formátu ak b je číslo#&#+
-\$. ak b je string %!
Príklady : x= prázdne miesto

b=	a\$=	výsledok
1	###	xx1
1234	###	234
12	&&&	012
12	***	+12
123.456	###.##	123.46
5216	##,###	x5,216
43.7	+###.#	+x43.7
-43.7	+###.#	-x43.7
3.75	++++.##	xx+3.75
-43.17	***.***	+43.17-
34.2	+\$###.##	+S34.20
1572563.4	\$\$,\$\$\$,\$\$\$.#+	\$1,572,563.4+
72.68	total=##.##	total=72.7+
25,19,7	xx##	xx25xx19xxx7
abc	%%%	xabc
abc	!!!!	abcx
abc	%%	ab

PROTECT - PRO

Protect"D:*.EXL" - chráni program pred vymazaním

RANDOM

Random(a,b) - generuje nahodné celé čísla od a po b
Random(b) = čísla od 0 po b

RENAME

Rename"D1:*.CAR,*.BAS" - premenuje súbor zCAR naBAS

RENUM

Renum a,b - prečísľuje Basic riadky /aj za Gosub,Goto,
Restore/ a= nové číslo prvého riadku
b= diferencia

RGET - RG.

Rget#a,b,c\$.... - uloží dáta do premenných b,c\$....atď.
a= číslo kanálu

RIGHT\$

Right\$(a\$,b) - ako Left\$ ale pravá časť.

RPUT - RP.

Rput#a,b,c\$.... - vyšle dáta premenných b,c\$....atď.
a= ako u Rget

SET

Set a,b - viacfunkčný príkaz /pb=pôvodné hodnota b

a=	pb=	b=	funkcia
0	0	0	normálny BREAK
		1	pri BREAK sa program chová,ako by prišlo k chybe /Error/
		128	ignoruje BREAK
1	10	3až127	dĺžka medzery pri použití /,ako tabul. pri PRINT
2	63	0až255	ATASCI kód pýtacieho znaku INPUT
3	0	0	For...Next sa uskutoční aspoň raz aj pri naplnení
		1	pri naplnení ani raz
4	1	0	Normálny INPUT
		1	INPUT sa môže pýtať na viac položiek/a,cS
5	1	0	normál
		1	malé a inver.prvky sa premenia na veľké
6	0	0	neinverz./Neprevádza medzi úvodzovkami/
		1	pri ERROR tlačí jeho číslo a popis
7	0	0	tlačí len čísla ERROR
		0	pri vert.vybehnutí v PMG sa hráč/strela stratí
		1	uzatvára sa vert. okruh
8	1	0	pri USRz Bsaicu nedáva počet prenášaných parametrov do zšobníku
		1	normál
9	0	0	???
		1	???
10	0	0	normál PMG
		1	všetky 4 strely sa zmenia na piateho hráča
11	40	1až255	pri auto dimenzovaní zadimenzuje string na uvedenú veľkosť
		0	ako ATARI Basic
12	1	0	normál list
		1	pri LIST oddeľuje programové konštrukcie

SYS

Sys(a) - ukazuje status daný systémom funkcií/pomocou SET/
a= ako u SET

TAB

Tab#a,b - progr.tab.Vytvára prázdne miesto s dĺžkou b.
a= číslo kanálu Tab b - na obrazovku

TRACE

pri chode programu sa budú vypisovať Basic riadky práve prevádzané

TRACEOFF

ruší trace módus

UNPROTECT - UNP.

Unprotect "D:*.*)" - ruší chránenie programov pomocou Protect

VSTICK

Vstick(a) - udáva vertikálnu polohu joysticku.

Hore = +1

dole = -1

stred = 0

/a = číslo joysticku/

WHILE - WH.

ENDWHILE - ENDW.

While a:b:Endwhile - b sa opakuje pokiaľ paltí podmienka a. Ak a=0 b sa neuskutoční. Ak a=1 b sa stále opakuje.

ATARI KLUB ŠALA

Program ASED je popísaný v "Průvodce assemblerem".

