

ÚNOR

FUN

1993

with Commodore

časopis uživatelů Commodore 64/128

4. číslo

BASIC
DYCP grafika
RS 232
MIDI
A 600
ASSEMBLER

SLOSOVÁNÍ A VÝHRY!
5.000,- Kč
1. cena

INZERCE

Nabízím komplet programů pro obsluhu zapisovače ALFI počítačem C64.

Bližší informace na adrese:

*Mottl Vladimír,
Hátka 389, 267 24 HOSTOMICE*

Prodám tiskárnu BT100 + kabel k C64 + ovládací software na kazetě nebo disku za 400,- Kč. Cartridge „Eprom karta V2.0“ obsahující Simoncs Basic, Turbo ROM V2, Copy, Turbo64, program pro nastavení mgf. hlavy, Monitor stroj. kódu, plp. klávesnice a definici funkčních. kláves za 500,- Kč. Cartridge „Super Games“ obsahující 3 hry za 140,- Kč:

Michal Motalík, U Trojáku 4596, 760 05 Zlín

1/ Nabízím vlastní Slovník počítačových výrazů - anglicko/český - na disketě 5,25" za 50,- Kč + poštovné za dobírku tj. 21,- Kč, nebo na kazetě za 65,- Kč + poštovné, případně včetně výtisku na 25 stranách formátu A4, tedy plus dalších 30,- Kč!

2/ Zašlu Seznam počítačových her na 15 stránkách formátu A4 (celkový počet 1350 her) vážným zájemcům, kteří pošlou předem platné poštovní známky v hodnotě 30,- Kč (t.j. 10 ks po 3,- Kč), nebo Seznam různých počítačových programů, ve stejném rozsahu i ceně (hudební, učební, karetní, databázové, uživatelské, výpočetní, jazykové a grafické)

*Luboš Slaný,
sídl. 9. května 954, 790 01 Jeseník*

Prodám GEOS 2.0 (1099,-), GEOPUBLISH (1000,-) - originálně bal., tlač. MPS 1230 + kábel (4000,-):

*Peter Kuna, Hviezdoslavova 18, 900 21 Svätý Jur,
tel.: 071975 75*

PD programy C64/128 - 1000 disků - hry, užít, GEOS:

*fa SYNIV, Pod vrchem 2889, Mělník,
tel. 0206/670759*

Prodám nový C64-II + VC 1541-II + 20 disket (Geos...) + joystick. Cena: 10 000,- Ks.

Info: Peter Kačmarik, Budovatelská 9, 080 01 Prešov

Chcete bez problémů projít autoškolou? Kupte si program na zkoušení z pravidel sil. provozu na C64. Cena včetně diskety 45,- Kč.

Info: Petr Labounek, Heyrovského 6, 779 00 Olomouc

Vážení čtenáři,

předkládám vám k posouzení první číslo nového ročníku časopisu FUN. Vedle rubrik, na které jste byli zvyklí (Basic, Assembler), v něm najdete pokračování seriálu o grafice a pár horkých novinek.

Jak záhy zjistíte, dali jsme tentokrát větší prostor vašim kolegům, čtenářům FUNu. Čtenářské příspěvky jsme mimo rubriku Dopisy a Tipy a triky zařadili také do praktické rubriky pro programátory Basicu. Doufám, že vás tyto příspěvky zaujmou stejně, jako zaujaly mne.

V průběhu minulého roku se na našem trhu objevila řada levných tiskáren. Množí se dotazy na způsob jejich připojení k C64 resp. na vhodné programové vybavení. Rozhodli jsme se vyjít vám vstříc a přiblížit vám problematiku výměny dat mezi C64 a periferními přístroji. V tomto čísle začínáme se seriovým přenosem dat a jeho praktickým využitím v MIDI interfejsu a u již zmíněných tiskáren s interfejsem RS232 (Robotron, D100 a další).

Pro naše slovenské zákazníky mám dobrou zprávu. Dohodli jsme se s firmou Prington Slovakia z Nitry o zastupování našich zájmů na Slovensku. Prostřednictvím této firmy bude probíhat ve Slovenské republice distribuce výrobků firmy Comotronic i časopisu FUN. Doufám, že spolupráce bude probíhat k naprosté spokojenosti vás, zákazníků.

Však dosti řečí! Pojděte, prolistujte časopis a začtěte se. Nezapomeňte nám sdělit vaše dojmy. Čekáme na váš ohlas.

redaktor 1. čísla FUN
Jaroslav Vančura
Šumperk, únor 1993

Dovoľte, aby som vám predstavil svoj nový program „Písací stroj“. Tento program je vytvorený pre grafickú jednotku XY 4150, ktorá sa pomocou tohto programu premení na univerzálny písací stroj. Popis najdete v priloženom manuáli. Program mám v disketovej verzii. Prednosti programu sú opísané v texte.

Program nahrajte príkazom LOAD „pisaci stroj“, 8,1 a zadajte príkazy
SYS 50000: SYS 21500: SYS 21800: RUN. Vypíše sa hlavné menu.

P - priame vkladanie. Po stlačení písmena sa hneď vypíše na papier.

N - nepriame vkladanie. Zadanie písmena musíte potvrdiť returnom.

Priamy mód:

1. Zadajte uhol smeru tlače a uhol sklonu písmena. Uhly sú od 0 do 360.
2. Rýchlosť pera od 1-najpomalšie do 4-najrýchlejšie.
3. Veľkosť písma.
4. Obsahovanie X, Y. X-koľkokrát, Y-posun.
5. Nastavenie ľavého okraja.
6. Priamy výpis písmien.
7. Prechod do nepriameho modu.

Ad 6. Priamy výpis písmen:

Malé písmená vytlačíte po stlačení priamo, ako aj číselky. Veľké písmená po stlačení klávesy SHIFT + písmeno, ako aj pomocné znaky. Písmená so slovenskou diakritikou získate pomocou klávesy C= + písmeno. Tlačítko C= používame pri písaní dlžňa, mäkčeňa, dvojbodky a krúžku.

Veľké písmená: Cf, Cg, Ch, Cj.

Malé písmená: Cv, Cw, Cm, Cx.

Tlačítko C* vracia pero na nový riadok.

Tlačítko C+ a C- posunie pero o 0,1 mm.

Šipka (klávesa v ľavom hornom rohu) vás vráti späť do menu. Tlačítko C@ vás presunie do textovej obrazovky.

Nepriamy mód:

V menu sú všetky funkcie rovnaké, ako v priamom mode.

Nepriamy výpis písmen:

V nepriamom mode musíte každé písmeno potvrdiť returnom a až potom sa vám vytlačí na papier. Pri písaní dlžňov a iných znakov postupujte podľa príkladu: Dňé a. Najprv prejdite s kurzorom na dlžeň a stlačte return. Vytlačí sa dlžeň a potom stlačíte klávesu a + return.

Textová obrazovka

1. Text: text, ktorý napíšete sa uloží do pamäti. Všetky tlačítka sú rovnaké ako v priamom mode. Text ukončíte returnom jedno políčko pred koncom riadku. Pri dokončení slova na konci riadku stlačte return a aby ste dodržali odskok stlačte na klávesnici medzeru.
- 2.-3. Load, Save: zadajte názov súboru.
4. Výpis textu z pamäti na papier.
5. Directory: vypíše obsah diskety. Pre návrat späť stlačte medzeru.
6. Návrat späť do menu.
Z textu do menu sa dostanete klávesou C@ a return. Obrazovku vymažete takto: zadajte 1. text a hneď prvé písmeno zadajte klávesou C@, týmto sa text vymaže.

Záver

Pri zakladaní papiera dbajte na dostačnú hĺbku zasunutia. Po zachytení papiera posúvajte list pomocou funkcie 5.- nastavenie okrajov.

Pri nedostatečnej dĺžke okrajov vám počítač vyhodí chybu a zastaví program, ktorý musíte spustiť príkazom run a znova nastaviť parametre. To isté platí aj o pravom okraji. Ak prekročíte 70 znakov na riadok (výška písma 2,8), program sa preruší.

Program zašlem v disketovej verzii. Vážny záujemcovia môžu písať na adresu: Vladimír Bielka
Jalšové 132
922 31 Trnava

BT 100

Ve druhom čísle časopisu FUN byl v rubrice „Dopisy čtenářů“ dotaz na připojení tiskárny BT 100 k C64. Zapojení i program podle L. Černého byl otištěn ve starším čísle časopisu Elektronika. Jedná se o samostatnou rutinu pro výtisk grafické paměti od adresy 8192 do 16191. Tiskárna je připojena přes user-port. Protože tiskárna tiskne pouze v jednom směru, trvá hardcopy obrazovky asi 20 minut. I přes tuto časovou náročnost však může program dobře posloužit při vytištění obrázků z ART - studia nebo z některých her. Zájemcům pošlu proti známce výpis programu, stručný manuál a popis připojení tiskárny BT 100 k C64:

Marek Uher
Těřeškovové 2257
734 01 Karviná - Mizerov

Jak mám nahrát vlastní programy ve strojovém kódu na mgf. pásek, aby obsahovaly basicovský řádek s příkazem SYS tak, jako v každé hře.

(M. Uher)

Problém

Většina z vás jistě ovládá dokonale svůj počítač, programy tvoříte levou rukou. Chybí vám zrovna nápady co programovat a stáváte se „herními maniakky“ ? Předkládám vám k řešení můj problém.

Máte čtvercové pole rozměrů 5x5 čtverců, ve kterém se pohybuje šachový kůň. Kůň chodí v poli podle pravidel šachu.