QMEG OS V3 (Quarter Mega OS)

Operacny system QMEG OS V3 je urceny pre pocitace ATARI 800XL, alebo 130 XE s moznostou ovladania RAMdisku.

Ovladanie pridanej pamete 192 Kb prebieha zatienenim 12-16K pametovych baniek v rozsahu adresy \$4000-\$7FFF cez PORTB(\$D301). QMEG obsluhuje vsetky Percom kompatibilne zrychlovace pre floppy 1050 (Happy, SPEEDY, Turbo) zaroven je implementovane rozhranie Centronics ovladane cez I/O stranky 6. Dodatočne sa moze pomocou tlaciarne Epson vylacit cela znaková sada ATARI (alebo lubovolna ina)

Oproti originalnemu OS, QMEG OS nevlada systemovu zbernicu cez SELF TEST, nakoľko sa tu už nenachadza. Taktiez medzinarodna znaková sada je vynechana. Aby boli texty nadalej citatelne ignoruje QMEG POKE 756,204.

Ostatne je QMEG OS, kompaktibilny s OS 800 XL. Urcite problemy vsak mozu nastat u programov, ktore vyuzivaju OS Ram stranku 2a3 pre vlastne ucely. Naopak su programy, ktore funguju s QMEG OS a ATARI 400/800, ale nie s OS 800 XL.

1.2 Zabudovanie QMEG OS

Zabudovanie je velmi jednoduché. Povodna ROM OS, sa vymeni za EPROM QMEG OS. Na testovanie QMEG zapneme pocitac pri vypnutom floppy disku ! Ak sa na obrazovke objaví OS Monitor s modrym pozadim je vsetko v poriadku. Ak sa objaví MLM monitor zo zelenym pozadim je EPROM vadny.

2. OS Monitor

OS monitor je ulozeny namiesto SELF testu. tadiaľ sa aj ovladaju vsetky funkcie QMEG OSu.

2.1 Skok do OS monitoru.

- a) z Basicu "BYE"
- b) z Dos II+/D "RUN E471"
- c) z MLM 2.1 "Q"
- d) pri zatlacenom Select a Reset

S kombinaciou Select a Reset sa dostanete v kazdom pripade do OS monitoru, aj ked je system "zruteny". Pamet sa pri zatlacení Select a Reset nevyymaze, neide teda o studeny start systemu.

Monitor sa ohlasi touto obrazovkou:

```
QMEG OS V3 (c)'87 S.Dorndorf
  WAITING
```

```
-----
Prg s: Drive 1 2 3 4 8! AB Ram
Ram : Type      A B A! MX Disk
-----
```

2.2 Ovladanie RAM disku

QMEG OS vie o dodatočných 192 k rozšírené 800 XL (130 XE) ovládať formou dvoch Ram diskov A a B. Jednotlivé možnosti sú:

Ram disk A	Ram disk B
------------	------------

- | | |
|---------------------|----------------|
| a) SS jedn.hust 90k | jedn.hust. 90k |
| b) MX stredna 128k | XE hustota 64k |
| c) D dvojita 180k | ziadna |

Zatlacením S, M, D sa môže zvolit ľubovoľný formát Ramdisku

A. Na obrazovke v pravo sa objaví jej skratka

Ram disk X je komp. k Ram disku D8:

v Dose II+/D.M aj D Ram disky sú komp.k Ram disku v Dose II+/D verzia 6.2.

Na využitie oboch Ram diskov sa musí zvolit jeho číslo 1 až 4 alebo 8 (A alebo B), pomocou klaves A a B.

2.3. Ovládanie zoradenia prívov pomocou kurzora

S medzernikom môžete kurzor (inverzne zaradiť na

zodpovedajúci drive 1 4, alebo 8. Zatlacením A, alebo B sa

zvolí číslo floppy disku #n v Ram disku A, alebo B. Všetky

zasaahy, ktoré robíte z Basicu, alebo z Dosu na ten ktorý drive budú zodpovedať zvolenému floppy Ram disku.