Projděte koněm všech 25 polí tak, abyste na každé pole vstoupili pouze jedenkrát. Řešení je jednoduché, hledání algoritmu řešení už méně. Můj program (kompilova-

ný basic) hledal řešení 5s. Ale po odstartování z jiného políčka jsem musel test po 1,5 h přerušit psychicky zcela vyčerpán.

Plán pole:

| | a | b | c | d | e |
|---|---|---|---|---|---|
| 1 | * | * | * | * | * |
| 2 | * | * | * | * | * |
| 3 | * | * | * | * | * |
| 4 | * | * | * | * | * |
| 5 | * | * | * | * | * |

Postup koně píše podle vzoru: start c3, d1, b2... (celkem 25 dvojic)

Opatřete své řešení listingem a pošlete mi jej na adresu:

Jan Stodůlka
U pergamenky 8
170 00 Praha 7

BASIC

Komfortní práce s drajvem.

Bylo již popsáno několik způsobů jak v Basicovém programu ošetřovat příkazy pro práci s disketovou jednotkou. Rutiny jsou dlouhé a přes jejich zdánlivou dokonalost může při chybné manipulaci u drajvu dojít k přerušení programu ! Nabízím proto krátkou strojovou rutinu, která práci s periferií značně ulehčí a odstraní popsané nedostatky. Tyto rutiny komunikují přímo s drajvem a nenarušují činnost již otevřených souborů. Po každém úkonu se lze dotázat na chybový kanál a zjišťovat tak stav. Pokud je první znak textu hlášení "0" (ASCII 48) bylo vše v pořádku. Rutina obsahuje LOAD " ", 8, 1; SAVE " ", 8 (části paměti); výpis directory na obrazovku; DOS " " příkazy a výpis textu chybového kanálu. Příkazy (LOAD SAVE DOS) potřebují zadat text názvu. Ten se zapíše od bajtu 700 v podobě ASCII kódů, do bajtu 780 se zadá počet znaků názvu a odešle se (SYS) na příslušnou rutinu. U SAVE paměti je ještě třeba zadat počáteční (Flag 193) a koncovou + 1 (Flag 174) adresu, pak teprve SYS. Při použití LOAD se data uloží na své místo v paměti a po LOAD není provedeno RUN(program pokračuje dál). Directory a chybový text voláme přímo bez přípravy. Basicová data programku jsou psána v HEXA tvaru, aby šla použít i při zápisu monitorem (od SCB20). Rutina musí být umístěna na 52000. (Kdo by jich chtěl jinam, musel by přepsat skoky.)
Syntaxe příkazů:

LOAD : zadat název, poč. znaků pak SYS 52024

SAVE : zadat název, poč. znaků, zač. a konc. adresu pak SYS 52034

DOS : zadat text, poč. znaků pak SYS 52016

CH. kanál: SYS 52040 (výpis je proveden přímo na obrazovku)

DIREC : SYS 52080 (výpis je proveden přímo na obrazovku)

Před použitím příkazů není třeba zjišťovat zapnutí dra-

jvu. Nebyl-li dostupný, příkaz se ignoruje. Kdo neví jak flag použít, pro toho uvádím příklad: [SAVE 20480 - 24575] tedy POKE 193,0: POKE 194,80: POKE 174,0: POKE 175,96

Zde je vlastní zdrojový program:

```

10 FOR E=52000 TO 52143: READ A$: FOR F=1 TO 2
11 A=ASC(MID$(A$,F)):A=A-48+7*(A>64)
12 B=16*B+A:NEXT:POKE E,B: S=S+B: B=0:
NEXT
13 IF S<>19477 THEN PRINT " CHYBA
V DATECH ' '14 :15 DATA 84, B6, A2,
BC, A0, 02, 20, BD
16 DATA FF, A2, 08, A4, B6, 4C, BA, FF
17 DATA A0, 6F, 20, 20, CB, 4C, D5, F3
18 DATA A0, 01, 20, 20, CB, A9, 00, 4C
19 DATA D5, FF, 20, 20, CB, 4C, ED, F5
20 DATA A9, 08, 20, B4, FF, 30, 11, A9
21 DATA 6F, 20, 96, FF, 20, A5, FF, 20
22 DATA D2, FF, C9, 0D, D0, F6, F0, 0D
23 DATA A2, EE, BD, E2, A0, 29, 7F, 20
24 DATA D2, FF, E8, D0, F5, 4C, 46, F6
25 DATA A9, 24, 8D, BC, 02, A9, 01, A0
26 DATA 60, 20, 32, CB, A9, 08, 20, B4
27 DATA FF, A9, 60, 20, 96, FF, A0, 06
28 DATA AA, 20, A5, FF, 48, A5, 90, D0
29 DATA 1A, 68, 88, D0, 05, 20, CD, BD
30 DATA A0, 00, 10, EC, 20, D2, FF, D0
31 DATA E8, A5, C6, D0, 07, A9, 0D, A0
32 DATA 04, D0, F1, 68, 4C, 46, F6, 00

```

(J.Renda)

TOOL

Jistě jste si stačili všimnout některých nedostatků Basicu. Následující strojová rutina nabízí mnoho dobrých funkcí:

1. goto gg
2. pause pa
3. poke po
4. at at
5. get gt
6. open op
7. fill fi
8. change ch
9. pomocné rutiny

Předtím než začnete testovat program, napište si tyto řádky (pište je před každým níže uvedeným vzorovým programkem).

5 GG=49167: SP=49180: PA=49201: GT=49269:
 OP=49370: CH=49564
 7 PO=49173: AT=49222: FI=49452

Goto

Basic V2.0 nedovoluje výpočet řádky, tato rutina ano:
 10 A=10: SYS GG,A*10
 20 STOP
 100 PRINT "OK"

Pause

Příkaz není snad nutno ani představovat. Velikost pauzy je 0 - 65535.
 10 PRINT "start": SYS PA,20000: PRINT "konec"

Fill

Tento příkaz otevírá brány k profesionálnímu ovládnutí obrazovky. Má následující syntaxi:

SYS FI,R,S,D,V,B,ZN,K

Symboly označují R-řádek, S-sloupec, D-délka, V-výška, B-barva, ZN-znak (poke-kód), K-kód. Kódem lze nastavit: bit 0=1 - barvu, bit 1=1 - znaky, bit 2=1 - inverze dané oblasti. Kombinováním těchto bitů získáte různé možnosti. Př.:

SYS FI,R,S,D,V,B1,BC,8

způsobí přepsání všech barev B1 na barvu BC. Vyzkoušejte si zadat tyto příklady:

SYS FI,0,0,40,25,7,1,3

SYS FI,10,10,20,3,1,1,4

SYS FI,0,0,40,25,7,0,8

Chci poznamenat, že kód 1 používá pouze barvy. Číslo kódu znaky může být libovolné (0-255).

Change

Mění znak v řetězcové proměnné. Příkaz má následující syntaxi:

SYS CH,pozice,znak(ASCII),proměnná

10 AS="AHOJ"

20 SYS CH,3,65,AS

Pozor! Necháte-li řádku 10 tak, jak je, bude přepsána.

Nutno ji upravit na:

10 AS="AHOJ": AS=AS+""

neboť proměnná musí být nějak zadefinována.

Open

Příkazem jdou dělat skvělé posuny obrazovky nebo

uschovat jednotlivé části. Syntaxe příkazu je:

SYS OP,R,S,D,V,kód,adresa

Kód 0 byl zvolen pro čtení z paměti na obrazovku, kód 1 opačně. Adresa udává, kam bude daná oblast paměti uložena.

10 SYS OP,0,0,40,25,1,20000: PRINT"<CLR>":SYS PA,500

20 SYS OP,0,0,40,25,0,20000

Důležité je zachovat proměnné D a V, řádek a sloupec můžete měnit. Pro posuny obrazovky se dá napsat:

SYS 0,0,39,25,1,20000

SYS 0,1,39,25,0,20000 (doprava)

Poke

Toto rozšíření uvítá každý, kdo pracuje se zvukem či sprajty. Namísto složitého zápisu:

10 POKE 2040,13:POKE 2041,13:POKE 2042,13

lze napsat jednodušeji:

10 SYS PO,2040,13,13,13

At

Na umístění kurzoru do určité pozice existují různé triky. Nejlepší je tohle:

10 SYS AT,sloupec,řádek, dále dle syntaxe pro print

Get

Pokud chcete povolit vstup jen některých kláves, tak máte vyhráno:

10 SYS GT,"1234567890": AS=CHRS(PEEK(780))

20 PRINT AS: GOTO 10

ASCII kód stisknuté klávesy je na adrese 780. SYS čeká, dokud není stisknuta požadovaná klávesa.

10 SYS GT," "

Tato syntaxe čeká na stisk jakékoli klávesy.