Spetne zaradenie sa deje pomocou klavesy "-". Zodpovedajúce

písmena sa objavajú pod číslom floppy disku v monitore.

2.4 Citanie a zápis Ramdisku A. (Kopírovanie diskiet)

Nacítanie celej diskety do Ram disku A, založte zdrojovú disketu do drivu #n a nastavte kurzor na "n". Potom zatlačte klavesu R monitor zostane zelený, t.j. obsah diskety sa nacítava .

PO ztlaceni Start sa Ram Disk A, formátuje a nacíta. Na

obrazovke sa ukazu čísla nacítavanych sektorov.

Pri vyskyte chyby máte nasledovne možnosti:

- Start - práve vykonávanú operáciu zopakovať.
- Select - preskocenie nasledovnej operácie, napr. vynechanie vadných sektorov.
- Option - nacítanie sa ukončí Po ukončení nacítavania obrzovka opät zmodra a obsah diskety je uložený v Ram disku A, ktorý má zodpovedajúci formát zdrojovej diskety.

Na prepísanie obsahu Ramdisku A, založte cieľovú disketu

do drivu a kurzorom zajdite na požadované miesto. Po

ztlaceni "W" sa pri červenej obrazovke a ztlaceni Startu

obsah Ramdisku najprv naformátuje a zapíše na cieľovú

disketu. Obdobne je možné kopírovať obsahy jednotlivých

Ramdiskov medzi sebou.

Ak nechcete formátovať cieľovú disketu miesto Start zatlačte Select.

2.5. Directory diskety v DOS 2.

Zatlačte L a zobrazí sa vám prvých 8 dát na diskete (alebo Ramdisku). Po ďalšom ztlaceni L sa objavajú zvyšné directory. S medzernikom sa dostanete opät do monitoru.

POZOR !.Pri zablokovanom Ramdisku musite zatlacit Shift L.Tak sa samozrejme RAM od \$7F00 az \$7FFF prepise!!

2.6. Natahovanie strojovych programov formatu COM a BIN.

Obdobne ako v bode 2.5 precitajte si obsah diskety(Ramdisku)pomocou "L".a zatlacte pozadovane cislo programu,ktory chcete natiahnut. Po tomto sa obsah celej pracovnej pameti vymaze a program sa natiahne ("Loadingxxxxxxxxxyy")a odstaruje.Strojovy program musi byt odstartovatelny aj bez DOSu.

2.7 Ovladanie Basicu a Cartidge.

klavesa funkcia skratka(Prg)

- a) Start-Z Basic rom ON BAS
- b) Start-C Cartidge ON CAR
- c) Start-X RAM do \$ C000 ON RAM

Basic resp. Cartridge sa startuju za studena.toho sa dosiahne kombinaciou Reset a Start

2.8 Start Dosu z Ramdisku C.

Ram disk C je vyhradeny pre max. 5.25 Kb. dlhy DOS.Pre odloženie DOSu dos Ram disku C,natiahnite DOS a zadajte (z Basicu) POKE 1792,196.Takto upraveny DOS sa odlozi na novu disketu.

Zakazdym ked budete tuto disketu natahovat,sa DOS automaticky natiahne do Ramdisku C.Na opetovny start DOSu z Ramdisku C zatlacte(z OS monitora) "Ctrl Caps".

S touto moznostou odpada opetovne natahovanie DOSu pokym nevypnete pocitac.

Pre pouzitie zodpoveda DOS II+/D (verzia 6.1 a 6.2),ktory spolupracuje s QMEG OSom v dvojansobnej hustote a roznymi Ramdiskmi.

Ramdisk C obsadzuje poslednych 44 sektorov Ramdisku "X".Preto je tento z QMEG OSu nepristupny(ERROR 139)

2.9. Rezim Ultra(Super,Turbo)a Happy.

Na ovladanie Floppy disku #n s Ultra Speedy chodte s kurzorom na "n" a zatlacte "U".Ak je vsetko v poriadku tak sa objavi "U" inac "--".Na vypnutie Happy zatlacte "H".Takto zapnuty pohon bude nadalej pracovat so zvyšenou prenosovov rychlostou.Na normal zatl. "--".