Nyní ještě nezbytný budící program:

80 FOR A=49152 TO 49583: READ Q:
 POKE A,Q: S=S+Q: NEXT

85 IF S<>55062 THEN PRINT "Chyba
 v datech": END

90 PRINT "OK"

100 DATA 32, 253, 174, 32, 158, 173,
 76, 247, 183, 32, 253, 174, 76,
 158, 183, 32, 0, 192, 76

110 DATA 163, 168, 32, 0, 192, 133,
 252, 132, 251, 32, 9, 192, 138,
 160, 0, 145, 251, 230, 251

120 DATA 208, 2, 230, 252, 169, 44,
 209, 122, 240, 236, 96, 32, 0,
 192, 230, 20, 230, 21, 160, 0

130 DATA 1 36, 208, 253, 198, 20, 208,
 247, 198, 21, 208, 243, 96, 32,
 9, 192, 134, 38, 32, 9, 192

140 DATA 164, 38, 224, 25, 176,
 14, 192, 40, 176,

10,24,32,10,229,32,253,174,76,157,170

```

150 DATA 76, 8, 175, 32, 253, 174,
169, 255, 133, 13, 32, 158, 173,
32, 130, 183, 133, 38, 96, 32
160 DATA 101, 192, 165, 198, 240,
252, 32, 180, 229, 164, 38, 240,
86, 136, 209, 34, 240, 81
170 DATA 200, 136, 208, 247, 76, 120,
192, 32, 9, 192, 134, 38, 32, 9,
192, 134, 39, 32, 9, 192
180 DATA 138, 240, 195, 24, 101, 39,
201, 41, 176, 188, 134, 40, 32,
9, 192, 138, 240, 180, 24
190 DATA 101, 38, 201, 26, 176, 173,
134, 41, 32, 9, 192, 134, 42,
96, 160, 4, 232, 202, 240, 9
200 DATA 24, 105, 40, 144, 248, 200,
76, 192, 192, 170, 152, 24, 105,
212, 133, 254, 132, 252
210 DATA 134, 253, 134, 251, 96, 32,
143, 192, 32, 0, 192, 165, 39,
166, 38, 32, 189, 192, 165
220 DATA 21, 133, 152, 165, 40, 24,
101, 20, 133, 151, 144, 2, 230,
152, 164, 40, 136, 165, 42
230 DATA 240, 14, 177, 251, 145, 20,
177, 253, 145, 151, 136, 16,
245, 76, 22, 193, 177, 20, 145
240 DATA 251, 177, 151, 145, 253,
136, 16, 245, 165, 152, 133, 21,
165, 40, 24, 101, 151, 133
250 DATA 20, 144, 2, 230, 21, 230,
38, 198, 41, 208, 181, 96, 32,
143, 192, 198, 40, 32, 9, 192
260 DATA 134, 150, 32, 9, 192,
134, 151, 165, 39, 166, 38, 32,
189, 192, 164, 40, 165, 151, 41,
1
270 DATA 240, 9, 165, 42, 145, 253,
136, 16, 251, 164, 40, 165, 151,
41, 2, 240, 9, 165, 150, 145
280 DATA 251, 136, 16, 251, 164, 40,
165, 151, 41, 4, 240, 11, 169,
128, 81, 251, 145, 251, 136
290 DATA 16, 247, 164, 40, 165, 151,
201, 8, 208, 15, 177, 253, 41,
15, 197, 42, 208, 4, 165, 150
300 DATA 145, 253, 136, 16, 241, 230,
38, 198, 41, 208, 173, 96, 32,
9, 192, 134, 251, 32, 9, 192
310 DATA 134, 252, 76, 101, 192, 32,
143, 193, 197, 251, 144, 10,
164, 251, 240, 6, 136, 165
320 DATA 252, 145, 34, 96, 76, 8,
175

```

Elektronické bicí

Následující krátký program umožní napodobit elektronické bicí. Program pozorně opište do počítače a spusťte příkazem RUN.

```

100 S= 54272: FOR A=S TO S+24: POKE A,0: NEXT
110 POKE S+5,5: POKE S+6,5: POKE S+24,15
120 P=16: E=128: DIMR(P-1),B(3)
130 B(0)=0: B(1)=20: B(2)=80: B(3)=250
140 POKE 53280,0: POKE 53281,0
150 PRINT CHR$(147);CHR$(30)
160 PRINT " CÍSLA DOSTUPNÝCH EFEKTU: 0,1,2,3 "
170 PRINT
180 FOR A=0 TO P-1: PRINT R(A);CHR$(157);: NEXT:
PRINT
190 PRINT
200 PRINT „F1 - SPUSTENI RYTMU, F3 - NASTAVENI
RYTMU“
210 GET AS: IF AS<CHR$(133) OR AS>CHR$(134)
THEN 210
220 ONASC(AS) - 132 GOTO 300,400
300 FOR A=0 TO P-1
310 POKE S+1,B(R(A)): POKE S+4,E+1
320 POKE 1145+2*A,(R(A)+48) OR 128
330 FOR D=0 TO SZ: NEXT
340 POKE S+4,E
350 POKE 1145+2*A,R(A)+48
360 NEXT
370 GET AS: IF AS<> „ “ THEN 210
380 GOTO 300
400 FOR A=0 TO P-1
410 POKE 214,11: PRINT: PRINT „ “; R(A);
CHR$(157);CHR$(157);CHR$(157); CHR$(157);
420 INPUT R(A): IF R(A)<0 OR R(A)>3 THEN 410
430 POKE 1145+2*A,(R(A)+48) OR 128
440 NEXT
450 INPUT "RYCHLOST";SZ
460 GOTO 210

```

Postiskem klávesy F3 se zadá vlastní rytmus. K vytvoření rytmů se volí čísla od 0 do 3. 0 funguje jako pauza, ostatní čísla jako údery v rytmech. Údery a pauzy se skládají do 16 místných skupin (jejich počet se dá rozšířit změnou hodnoty proměnné P v řádku 120). Po naplnění všech míst se program zeptá na rychlost, kterou má být rytmus později zahrán. Zadat můžete čísla v rozmezí 0 až 200. Čím vyšší číslo zvolíte, tím půjdou údery pomaleji za sebou. Po zadání celého rytmu jej můžete vyzkoušet stiskem klávesy F1. Poslech se dá ukončit stiskem kterékoliv klávesy. Pro zadání nového rytmu stiskněte klávesu F3. Závěrem uvádím několik příkladů krátkých rytmů:

| | | | |
|------|------|------|------|
| 1100 | 2001 | 0010 | 0020 |
| 1110 | 3001 | 1010 | 3022 |
| 1021 | 0121 | 1120 | 1120 |
| 3333 | 2222 | 1111 | 0000 |

1110 2021 0210 2030

(Jan Stodůlka)

(D. Šutera)

ASSEMBLER NA C64

(3. pokračování)

6.6 LOGICKÉ PŘÍKAZY

Kromě možnosti spojení pomocí příkazů pro sčítání a odčítání ovládá procesor 6502/6510/7501 také tzv. logické příkazy. Pod tímto pojmem jsou schovány logické funkce AND, OR a EXCLUSIVE-OR. K uplatnění logických funkcí potřebujeme stejně jako k matematickým funkcím dvě hodnoty, první z těchto dvou hodnot je vždy uložena v akumulátoru, zatímco druhá může být procesoru sdělena nejrůznějšími způsoby adresování. Při logickém spojení obou hodnot dohromady jsou logickou funkcí zpracovávány jednotlivé bity obou čísel, t.j. dvojice bitů 0 - 7. Výsledek tohoto logického spojení je vždy uložen v akumulátoru.

1. Logické spojení AND

Při logickém spojení AND bude odpovídající bit ve výsledku nastaven v tom případě, že jsou nastaveny tyto bity u porovnávaných čísel. Dále jsou uvedeny výsledky logické funkce AND podle hodnot bitů obou čísel:

0 AND 0 = 0
0 AND 1 = 0
1 AND 0 = 0
1 AND 1 = 1

Jako příklad si uvedeme logické svázání čísel \$59 a \$F4.

\$59 = %01011001
\$F4 = %11110100

Logický součet těchto dvou čísel:

```
%01011001
AND %11110100
-----
= %01010000
```

Výsledek logického spojení čísel \$59 a \$F4 je \$50. Analogicky funguje logické spojení také v procesoru. Potřebný příkaz zní i pro procesor AND. V assembleru zapsaný příkaz vypadá takto:

LDA \$59 ;uložení první hodnoty do akumulátoru
AND \$F4 ;spojení s druhou hodnotou

Výsledek \$50 je po provedení příkazu uložen v akumulátoru. Příkaz AND se používá například tehdy, když je třeba testovat určité bity nějakého čísla. K tomu jeden příklad:

Chceme testovat, zda 6. bit v paměťovém místě \$D8 je nastaven. To se dá udělat například následovně:

LDA \$D8 AND %01000000 ; 6. bit testován

Po provedení obou těchto příkazů kontrolujeme obsah akumulátoru. Pokud je roven 0 (nulový příznak je nastaven), pak nebyl testovaný bit nastaven. Pokud je obsah akumulátoru různý od 0 (nulový příznak je vymazán), byl testovaný bit nastaven.

Kromě nulového příznaku bude po provedení příkazu nastaven také negativní příznak, pokud bude výsledek větší než \$7F, to znamená, že bude ve výsledku nastaven bit 7. Které způsoby adresování jsou pro příkaz AND použitelné, zjistíte v tabulce na konci kapitoly.

2. Příkaz OR

Také při logickém spojení příkazem OR jsou obě hodnoty porovnávány po jednotlivých bitech. Pokud je v prvním čísle nebo ve druhém čísle porovnávaný bit nastaven, je nastaven i odpovídající bit ve výsledku. Pro příkaz OR (nebo) platí následující pravidla:

0 OR 0 = 0
0 OR 1 = 1
1 OR 0 = 1
1 OR 1 = 1

K tomu jeden příklad, ve kterém spojíme dříve testovaná čísla logickým příkazem OR.

\$59 = %01011001
\$F4 = %11110100

```
% 01011001
OR % 11110100
-----
= % 11111101
```

Jak vidíte, je výsledek roven 1 i v případě, že jsou obě hodnoty rovny 1. Tento druh logické funkce NEBO se označuje také inclusive OR.