2.10. Dodatocne funkcie pre Happy.

Shift-I studeny start floppy disku
Shift-D vypnutie track Buffra(ktory urcuje prenosovu rychlost)

2.11. Kontrolovaný studený start

Prevádza sa automaticky (vid funkciu "U") alebo zatl. "N". U niektorých chránených programov použijeme "Shift Inverz" čím nastane studený start systému. Samozrejme z Basicu treba zatlačiť Option.

2.12. Zap/vyp Ramdisku a prepínanie baniek

Zatlacte "P" na zablokovanie Ramdisku. Obrazovka sa zafarbi na ružovo a adresa 54017 (PORTB) je chránená proti prepísaniu. To môže byť veľmi dôležité pre natiiahnutie programov pre radu 400/800. Aj (MLM nefunguje pri zablokovaní PORTB, nakoľko leží v oblasti kde má 800 XL Self test. Opätovným zatl. "P" sa systém odblokuje.

2.13. Funkcia \$

Pre majiteľov čierne bielych TV a monitorov, je pod znakom \$ zobrazený aktuálny stav OS monitoru.

- a) "-" normalná prevádzka
- b) "R" zatl. R (cítanie)
- c) "W" zatl. W (zápis)
- d) "P" Ramdisk blokovany

2.14. Opustenie OS monitora

- a) ESC navrat do DOSu
- b) Ctrl Caps Nový start DOSu z ramdisku C
- c) Reset navrat do DOSu, resp. Basic a Cartridge
- d) Shift Inverz kontrolovaný studený start
- e) Return skok do MLM 2.1

3. MLM 2.1

MLM 2.1 je jednoducho monitor na analyzovanie a zmeny stroj. programov.

3.1 Skok do MLM 2.1

Z OS monitora zatl. "Return" alebo z DOSu "RUN E480". Takmer všetky príkazy MLM sa začínajú adresou, ktorá sa nachádza v programovom citácii AD a je pevne stanovená. Pri týchto príkazoch sa môže adresa aj opustiť, pokiaľ sa v MLM nahradi hodnotou AD. Pamätová bunka PBCTL (\$ D303) sa nesmie z MLM meniť. Rovnako sa nesmú používať prerušenia v rozsahu \$5000 \$5800

3.2 Zadávanie v MLM.

- a) \$ všetky adresy hexadecimálne sa očakávajú a zadávajú. Dekadické adresy sa musia označiť #
- b) # všetky adresy dekadické sa očakávajú a zadávajú. Hexa. sa musia označiť \$

- c) ? zadanie nepreveditelne
- d) >xxx vyskyt I/O poruchy (dekadicky)

3.3 Prikazy MLM

- N- novy start MLM (alebo "\$E480 GO")
- Q- skok do OS mod. (alebo "\$E471 GO")
- H- zapnutie hex. modu \$
- Z- zapnutie deka. modu #

3.4 Jednoduché matemat. operácie v MLM.

- a) adr= zobrazenie adresy hex. alebo dekadicky
- b) adr+adr scitanie jednotlivych adr.
- c) adr-adr odcitanie jednotliv. adres
- d) = ukazuje hodnotu D

3.5 Zobrazenie pametovych oblasti

- a) adr; prvych 8 bytov sa zobrazi adr. hexa (resp deka.)
- b) adr ako a) ale pri hex. mode sa zobrazi prvych 8 bytov v ATASCII kóde
- c) adr" prvych 8 bytov od adresy a zada v ATASCII kóde
- d) adr' prvych 8 bytov od adresy su zadane v obraz. kóde
- e) adrL 20 riadkov od adr sa disasembuje

Pri vsetkych prikazoch sa moze namiesto "adr" pouzit "adr.adr", pricom sa zobrazi obsah medzi dvomi adresami. Ak je prikaz doplneni s "*" posledny prikaz sa zopakuje. Prerusenie je cez "Break".