Ve strojovém jazyku procesoru je pro tuto funkci určen příkaz ORA. Zápis tohoto příkazu v assembleru je potom následující:

LDA \$59 ;uložení první hodnoty do akumulátoru
ORA \$F4 ;logické svázání s druhým číslem

Logický příkaz OR je používán většinou k nastavení bitu bez ovlivnění bitů ostatních, k čemuž dojde například při použití příkazu STA. K nastavení například 6. bitu v paměťovém místě SD8 slouží dále uvedená programová sekvence:

```
LDA SD8 ;obsah SD8 do akumulátoru
ORA %01000000 ;logické spojení s bitem 6
STA SD8 ;výsledek uložit zpět do SD8
```

3. Příkaz EXCLUSIVE-OR (EXOR)

Příkaz EXCLUSIVE-OR funguje téměř stejně jako příkaz OR. Jediný rozdíl mezi těmito dvěma příkazy je, že u příkazu EXOR je výsledek roven 0, pokud jsou bity v obou prováděných číslech nastaveny (rovny 1). Tento druh logického příkazu NEBO se nazývá také exclusive OR, na rozdíl od předchozího inclusive OR. Pravdivostní tabulka příkazu EXOR je následující:

```
0 EXOR 0 = 0
1 EXOR 0 = 1
0 EXOR 1 = 1
1 EXOR 1 = 0
```

K tomu opět pro názornost příklad s čísly \$59 a \$F4:

```
$59 = %01011001
$F4 = %11110100
```

```
%01011001
EXOR %11110100
-----
= %10101101
```

Využití nachází příkaz EXOR například tam, kde je třeba invertovat číslo nebo bit. Proto se k tomuto logickému příkazu vrátíme ještě později.

Ve strojovém jazyku je pro příkaz EXCLUSIVE OR určena instrukce EOR. Podle toho pak bude v assembleru vypadat zápis příkazu:

```
LDA $59 ;uložení první hodnoty do akumulátoru
EOR SF4 ;logické svázání s druhou hodnotou příkazem EXOR
```

Kromě již dříve uvedeného příkazu AND je ve strojovém jazyku příkaz BIT. Také jeho funkce je spojena s logickou funkcí AND, ale nedochází k ovlivňování žádných registrů. Pokud procesor při zpracovávání programu narazí na instrukci BIT, pak proveden logickou operaci AND mezi obsahem akumulátoru a adresovaným paměťovým místem. Pokud je výsledek operace 0, nastaví se nulový příznak. Dále bude 6. bit adresovaného paměťového místa převzat jako příznak přetečení a 7. bit adresovaného místa jako negativní příznak. K tomu jeden příklad:

```
BIT $1000
```

AKumulátor obsahuje hodnotu \$1000. Logické spojení AND pak vypadá následovně:

```
$E3 = % 11100011
$BC = % 10111100
```

```
% 11100011
AND % 10111100
```

```
-----
= % 10100000
```

Nulový příznak bude vymazán, příznak přetečení a negativní příznak nastaveny (viz hodnota \$E3 v binárním vyjádření).

Pro příkaz BIT jsou použitelné jen oba způsoby adresování nulté stránky a absolutní adresování:

A na závěr kapitoly je zde slíbená tabulka se všemi možnými způsoby adresování pro příkazy AND, OR, EOR:

| Druh adresování | příklad | AND | ORA | EOR |
|-------------------------|--------------------------|------|------|------|
| přímé | mnemokód bajt | \$29 | \$09 | \$49 |
| absolutní | mnemokód \$adresa | \$2D | \$0D | \$4D |
| absolutní indexované X | mnemokód \$adresa,X | \$3D | \$1D | \$5D |
| absolutní indexované Y | mnemokód \$adresa,Y | \$39 | \$19 | \$59 |
| nulté stránky | mnemokód \$1b.adresa | \$25 | \$05 | \$45 |
| nulté str. indexované X | mnemokód \$1b.adresa,X | \$35 | \$15 | \$55 |
| nulté str. indexované Y | mnemokód \$1b.adresa,Y | --- | --- | --- |
| nepřímé | | | | |
| indexované | mnemokód (\$1b.adresa),Y | \$31 | \$11 | \$51 |
| indexované nepřímé | mnemokód (\$1b.adresa,X) | \$21 | \$01 | \$41 |

6.7 PŘÍKAZY PRO POROVNÁNÍ CMP, CPX, CPY

Nyní se dostáváme k velmi důležité skupině příkazů, příkazům pro porovnání. Pomocí těchto příkazů je možno porovnávat dvě hodnoty mezi sebou. První hodnota musí být vždy v akumulátoru, registru X nebo Y, zatím co druhá hodnota může být sdělena různými způsoby adresování.

Princip srovnávání dvou hodnot je následující. Druhá hodnota bude od první odečtena, čili bude odečtena hodnota adresovaného bajtu od obsahu jednoho z uvedených registrů (aku, X, Y). Toto odčítání se však děje uvnitř procesoru, neovlivní se tím obsah žádného registru s výjimkou příznaků.

Příkazy pro porovnávání ovlivňují zvláště příznak přenosu, nulový, negativní následujícím způsobem:

C=1 Příznak přenosu se nastaví, když je první z porovnávaných hodnot větší nebo rovna druhé hodnotě. Odečtení dá tedy kladný nebo nulový výsledek. Návěští přenosu = 1 znamená, že H1 je větší nebo rovna H2

C=0 Příznak přenosu se vymaže, když je první hodnota menší než druhá, to znamená, že výsledkem rozdílu je záporná hodnota. Návěští přenosu = 0 znamená, že H1 je menší než H2

Z=1 Nulový příznak je nastaven tehdy, když jsou obě hodnoty shodné, to znamená když výsledkem rozdílu je nula. Nulový příznak = 0 znamená, že H1 = H2

Z=0 Pokud je nulový příznak po srovnávání vymazán, znamená to, že obě porovnávané hodnoty jsou různé. Nulový příznak = 0 znamená, že H1 je různé od H2

N=1 Pokud je po porovnávání negativní příznak nastaven, znamená to, že výsledek odčítání je větší, než \$7F. Je to ten případ, kdy jsou splněny následující podmínky:

1. H1 + \$80 je menší než H2 (pro H1 < H2)
2. H1 + \$80 je větší než H2 (pro H1 > H2)

N=0 Pokud je po porovnávání negativní příznak vymazán, znamená to, že rozdíl mezi oběma hodnotami byl menší než \$80.

1. H1 + \$80 je větší než H2 (pro H1 < H2)
2. H1 + \$80 je menší než H2 (pro H1 > H2)

Pod H1 je rozuměna ve všech případech hodnota uložená v akumulátoru, registru X nebo Y. Hodnota H2 je vždy uložena v paměti a procesoru sdělena povoleným způsobem adresování.

Nyní k jednotlivým příkazům

Procesor 65XX zná porovnávací příkazy CMP, CPX, CPY. Příkaz CMP porovná hodnotu s obsahem akumulátoru, Příkaz CPX s obsahem registru X a CPY s obsahem registru Y. Pro příkaz CMP je použitelná většina druhů adresování (viz tabulka na konci kapitoly).

Nyní k jednotlivým příkazům. Budeme se podrobně zabývat jen příkazem CMP, protože příkazy CPX, CPY jsou s ním principiálně shodné.

V prvním příkladu vyjdeme z předpokladu, že v akumulátoru je obsažena hodnota hodnota \$4F, která má být porována s hodnotou \$3B. Interní rozdíl vypadá takto:

```

$4F LDA $4F
- $3B CMP $3B
-----
= $14 C=1; Z=0; N=0

```

Výsledek interního rozdílu byl v našem případě \$14. Protože přitom nedošlo k podtečení, bude přenosový příznak nastaven. Obě hodnoty se lišily, to znamená, že nulový příznak bude vymazán. Nakonec bude vymazán také negativní příznak, protože výsledek rozdílu byl menší než \$7F.

Ve druhém případě jsou obě hodnoty shodné. Vezme-me například hodnotu \$3B pro obě hodnoty, pak bude operace vypadat následovně.

```

$3B LDA $3B
- $3B CMP $3B
-----
= $00 C=0; Z=1; N=0

```

Jak vidíte, byl výsledek rozdílu roven nule, což se projevilo v odlišném nastavení příznaků. Protože při výsledku rovném nule nejde ještě o podtečení, nastaví se příznak přenosu. Také nulový příznak se tentokrát nastaví, protože obě hodnoty byly identické a výsledek byl roven 0. Nakonec negativní příznak zůstane také nenastaven, protože výsledek byl menší než \$7F.

Ve třetím příkladě zvolíme první hodnotu menší než druhou, to znamená, že výsledek vnitřního rozdílu bude roven 0. H1 bude \$0B a H2 bude \$3B.

```

$0B LDA $0B
- $3B CMP $3B
-----
= $D0 C=0; Z=1; N=1

```

Protože v tomto případě byla druhá hodnota větší než první, došlo při odčítání k podtečení, proto byl příznak přenosu vymazán. Obě hodnoty nebyly identické, proto je vymazán nulový příznak. Negativní příznak je v tomto případě nastaven, protože výsledek (\$D0) je větší než \$7F.

Interní rozdíl lze také porovnat s rozdílovým příkazem SBC. Význam pro příznaky je stejný. Jen v akumulátoru není uložen výsledek, pouze příznaky jsou změněny.

Nyní si můžeme shrnout, že pokud srovnáváme dvě hodnoty mezi sebou, musí platit následující:

- H1 > H2 když C=1 a Z=0
- H1 = H2 když Z=0
- H1 > H2 když C=0
- H1 >= H2 když C=1

K ověření, zda H1 > H2 musí se nutně otestovat dva příznaky.