Vytlacenie dat na tlaciaren cez "umocnenie"

3.6 Zmena obsahu pametovych miest.

- a) adr; byte byte byte
Pamet od adresy sa zaplni bytmi, ktore stoja za nou. Je mozne miesto bytov zapisat hodnotu (slovo >255) potom sa zapise najprv LO byte potom HI byte do pamete.
- b) adr" ATASCII znakovy retazec, uložia sa ATASCII znaky od adresy
- c) adr' obrazovy kod znakovy retazec ako b) ale pre ATARI obrazov. kod

3.7 Presuvanie , porovnavanie a plnenie pamete.

- a) adr1. adr2 M adr3
Obsah pam. od adresy 1 po adresu 2-1 sa presunie do adresy 3 prve byty z adresy 1.
- b) adr1. adr2 V adr3
Osah pam. z adresy 1 po adresu 2-1 sa sa porovnavaju

Spam. po adrsu 3.

Pri rozdielie sa zobrazia adr. a za nimi byty, s ktorymi sa porovnavalo.

c) adr1; byte

adr1. adr2 M adr1+1

Obsah pam. od adr1 po adr2 sa naplni bytmi v adre. 1

3.8 Prehľadavanie pracovnej pamete.

a) adr1.adr2;byte byte...

pracovna pamet sa prehlada od adresy 1 po adresu 2-1 v poradi bytov (alebo

slov) "byte byte...". Adresy najdenych miest su zobrazene.

b) adr1.adr2 "ATASCII zn.retazec ako v bode a) len s hladanim ATASC.zn.

c) adr1.adr2' kod obrazovky zn.retazec ako v bode a) s hladanim obrazov.znakov

3.9. Citanie/zapis sektorov Z/NA disk.

a) adr<secnr.count Nacitaju sa sektory od secnr po secnr+count 1

z diskety #1 a zapisu sa nad adresu.

b) adr>secnr.count Zapise sa obsah pam. nad adrsou az po sektor secnr

do secnr+count 1 na na disketu

Hodnota count musi byt mensia ako 255 (0=256). Obidva prikazy pracuju aj pri dvojnásobnej hustote zaznamu.

3.10. Zmeny registrov a start progamov.

a) R zobrazia sa obsahy registrov simulovaneho proc.6502

b) W=byte akumulator zachova hodnotu"byte"

c) X=byte reg.X zachova hodn. "byte"

d) Y=byte reg.Y zachova hodn. "byte"

e) P=byte priznak.reg.zachova "byte"

f) adr GO MLM natiahne reg. A,X,Y,P s hodnotami simul. 6502 a skoci

cez JSR na adresu kde je ulozeny strojovy program.

Ak narazi 6502 na BRK program sa v MLM prerusi a zavola sa s prikazom "R".

3.11. RAM obsadena s MLM.

a) nulna stranka:4 az 7,21,22,112 az 120

b) OS Ram:583 az 619,713 az 724.

Tieto adresy sa nesmu z MLM menit !!!

4. Funkcne klavesy.

- 4.1 SHIFT HELP
zablokovanie klavesnice
- 4.2 CTRL HELP
odblokovanie klavesnice

4.3 Funkcne klavesy 4 az 0

CTRL-4 zapina a vypina zvuk klavesnice

CTRL-5 zapina a vypina blikavy kurzor

CTRL-6 vypina obrazovku cim ATARI pocita o 30%
rychlejsie. Po zatlaceni
inej klavesy obrazovka zapnuta.

CTRL-7 rychlost Auto repeatu, na pomale, stredne a rychle.

CTRL-8 ako BREAK, ale vyhodnocuje sa len pri editovani z
klavesnice

CTRL-9 ak nie je kurzor v lavom rohu, skoci tam. Inak sa zap.
medzi 0 a ".

CTRL-0 urcuje kurzor do pozicie BOTTOM

5. OVLADANIE TLACIARNE

1. Vytlacenie znakovkej sady pomocou tlaciarne EPSON

Pomocou tlaciarne EPSON mozete bez tazkosti vytlacit vsetky
znakove sady, teda tlacit aj perfektne listingy, pricom
pouzijete prikaz "P:" ako skratku "P2:".