A na závěr obvyklá tabulka možných způsobů adresování

| Druh adresování | příklad | CMP | CPX | CPY |
|-----------------|------------------------------|------|------|------|
| přímé | mnemokód bajt | \$C9 | \$E0 | \$C0 |
| absolutní | mnemokód \$adresa | \$CD | \$EC | \$CC |
| absolutní | | | | |
| indexované X | mnemokód \$adresa,X | \$DD | — | — |
| absolutní | | | | |
| indexované Y | mnemokód \$adresa,Y | \$D9 | — | — |
| nulté stránky | mnemokód \$1b.adresa | \$C5 | \$E4 | \$C4 |
| nulté str. | | | | |
| indexované X | mnemokód \$1b.adresa,X | \$D5 | — | — |
| nulté str. | | | | |
| indexované Y | mnemokód \$1b.adresa,Y | — | — | — |
| nepřímé | | | | |
| indexované | mnemokód (\$1b.adresa),Y | \$D1 | — | — |
| indexované | | | | |
| nepřímé | mnemokód (\$1b.adresa,X)\$C1 | — | — | — |

6.8 PODMÍNĚNÉ SKOKY BEQ, BNE, BCS, BCC, BMI, BPL, BVS, BVC

K rozhodování podle určitého výsledku a nastavení příznaků slouží takzvané skokové příkazy. Podle toho, zda jsou příznaky nastaveny, nebo nikoliv, dochází v programu k takto definovaným skokům.

Pro každý z příznaků – nulový, přenosu, negativní, přetečení – existují vždy 2 příkazy skoku. Jeden příkaz, při kterém dochází ke skoku v případě, že příznak je nenastaven a druhý, kdy dochází ke skoku tehdy, když je příznak nastaven. Kromě příkazu ke skoku potřebuje procesor u každého příkazu ke skoku také cílovou adresu. Pod cílovou adresou se rozumí novou hodnotu programového čítače, tedy místo, odkud si má procesor vyzvednout další příkaz. Normálně jsou adresy v 16. bitovém formátu. Tento druh adresování platí i pro adresování skoků. Kromě toho se zde však objevuje nový druh adresování, tzv. adresování relativní.

Protože při podmíněném adresování se málo kdy musí skákat přes velký adresový prostor, vymysleli autoři procesoru 65XX a 75XX nový druh adresování, relativní. Co si však máme pod tzv. relativním adresováním představit?

První rozdíl proti normálnímu adresování je v tom, že adresa není definována dvěma bajty, jak je to obvyklé, ale jen jedním. Podmíněné příkazy skoku nesestávají tedy ze tří, ale jen ze dvou bajtů. To má jednu přednost, že zpracování příkazu je rychlejší, než u tří bajtů dlouhých příkazů.

Nyní ještě k principu nepřímého adresování.

Jak již sám název říká, je skok zadáván relativně. Aby bylo možno adresu zadávat relativně, je nutno mít výchozí bod, v našem případě adresu, potom podmíněný příkaz skoku. Relativní zadání cíle skoku může vypadat třeba následovně:

„skoč ze současné adresy o 10 paměťových míst vpřed“
nebo

„skoč ze současné adresy o 0 paměťových míst vzad“

Zvláště důležité v tomto případě je, že se skáče z momentální adresy. Pokud program v paměti posunu, u relativních skoků není třeba měnit adresy skoků, které jsou postaveny na vzdálenosti paměťových míst od sebe a ne na jejich přesných adresách. Podmíněné skoky se tedy nevztahují absolutně k určitým paměťovým místům.

Jak již víme, je možno s pomocí bajtu zobrazovat různých čísel (0–255). Tento hodnotový rozsah tedy stačí ke skoku maximálně o 256 bajtů vpřed nebo vzad. Lepší by samozřejmě bylo, kdyby se dalo skákat stejně vpřed jako vzad. Dala by se přitom jedním bajtem zobrazit jak kladná, tak negativní čísla pro skok. Snad si teď vzpomenete na příkaz SBC, nebo na souvislosti s negativním příznakem. Tam jsme se dozvěděli o možnosti zobrazit jedním číslem jak záporná, tak kladná čísla. Význam přitom hrál 7. bit jako předznaménko. Pokud byl v čísle bit č.7 nastaven, znamenalo to, že číslo je záporné. Pokud byl bit 7 vymazán, tedy roven 0, jednalo se o zcela normální kladné číslo. Zcela stejné to bude u podmíněných příkazů skoku. Také zde se bude na 7. bit nazírat jako na předznaménko. Pokud je nastaven, skáče se vzad, pokud je vymazán, skáče se vpřed.

Při výpočtu záporného čísla se postupuje následovně.

Všechny bity čísla se invertují (0 se nahradí 1 a 1 se nahradí 0) přidá se jedna. Srozumitelně vyjádřeno, je možno při výpočtu záporného čísla postupovat následovně.

Pokud je 7. bit čísla nastaven, je toto posuzováno jako záporné číslo. Absolutní hodnota (hodnota bez předznaménka) čísla se spočítá takto: 128 mínus hodnota bitů 0–6. Před výsledek se jen přidá znaménko mínus a již máte hodnotu našeho záporného čísla.

Pokud vám výpočet cílové adresy připadá komplikovaný a neprůhledný, mohu vás uklidnit. O výpočet cílové adresy podmíněného skoku se postará editor strojového kódu nebo dobrý assembler. Cílovou adresu zadáte absolutně (například BEQ \$1000) a editor strojového kódu spočítá sám parametr příkazu skoku a vypíše vám jej na obrazovku. Jak vyplývá z použití 7. bitu jako znaménka, je možno podmíněně skákat o 127 bajtů vpřed nebo vzad.



(pokračování)
(JK)

Čoština na C64.

Jistě je příjemnější, komunikuje-li s vámi počítač česky. Jak však použít české znakové sady ve vlastních programech? Jedna z možností je tato: Natáhnout českou znakovou sadu od adresy - \$3800 do \$4000 - např.:

```
10 IF A=0 THEN A=1 : LOAD "name",8,1
Tuto znakovou sadu lze získat např. úpravou ROM znakové sady pomocí některého monitoru (SMON Plus). Aktivace této znakové sady se provede takto:
```

```
100 POKE 53272, PEEK ( 53272 ) AND 240 OR 14:
RETURN
Původní znaková sada je aktivována takto:
```

```
200 POKE 53272, PEEK ( 53272 ) AND 240 OR 21:
RETURN
```

Délka vlastního programu je v tomto případě pochopitelně omezena umístěním znakové sady na 12 kB.
(P. Záruba)

Grafika a text

Po zadání příkazu:

```
SYS 49152, začiatkový riadok, koncový riadok, grafika:
začiatková adresa - sa v určených riadkoch obrazovky
inicializuje grafika a v ostatných zostáva textový režim.
(Pre úplných začiatkových: grafika-začiatková adresa
používajte 24). Program som napísal v Basicu. Takuto
„hlúpu“ formu zápisu som si zvolil z viacerých dôvodov.
Z vlastných skúseností viem, že keď som bol úplný
začiatovník nemal som žiadny Monitor ani žiadny
MachineLanguageEditor od Markt & Technik, iba Basic.
Tak keď som chcel písať assembler, robil som to takto
a myslím si, že začiatovníci aj tak z výpisu assemble-
rovského programu nič nemajú a takto to môžu použiť:
```

```
10 FOR A= 49152 TO 49271
11 READ B12 POKE A,B
13 S=S+B
14 NEXT
20 IF S<>14020 THEN PRINT "CHYBA"
50 DATA 32, 253, 174, 32, 158, 183, 138, 10, 10, 10
51 DATA 105, 50, 141, 113, 192, 32, 253, 174, 32, 158
52 DATA 183, 138, 10, 10, 10, 105, 50, 141, 87, 192
53 DATA 32, 253, 174, 32, 158, 183, 142, 82, 192, 120
54 DATA 169, 1, 141, 26, 208, 169, 127, 141, 13, 220
55 DATA 173, 113, 192, 141, 18, 208, 169, 68, 141, 20
56 DATA 3, 169, 192, 141, 21, 3, 88, 96, 238, 25
57 DATA 208, 169, 94, 141, 20, 3, 169, 59, 141, 17
58 DATA 208, 169, 24, 141, 24, 208, 169, 250, 141, 18
59 DATA 208, 76, 129, 234, 238, 25, 208, 169, 68, 141
60 DATA 20, 3, 169, 27, 141, 17, 208, 169, 21, 141
61 DATA 24, 208, 169, 130, 141, 18, 208, 76, 49, 234
```

Intromaker

```
10 POKE 53280,0
11 POKE 53281,0
12 PRINT "(SHIFT/CLR)C=/4)"
13 AD=49152
```

```
20 PRINT "(CTRL/RVSON) (SPACE) (CTRL/RVSOFF)
(LEFT)";
21 GET AS
22 IF AS="" THEN PRINT "(SPACE) (LEFT)"; GOTO 20
30 PRINT "(SPACE) (LEFT)";
40 IF AS="(F1)" THEN 70
50 POKE AD+A,ASC(AS)
60 PRINT AS;
61 A=A+1
62 GOTO 20
70 PRINT "(SHIFT/CLR) (C=/4)";
80 FOR B=1 TO A
81 PRINT CHR$(PEEK(AD-1+B))"(CTRL/RVSON)(SPA-
CE) (CTRL/RVSOFF) (LEFT)";
82 FOR C=1 TO 50: NEXT
83 PRINT "(SPACE) (LEFT)";
90 NEXT
```

Na tento program existuje veľa krajších a lepších prevedení, ja ho posielam iba kvôli jeho jednoduchosti. Po spustení je obrazovka prázdna a môžeme napísať ľubovoľný text, používať obrazovku, meniť farbu presne ako v editore Basicu. Keď sme hotoví, stlačíme kláves F1 a všetka naša doterajšia činnosť sa objaví odznova na obrazovke. Ak uchováme príslušnú časť pamäte (začína na adrese 49152 a má džku A), tak si neskôr znova môžeme prezrieť náš výtvar.