Na zaistenie formatu vami tlaceneho textu mozete zmenit
nasledovne adresy.

adresa	Default	funkcia	29	RAMTOP-B
vyssi Byte adresy		puffra grafiky		
30	60	max pocet znaciek na riadok		
31	60	pocet riadkov na stranu		
757	224	HI-byte adresy suboru znakov pouzitych pri vytlaceni		
758	'K	prikaz pre EPSON tlac 'K alebo 'L		

hodnoty Default sa po kazdom Resete nastavia na novo.
Dolezite: pri tlaci pomocou "P2:" musi mat ovladac tlaciarne k
dispozicii puffer 512 Bytov. HI-Byte zaciatku puffra sa musi
zadat do pametoveho miesta 29.

Po tolkych riadkoch, ako je nastavene v adrese 31, dojde
k automatickeму posuvu strany. 0 v adrese 31 vypne posuv
stran.

5.2 Tlac s dvojitou hustotou bodov.

-120 znakov na riadok. Do adresy 758 sa zada 'L (dekadicky 76) a adresa 30 sa zodpovedajucou zmeni.

Zmena adresy 758 a 30 pouzitim skratky "P3:gn":

g je prikaz grafiky EPSON K alebo L
n je pocet znakov na riadok, tie znamenaju
3 az 9:30 az 90 znakov
0 az 2:100 az 120 znakov

Priklad: list "P3:L2" mozete Basic listing vytlacit dvojitou hustotou pri 120 znakov na riadok.

Pozor: pri pouziti dvojitej hustoty, musi byt puffer tlačiárne dlhý 1kByte. Pri tlaci s "P2:" nestlacit BREAK !! Ked pouzijete prikaz ON v DOS II+/D musite pred prikazom k tlaci zatlacit RESET, pretoze inac by adresa 29 ukazovala na oblast BASIC-ROM.

5.3. Rozhranie Centronics.

Pomocou druhého PIA 6520, ktoré obsadzuje adresy \$D600-\$D6FF sa da zabudovat rozhranie Centronics k vystupu pre tlačiárne. Zapojenie portov PIA musi byt nasledovne:

PIA-pin	funkcia	centronics-pin
PA0-PA07	datove bity 0-7	2-9
PB0	DATA STROBE	1
PB6	BUSY	11
PB7	0=rozhr. zap, 1=vyp	-

ak je PD7 Low, tak sa pri kazdej SIO tlac. prikaze dotazuje rozhranie Centronics, inak normalne seriove rozhranie.

Pri tlaci cez Centronics sa vyskytuju nasledovne poruchove hlasy:

- 128 zatlacena klavesa Break
 - 138 po prikaze pre tlac vodič BUSY (PB6) nie je na 1. to znamena napríklad zly kontakt kabla tlačiárne.
- POZOR ! Ak je tlačiárne Off-line (napr. koniec papiera) tak QMEG-OS caka na On-line, alebo sa zatlacilo Break.

6. SIO a DISKINTERFACE

6.1 Diskinterface (DSKINV)

Pre programatorov v Assamlere dobre znamy Diskinterface DSKIF (adresa DSKINV (58451)) obhospaduruje teraz aj dvojitou hustotou. Povolene dodatocne prikazy su nasledujuce (hodnota je ulozena v DCOMND (770)):

- | | | | |
|----|----|---|---|
| a) | 34 | " | formatovanie v MD |
| b) | 35 | # | automaticke formatovanie v Percom-formate v dvojitej hustote (DD) |
| c) | 63 | ? | zapina rychly zapis a Ultra-Speed |
| d) | 79 | Q | zastavenie motora |
| e) | 72 | H | v zavislosti od DAUX1 a DAUX2 (778 a 779), vypina dodatocne Happy funkcie |

f) 83 S stavovy prikaz zapina teraz R W (citanie a zapis) a prikaz P automaticky na spravnu dlzku sektorov (adresa DSKSLN(725))

6.2 SIO a USIO

V QMEG-OSe su pritomne dva SIO podprogramy. Jeden je USIO, ktory pracuje bez V/V preruseni, teda bezi rychlejsie ako normalne SIO. USIO je volane pri vsetkych rychlych disketovych operaciach (Speedy a Happy).