(prosím autora o zaslání adresy)

Autoload

Díky souboru "verze" lze automaticky jak ukládat, tak i nahrávat.

```
10 OPEN 1,8,0,"verze,seq":INPUT#1,V: CLOSE 1
20 PRINT "<CLR>LOAD"CHR$(34)"navez prg."V-.1;
CHR$(34)"8,;
```

```
30 POKE 198,2,: POKE 631,19: POKE 632,13
```

Doporučuji uložit tento program jako jednonázvový (PR: "b" -BOOT), pro rychlejší zápis příkazu. Nyní stačí zadávat:

```
LOAD "b",8
```

```
RUN
```

a nejnovější verzi máte v paměti počítače bez zdlouhavého vyhledávání v directory. Chci jen podotknout, pokud jste zvyklí psát PRINT jako otazník, pak při používání příkazu PRINT# vám naskočí chyba. Systém totiž rozlišuje dva zcela odlišné příkazy a to PRINT a PRINT#. Proto je nutné ten druhý vypisovat.

(Jan Stodůlka)

Autosave

Také se potýkáte s problémem, jak označit nové verze programu, který zrovna děláte? Stál jsem před tímto problémem. Používal jsem označení "název programu.. + den pořízení. Toto řešení je dobré. Po týdnu už ovšem může přestat vyhovovat. Změnil jsem dny na čísla a ukládání programu předal počítači. Následující pro-

gram nabízí efektivní archivaci programu, mazání starých a označování nových verzí.

```
1000 OPEN 1,8,0,"verze,seq":INPUT#1,V:CLOSE 1
1010 SAVE "navez prog"+str$(V),8
1020 OPEN 15,8,15,"S:naz.prog."+str$(V-.3):
PRINT#15,"S:verze,seq":CLOSE 15
1030 OPEN 1,8,1,"verze,seq":PRINT#1,V+.1:CLOSE 1
```

Podprogram lze nastartovat přímo z hlavního programu a pak dále pokračovat (bez přerušení). Před prvním použitím je nutno zadefinovat soubor "verze":

```
10 A=1: OPEN 1,8,1,"verze,seq":PRINT#1,A:CLOSE 1:
NEW
```

Nasazením a využíváním tohoto programku odpadá zdlouhavé hledání v directory diskety.

Nakonec ještě upozornění! Zkontrolujte zda počet mezer mezi názvem prog. a číslem verze souhlasí se starým značením.

(Jan Stodůlka)

Čára

Obdivujete barevné pruhy přes celou obrazovku a chcete je mít také? Vaše přání se dá splnit. Připomenu několik důležitých adres:

49206 - Barva pruhu

49204 - První rastrová řádka, kde dojde ke změně barvy

49211 - Poslední rastrová řádka, kde dojde ke změně barvy

Při určité kombinaci rastrových řádek vznikne ten správný efekt a obrazovka začne blikat jako u 16 - bitu. Chcete-li zvýraznit barevně pouze některou řádku, zadejte:

```
POKE 49204,50+8*řádka (0 až 24) POKE
49211,58+8*řádka (0 až 24)
```

Nyní už jen zdrojový program.

```
10 FOR A=49152 TO 49277: READ Q: POKE A,Q:
S=S+Q:NEXT
20 IF S<>15093 THEN PRINT "Chyba v datech":END
30 POKE 49206,5: POKE 49204,50: POKE 49211,58:
SYS 49152
100 DATA 120, 169, 127, 141, 13, 220, 173, 13,
220, 169, 51, 162, 192, 141, 20, 3, 142, 21, 3
110 DATA 169, 50, 162, 27, 160, 1, 141, 18, 208, 142,
17, 208, 140, 25, 208, 140, 26, 208, 88, 96
120 DATA 205, 18, 208, 208, 251, 142, 32, 208,
142, 33, 208, 96, 169, 82, 162, 5, 32, 39, 192
130 DATA 169, 90, 162, 0, 32, 39, 192, 238, 25,
208, 76, 49, 234, 160, 1, 141, 18, 208, 142, 17
140 DATA 208, 140, 25, 208, 140, 26, 208, 88, 96,
205, 18, 208, 208, 251, 142, 32, 208, 142, 33
150 DATA 208, 96, 169, 50, 162, 14, 32, 87, 192,
169, 160, 162, 5, 32, 87, 192, 169, 251, 162, 0
160 DATA 32, 87, 192, 238, 25, 208, 76, 49, 234
```

(Jan Stodůlka)

Výběr z menu

V posledním čísle FUN jsem četl v rubrice Tipy a triky o programovém ošetření výběru z menu. Chtěl bych uvést několik dalších způsobů volby z menu. Nejjednodušším je volba pomocí GET a rozhodovacího příkazu IF...THEN, např.

```
100 GET WS: IF WS="" THEN 100
```

Po vytištění menu na obrazovce program zůstane stát na uvedeném řádku a čeká na stisk volby:

```
110 IF WS="1" THEN 1000
120 IF WS="K" THEN END
130 GOTO 100
```

Při chybné volbě se nic nestane a program čeká dál na stisk správné klávesy. Dalším způsobem může být použití přepínače ON X GOTO:

```
110 IF WS="1" THEN X=1
120 IF WS="R" THEN X=2
130 ON X GOTO 1000,1100
```

Pokud potřebujete z menu volit podle kombinace znaků, lze požit např. tento způsob:

```
110 IF WS="W" THEN Q1=1
120 IF WS="T" THEN Q1=2
130 IF WS="1" THEN Q2=1
140 IF WS="2" THEN Q2=2
150 IF Q1=1 AND Q2=1 THEN 1000
160 IF Q1=1 AND Q2=2 THEN LOAD"LR",8
170 IF Q1=2 AND Q2=1 THEN END
```

Rozhodneme-li se volit z menu pomocí funkčních kláves, můžeme ke skokům do podprogramů použít přepínač ve tvaru:

```
200 ON ASC(AS)-132 GOTO 1000,1100,1200...
```

Rozhodovací část programu k volbě z menu bude pak např.

```
110 IF AS=CHR$(133) THEN 1000
120 IF AS=CHR$(137) THEN 1100
```

Při sestavování této části programu je třeba mít na zřetel, že CHR\$ kódy sudých funkčních kláves jsou číselně až za kódy lichých kláves.

(Čestmír Florián)

Vypnutie opakovania kláves

Automatické opakovanie je štandardne na klávesách SPACE, CRSR(UP/DOWN),CRSR(LEFT/RIGHT),INST/DEL. Niekedy je dobré vypnúť opakovanie týchto kláves. Dosiahneme to zadaním príkazu POKE 650,127. Môžeme však dosiahnuť aj opačný efekt t.j. zapnutie opakovania u všetkých kláves príkazom POKE 650,255. Do pôvodného stavu dáme počítač zadaním príkazu POKE 650,0.

(Peter Koreň)

DYCP GRAFIKA

Ve druhém čísle FUNu jsme začali s uveřejňováním různých grafických kouzel, která Commodore 64 dokáže a která jsou běžnému programátorovi prakticky nedostupná. FLI grafika je jedním z desítek triků, kterými se dá z C64 vyždímat něco neobvyklého. Záměrem této rubriky je vyburcovat naše špičkové programátory k tomu, aby se o své poznatky v této oblasti podělili s ostatními uživateli C64 a pomohli tak pěstovat novou generaci programátorů s velkým P. A až se nám v této rubrice nashromáždí programovací techniky větší počet, určitě se objeví i ve vašich programech.

MIK/FUTURE DESIGN PRESENTS: LONG DYCP!!

Obr. DYCP - Skrolování znaků ve formě sinusoidy

Proto v dnešní části speciálních technik programování grafiky věnujeme pozornost velmi zajímavé technice DYCP.

Zkratka DYCP znamená Different Yielding Character Position a je jednou z technik, nad kterou i ze zkušeného programátora vyjde povzdech. Tato technika umožňuje skrolování znaků po obrazovce ve formě sinusoidy.

Technika DYCP pracuje jen se znaky. Je to proto, že při skrolování ve směru osy Y lze pracovat jen s jednou hodnotou a to pro každý znak jednoho řádku. Při technice DYCP se znaky pohybují vertikálně různou rychlostí a zcela nezávisle. S pomocí skrolovacího registru SDO11 však toto nelze zajistit.