Uprednostnene je normalne (nezmenene) SIO 800 XL pokiaľ nieje volany Ramdisk.

SIO vysiela 4 poruchove hlasedia:

1. 128 zatlacena klavesa Break
2. 139 prikaz nieje povoleny
3. 140 chyba v prenose dat
4. 144 prikaz je korektny, ale nepreveditelny

6.3 Ramdisky A a B

Ramdisky sa mozu vyuzivat cez SIO a DSKIF rovnako, ako Floppy disky. Ramdisky rozumeju aj signalom pre Percom-Put a Stavovym prikazom.

Po naformatovani Ramdisku A ma ten isty format aj Ramdisk B.

Ramdisky maju tieto chybove hlasedia:

1. 139 cislo sektora male alebo velke
2. 138 pokus o zapis do Ramdisku B naformatovanom na inu hustotu

Ak mate rozsirenie pamete 1. typu (vid kapitolu 8) pri pouzivani Ramdisku v rozsahu od \$4000-\$7FFF nesmie tu byt pouzity ziadny Display-list a obrazovkove kody - tym by doslo k zruteniu systemu.

Pri pristupe na Ramdisk zapina QMEG-OS Display-list ak je v DLIEN(787) hodnota vecsia ako 127.

6.4 Tabulka zoradenia pohonov

Tabulka je pevne stanovena podla cisla pohonov (pozri 2.3). Je ulozena v oblasti adries od 1000 po 1015 (pre pohony #1 az #8). Pohon #0 je stale identicky s Ramdiskom A a pohon #5 je identicky s pohonom #8.

obsah	zoradenie	skratka v OS-monitore
0 alebo >4	normalny pristup	"-"
1	Ultra-Speed	"U"
2	Ramdisk A	"A"
3	Ramdisk B	"B"
4	Happy-Warp Speed	"H"

7. ORGANIZACIA PAMETE

Nasledujuce miesta v pameti maju iny vyznam ako 800 XL.

.1 Nulta stranka

adresa strucny popis

0,1 oba Byty su volne a nevymazu sa ani pri studenom
 starte

4-7 MLM-nulta stranka(okrem ineho)

18-20 ATARI-hodiny,nevymazu sa ani pri Resete

21,22 MLM,OEMON-Fileloader,DSKIF-nulta stranka

28-31 "P2:"-nulta stranka

54,55 ukazovatko puffru grafiky pre "P2:"

74,75 volne pouzitelne

96,97 IX,IY-pre rutiny Drawto

112-120 MLM-nulta stranka a premenne Drawto

7.2 OS-Ram (stranky 2 a 3)

563 volne pouzitelne

568,569 VCTRLB vektor pre Ctrl-8

581 FMIFL format Percom Ramdisk u

128=single;0=medium;1=double density

583-619 OSMON-Display list,MLM- riadkovy puffer ,"P2:"
 puffer zmeny bitu znaku

648 FNKEYS 128=funkcne klavesy zapnute;0=vypnute

654,655 STTEMP(Stefanove tempo)tempo pre DSKIF

700,745 volne pre rozsirenje QMEG-OSu

713-724 premenne pre MLM a Percom puffer pre DSKIF

727 FLASL 128=blikavykurzor;0=vypnuty

728 FLASH 205=blikavy kurzor aktivny

735 PLINE riadkovy citac pre automaticky posuv stran
 pre "P2:"

757 PFONT adresa znakovej sady pre "P2:"-vyssi Byte

758 P2CMD prikaz pre tlac cez "P2:"

'K=normal;'L=dvojita hustota znakov

759 PMODE 1=tlac cez "P2:";0=tlac cez "P:"

760,761 PCOL,PTEMP dalsie premenne "P2:"