Použitý trik je jednoduchý, účinný a funguje s jednou znakovou sadou. Přesněji řečeno je znakový soubor animován a to v osmi stupních pro každý znak. V demoprogramu je znaková sada obsažena, proto můžete ihned po spuštění programu příkazem SYS 4096 efekt vychutnat. Ale vraťme se k popisu funkce.

| | | | | | | | | | | |
|------|---|----|----|----|----|----|----|----|----|----|
| 11E0 | : | 08 | 06 | 04 | 02 | 02 | 01 | 01 | 01 | 5B |
| 11E8 | : | 01 | 01 | 01 | 01 | 01 | 01 | 02 | 02 | EF |
| 11F0 | : | 03 | 04 | 06 | 08 | 09 | 0A | 0C | 0D | A3 |
| 11F8 | : | 0F | 10 | 12 | 14 | A0 | 00 | 8C | EA | 29 |
| 1200 | : | 9A | A2 | 00 | 98 | 9D | 00 | 04 | AD | 44 |
| 1208 | : | EA | 9A | C9 | A0 | D0 | 03 | 4C | 00 | 1C |
| 1210 | : | 9B | 18 | 69 | 28 | 8D | EA | 9A | C8 | 43 |
| 1218 | : | 4C | E8 | 9A | A9 | 00 | 8D | EA | 9A | 01 |
| 1220 | : | E8 | C8 | E0 | 28 | D0 | DD | 60 | A5 | 72 |
| 1228 | : | 01 | 29 | FB | 85 | 01 | A9 | D6 | 85 | 31 |
| 1230 | : | FB | A9 | 00 | 85 | FA | A9 | 02 | 85 | C1 |

PROGRAMM : DYCP.OBJ

1000 12B3

| | | | | | | | | | | |
|------|---|----|----|----|----|----|----|----|----|----|
| 1000 | : | A2 | F4 | BD | 27 | 10 | 9D | 00 | 90 | 80 |
| 1008 | : | CA | E0 | FF | D0 | F5 | E8 | BD | 1B | 30 |
| 1010 | : | 11 | 9D | 00 | 9A | E8 | D0 | F7 | A2 | 7D |
| 1018 | : | 98 | BD | 1B | 12 | 9D | 00 | 9B | CA | 76 |
| 1020 | : | E0 | FF | D0 | F5 | 4C | 00 | 90 | A9 | 4D |
| 1028 | : | 38 | 8D | 0B | 90 | A9 | 00 | A2 | 00 | 21 |
| 1030 | : | 9D | 00 | 38 | E8 | D0 | FA | AD | 0B | AA |
| 1038 | : | 90 | C9 | 41 | F0 | 06 | EE | 0B | 90 | 40 |
| 1040 | : | 4C | 05 | 90 | 20 | 44 | E5 | 20 | E1 | EF |
| 1048 | : | 9A | 78 | 20 | 0C | 9B | A2 | 01 | 8E | 98 |
| 1050 | : | 1A | D0 | CA | 8E | 0E | DC | 8E | 23 | 9F |
| 1058 | : | 9B | A9 | 1B | 8D | 11 | D0 | A2 | 4B | F9 |
| 1060 | : | A0 | 90 | 8E | 14 | 03 | 8C | 15 | 03 | 5D |
| 1068 | : | A9 | 00 | 85 | D3 | A9 | 04 | 85 | D6 | 6C |
| 1070 | : | 58 | 60 | AE | F4 | 90 | CA | E0 | FF | 25 |
| 1078 | : | D0 | 02 | A2 | 3F | 8E | F4 | 90 | 2E | 09 |
| 1080 | : | 19 | D0 | A9 | 1E | 8D | 18 | D0 | A2 | 52 |
| 1088 | : | 72 | A0 | 90 | 8E | 14 | 03 | 8C | 15 | F6 |
| 1090 | : | 03 | A9 | 5A | 8D | 12 | D0 | 4C | BC | 02 |
| 1098 | : | FE | 2E | 19 | D0 | 2C | 19 | D0 | A9 | 30 |
| 10A0 | : | 15 | 8D | 18 | D0 | A2 | 4B | A0 | 90 | C4 |
| 10A8 | : | 8E | 14 | 03 | 8C | 15 | 03 | A9 | 30 | 03 |
| 10B0 | : | 8D | 12 | D0 | 20 | 92 | 90 | 4C | 31 | C0 |
| 10B8 | : | EA | A9 | 40 | 8D | AE | 90 | A9 | 38 | BF |
| 10C0 | : | 8D | B1 | 90 | A9 | 00 | 8D | AD | 90 | C3 |
| 10C8 | : | 8D | B0 | 90 | AE | F4 | 90 | BC | 00 | 6E |
| 10D0 | : | 9A | A2 | 00 | BD | 00 | 40 | 99 | 00 | DC |
| 10D8 | : | 38 | E8 | C8 | E0 | 0C | D0 | F4 | AD | 49 |
| 10E0 | : | AD | 90 | 18 | 69 | 0A | 8D | AD | 90 | ED |
| 10E8 | : | 90 | 03 | EE | AE | 90 | AD | B0 | 90 | E6 |
| 10F0 | : | 18 | 69 | 28 | 8D | B0 | 90 | 90 | 03 | 50 |
| 10F8 | : | EE | B1 | 90 | AD | A8 | 90 | 18 | 69 | DB |
| 1100 | : | 04 | 8D | A8 | 90 | 90 | 03 | EE | A9 | 37 |
| 1108 | : | 90 | AD | A8 | 90 | C9 | B4 | D0 | BB | A8 |
| 1110 | : | A2 | 00 | A0 | 9A | 8E | A8 | 90 | 8C | B7 |
| 1118 | : | A9 | 90 | 60 | 10 | 0E | 0C | 0A | 09 | 9F |
| 1120 | : | 08 | 06 | 05 | 04 | 03 | 02 | 02 | 01 | 38 |
| 1128 | : | 01 | 01 | 00 | 01 | 01 | 01 | 02 | 02 | EE |
| 1130 | : | 03 | 04 | 05 | 06 | 08 | 09 | 0A | 0C | 41 |
| 1138 | : | 0D | 0F | 10 | 12 | 14 | 15 | 17 | 18 | 8A |
| 1140 | : | 19 | 1B | 1C | 1C | 1D | 1D | 1E | 1E | E1 |
| 1148 | : | 1E | 1E | 1E | 1E | 1E | 1E | 1E | 1D | 46 |
| 1150 | : | 1D | 1C | 1C | 1B | 1A | 18 | 17 | 15 | CF |
| 1158 | : | 14 | 12 | 11 | 10 | 0E | 0C | 0A | 09 | 37 |
| 1160 | : | 08 | 06 | 05 | 04 | 03 | 02 | 02 | 01 | 78 |
| 1168 | : | 01 | 01 | 01 | 01 | 01 | 01 | 02 | 02 | 6F |
| 1170 | : | 03 | 04 | 05 | 06 | 08 | 09 | 0A | 0C | 81 |
| 1178 | : | 0D | 0F | 10 | 12 | 14 | 15 | 17 | 18 | CA |
| 1180 | : | 19 | 1B | 1C | 1C | 1D | 1D | 1E | 1E | 21 |
| 1188 | : | 1E | 1E | 1E | 1E | 1E | 1E | 1E | 1D | 86 |
| 1190 | : | 1D | 1C | 1C | 1B | 1A | 18 | 17 | 15 | 0F |
| 1198 | : | 14 | 12 | 11 | 10 | 0E | 0C | 0A | 09 | 77 |
| 11A0 | : | 08 | 06 | 05 | 04 | 03 | 02 | 02 | 01 | B8 |
| 11A8 | : | 01 | 01 | 01 | 01 | 01 | 01 | 02 | 02 | AF |
| 11B0 | : | 03 | 04 | 05 | 06 | 08 | 09 | 0A | 0C | C1 |
| 11B8 | : | 0D | 0F | 10 | 12 | 14 | 15 | 17 | 18 | 0A |
| 11C0 | : | 19 | 1B | 1C | 1C | 1D | 1D | 1E | 1E | 61 |
| 11C8 | : | 1E | 1E | 1E | 1E | 1E | 1E | 1E | 1D | C6 |
| 11D0 | : | 1D | 1C | 1C | 1B | 1A | 18 | 17 | 15 | 4F |
| 11D8 | : | 14 | 12 | 10 | 0F | 0D | 0C | 0A | 09 | 47 |

```

1240 : 6F 9B 98 18 69 40 A8 C0 63
1248 : 80 D0 07 A5 01 09 04 85 1A
1250 : 01 60 A5 FA 18 69 0E B0 98
1258 : 27 85 FA 88 D0 F4 A0 00 49
1260 : B1 FA 91 FC C8 C0 08 D0 E7
1268 : F7 EE 23 9B A5 FC 18 69 88
1270 : 0A 85 FC B0 10 A9 00 85 EB
1278 : FA A9 D6 85 FB 4C 22 9B 8F
1280 : E6 FB 4C 3E 9B E6 FD 4C C0
1288 : 5A 9B 4D 49 4B 2F 46 55 1E
1290 : 54 55 52 45 20 44 45 53 AC
1298 : 49 47 4E 20 50 52 45 53 70
12A0 : 45 4E 54 53 3A 20 20 4C 4A
12A8 : 4F 4E 47 20 44 59 43 50 B1
12B0 : 21 21 40 FF FF FF FF FF 72

```

Obr. Výpis programu DYCP.OBJ v MSE 1.1

Obdobně jako při vodorovném skrolování musíme znak nejdříve 8x po jednotlivých pixelech posunout ve směru Y, než pomocí hardscrollingu posuneme znak o celých 8 pixelů. Toho dosáhneme řadou animačních kroků u znaku. Když jsme takto pomocí animace znak

8x posunuli o pixel, můžeme sáhnout k hardscrollingu a posunout znak o celých 8 pixelů nahoru nebo dolů v obrazovkové masce. Musíme přitom jen sledovat různé odstupující jednotlivých znaků mezi sebou. K tomu slouží tabulka MAP ve zdrojovém kódu, která zajišťuje dynamickou vzdálenost znaků. Když znak dosáhne horní pozice, pokračuje v pohybu opačným směrem až po dosažení dolní pozice, odkud jeho pohyb pokračuje opět nahoru a tak stále dokola. Že to všechno musí probíhat pěkně svižně, je samozřejmostí.