787 DLIEN 128=DLI zapnuty podla Ramdisku

832-959 IOCB je pouzite aj pre obrazovku monitora OS

1003 RSTAPH (<)pri Resete sa APPMHI nastavi na
 0(presne stanovj,ze Editor moze byt stale otvoreny)

1004,1005 DBAUD prenosove rutiny pre Ultra-Speed

1006 PBLOCK 128=po Resete je Ramdisk blokovany

1007 TYPEA format Ramdisku A

1008-1015 ASSIGN zoradovacia tabulka floppy pohonov

1017 ROMCSM hodnota Checksumm pre cely QMEG-OS

1018-1020 volne pouzitelne

8. HARDWARE

QMEG-OS V3 spolupracuje s nasledovnym hardwarom.

8.1. Rozsirenja pamete

Ramdisky mozu mat 256k alebo 320k. Ovladanie dodatocnych pameti prebieha ako pri 130 KE zatienenim 16k pametovych baniek v rozsahu adries \$4000-\$7FFF cez jednotlivie bity portu B (\$D301).QMEG-OS vyuziva dva typy obsadenia portu B:

- 1) PORTB funkcia
 c. bitu
 7 0=zapn. Self test; 1=vypnutý(ako pri 800
 XL)
 4 0=prístup na ANTIC a CPU cez bity 2,3,5 a 6
 podľa zvolenej 16k banky v rozsahu \$4000-\$7FFF; 1=normalný
 prístup
- 2) PORTB (320k rozšírenie Compy Shop)
 c. bitu
 7 0=Self test zapnutý ak bity 4a5=1; ináč
 vypnuté
 5 0=len prístup ANTIC
 4 0=len prístup CPU
 cez bity 2,3,6 a 7 navolené 16k banky v
 rozsahu \$4000-\$7FFF; 1=normalný prístup pamäte

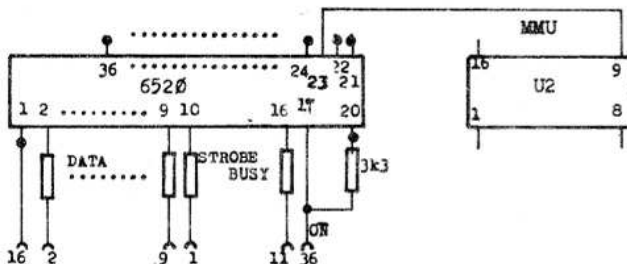
Pri oboch typoch rozšírenia sa využívajú banky c.1 až 15 (precisovanie je závislé od 4 zvolených bitov portuB)
 Rozlíšenie oboch typov prebieha automaticky.

8.2. Rozšírenia pre floppy disk 1050

QMEG-OS rozoznáva všetky tzv. Percom komp. rozšírenia pre 1050, pri dvojnásobnej hustote dát t.j. 180-kB (ako sú RANA, Happy 1050, Speedy 1050 a Turbo 1050).

Majitelia Happy a Speedy 1050 si prídu na svoje pretože QMEG-OS zabezpečuje prenosovú rýchlosť cca. 52000 Baudov. Treba si len zapamätať že pohony #1 a #2 môžu byť ľubovoľne (jeden Happy a jeden Speedy). Pohony #3 a #4 musia byť toho istého prevedenia ako pohon #1.

preložil a za prípadné
 chyby nezodpovedá
 prekladateľ amatér,
 Emil Gaspar Atari klub SALA
 29.05.1988



Konektor CENTRONICS

Vydáva: ZO Zväzarmu ELEKTRONIKA Martin

Zodpovedný redaktor: ing. Horvath Ján

Zostavil: Boďa Dušan

Názov: Zborník z 5.Celoslovenského seminára o počítačoch

Počet strán: 74

Obálku navrhol: Šproch Július

Tisk: NADAS Vrútky

Náklad: 3 000 kusov

Vydanie: 1 vydanie 1988 Neprešlo jazykovou úpravou

Schválil: OK ONV Martin č. 48/88

Len pre vnútornú potrebu Zväzarmu.