Pro programování DYCP je výhodnější nepoužívat programové smyčky. Zníto jako nesmysl, když jen kód pro tento efekt pak spolkně cca 4kB (i se znakovou sadou), dává to ale smysl, když pamatujeme také na to, kolik rastrového času nám zbývá na hudbu a ostatní efekty. Pro skalní příznivce uveřejníme v příštím čísle komentovaný výpis v assembleru a nakonec se vrátíme k technice DYSP, která tuto techniku předchází. Pro zajímavost, DYSP znamená Different Yielding Sprite Position, týká se tedy obdobné problematiky u sprajtlů.

(JK)

RS232

Mnoho komerčních a dnes už i domácích přístrojů používá k přenosu dat interfejs RS232. Co se za tímto konstatováním skrývá, vysvětlím dále.

Jistě není třeba připomínat, že user-port C64 se dá využívat jako rozhraní RS232. RS232 vystupuje jako protipól k paralelní normě Centronics. Základní ideou sériového způsobu přenosu dat je posílání nikoli celých bajtů dat, ale jednotlivých bitů. Aby si vůbec vysílač s přijímačem rozuměl, má přenos dat přesně stanovenou rychlost. Přenosová rychlost se měří v Baudech. Jeden Baud představuje rychlost, kdy se přenesou jeden bit za sekundu. Obvyklé rychlosti přenosu udává řada: 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200 bit/s

Aby bylo možno vysílač a přijímač přesně synchronizovat, přenášejí se v praxi pakety dat tvořené 5 až 8 datovými bity. Navíc bývají orámovány startbitem a jedním až dvěma stopbity. Startbit mívá zásadně úroveň L a stopbit H. Tím vznikne při přenosu mezi dvěma bajty vždy sestupná hrana, z níž vhodný přípravek v přijímači vytvoří zpětně takt přenosu. Tímto způsobem je možno data přenášet dvoužilovým kabelem. Pro minimalizaci přenosových chyb se dává před stopbit ještě tzv. paritní bit. Zda bit bude mít hodnotu log. l. či H, záleží na tom, jedná-li se o sudou nebo lichou paritu. Je-li například v 8 bitovém přenosu 5 bitů úroveň H, nastaví se při sudé paritě vysílačem ještě paritní bit. V přijímači se přezkouší stav bitů. Jednotlivé chyby mohou být rozpoznány a zobrazeny.

Aby byla zvýšena odolnost proti rušení, což je důležité zvláště u přenosů na větší vzdálenosti, byla u RS232

zvolena jiná úroveň signálů pro stav log. L a H než u logiky TTL. U delších vedení může samotný ohmický odpor vodičů způsobit takový útlum, že vyslaná úroveň H nedosáhne na přijímací straně nikdy 5V, ale stěží hodnoty 2V, což může přijímač vyhodnotit jako úroveň L.

Z těchto důvodů pracuje norma RS232 s úrovní napětí -12V až -3V pro log. H a +3V až +12V pro log. L. Protože se jedná o maximální potenciální rozdíl 24V, byla norma RS232 zpracována do normy V.24 (DIN 66020). Sama definice, které napětí je pro log. H a které pro log. L je rozdílná, což může vést k některým omylům. Zde nezbyvá než řešit každé připojení samostatně. Mnohdy pomůže pro přizpůsobení pouhé vložení invertoru.

Přenosový protokol

V předchozím odstavci bylo naznačeno, že přenos dat je možný dvojvodičovým vedením. Označení vodičů na vysílací straně je obecně TXD (Transmit-Data = vysílaná data); zbylé nevyužité vodiče kabelu slouží k propojení zemí.

Mají-li být zároveň posílána i data zpět, je k tomu zapotřebí třetího vodiče. Obvykle bývá označen RXD (Receive Data = přijímaná data). Tím mohou být ovládány i přístroje, které nepracují tak rychle jako vysílač. Určitými znaky na tomto "pětnovazebním" vedení se dá vysílač přibrzdit či zastavit. Jedná se o softwarový protokol mezi vysílačem a přijímačem. Nejběžnějším je protokol XON-XOFF. Používají se při něm oba řídicí znaky v ASCII, DC3 (XOFF=\$13=CTRL-S) a DC1 (XON=\$11=CTRL-Q). Pokud je přijímač zaměstnán ji-

nou úlohou, zastaví signálem XOFF vysílání protější stanice. Úlohou se rozumí tisk dat nebo jejich ukládání do paměti. Signál XON je vyhrazen pro hlášení připravenosti k dalšímu přijímání znaků. Vysílač i přijímač musí být ovšem schopen současně vysílat i přijímat. Úloha se dá řešit i jinak, jak ukazuje procedura EXT-ACK. Počítač vysílá celý blok dat, který je uzavřen signálem EXT (End Of Text). Jakmile je celý text přijímačem zpracován, vyšle se zpět řídicí znak ACK (Acknowledge) - potvrzení, z něhož počítač pozná, že má poslat následující blok dat. Procedura EXT-ACK předpokládá minimálně na straně přijímače použití vyrovnávací paměti pro zachycení celého bloku dat.

Pokud to hardware dovolí, realizuje se vzájemné potvrzování (handshake) ještě jednodušejší signálem DTR (Data Terminal Ready) - přijímač připraven. Pokud nemůže přijímač převzít další data, aktivuje toto vedení. Vysílač potom čeká tak dlouho, dokud tento signál opět není zrušen.

Zvláště při přenosech s vyššími přenosovými rychlostmi, po dlouhých vedeních, v prostředí se silným rušením, může docházet k těžkostem při určování správného taktu ze seriových dat. Mnohé přístroje ani nemají taktovací generátor, proto není ani účelné převádět synchronizační impulsy zvláštním vodičem. Provede se spojení vodičů TC (Transmit Clock = vysílací takt) a RC (Receiver Signal Clock).

RS232 počítače C64

Na začátku článku jsem se zmínil o tom, že u C64 se nepoužívá pro sériový přenos dat pomocí C64 sériový port CIA, ale user port. Jednotlivé piny user portu přitom dostanou následující význam:

| Pin | Signál | Název | EIA | zkratka | směr |
|-----|--------|----------------------|-----|---------|------|
| A | GND | Protective Ground | AA | GND | - |
| B | FLAG2 | Received Data | BR | Sin | in |
| C | PB0 | Received Data | BB | Sin | in |
| D | PB1 | Request To Send | CA | RTS | out |
| E | PB2 | Data Terminal Ready | CD | DTR | out |
| F | PB3 | Ring Indicator | CE | RI | in |
| H | PB4 | Received Line Signal | CF | DCD | in |
| J | PB5 | in | | | |
| K | PB6 | Clear To Send | CB | CTS | in |
| L | PB7 | Data Set Ready | CC | DSR | in |
| M | PA2 | Transmitted Data | BA | Sout | out |
| N | GND | Signal Ground | AA | GND | - |

C64 řídí rozhraní RS232 jako periferní přístroj s číslem 2. Pro určení parametrů přenosu jsou k dispozici dva registry, které je možno ovládat z basicu. Připojená tabulka ukazuje funkce jednotlivých bitů těchto registrů:

Kontrolní registr

| bit | obsah | hodnota | funkce | |
|---------|-------|---------|-----------------------|-----------------|
| 3,2,1,0 | 0000 | 0 | samodefinovaný přenos | |
| | 0001 | 1 | 50 bit/s | |
| | 0010 | 2 | 75 bit/s | |
| | 0011 | 3 | 110 bit/s | |
| | 0100 | 4 | 134,5 bit/s | |
| | 0101 | 5 | 150 bit/s | |
| | 0110 | 6 | 300 bit/s | |
| | 0111 | 7 | 600 bit/s | |
| | 1000 | 8 | 1200 bit/s | |
| | 1001 | 9 | 1800 bit/s | |
| 4 | 1010 | 10 | 2400 bit/s | |
| | X | 0 | nevyužívá se | |
| | 6,5 | 00 | 0 | 8 datových bitů |
| | | 01 | 32 | 7 datových bitů |
| 10 | | 64 | 6 datových bitů | |
| 7 | 11 | 96 | 5 datových bitů | |
| | 0 | 0 | 1 stopbit | |
| | 1 | 128 | 2 stopbity | |

| bit | obsah | hodnota | funkce |
|-------|-------|---------|----------------------------|
| 0 | 0 | 0 | 3-Line (softwarový prot.) |
| | 1 | 1 | X-Line (hardwarový prot.) |
| 3,2,1 | XXX | 0 | nevyužívá se |
| 4 | 0 | 0 | duplex |
| | 1 | 16 | poloviční duplex |
| 7,6,5 | 000 | 0 | bez zkoušky parity |
| | 001 | 32 | lichá parita |
| | 010 | 64 | bez zkoušky parity |
| | 011 | 96 | sudá parita |
| | 100 | 128 | bez zkoušky parity |
| | 101 | 160 | 8.datový bit=1, bez parity |
| | 110 | 192 | bez zkoušky parity |
| | 111 | 224 | 8.datový bit=0, bez parity |

Pomocí tabulky můžete určit hodnoty, které zapsat do obou registrů, aby fungoval přenosový protokol. Aktivace RS232 se děje pomocí posloupnosti příkazů: OPEN file,2,0,CHRS(kontrolní reg.) + CHRS(řídící registr)

Pro objasnění uvedu příklad. Sečtěte jednotlivé hodnoty z tabulky:

OPEN 1,2,0,CHRS(6+64+0) + CHRS(1+0+32)

Znamená to, že pro datový soubor 1 byl zvolen duplexní přenos s 1 stopbitem, 6 datovými bity, rychlostí přenosu 300 Baudů, lichou paritou a hardwarovým protokolem.

Pokud se zvolí číslo datového souboru větší než 128, vyšle po každém CR (Carriage Return) počítač Line-feed.

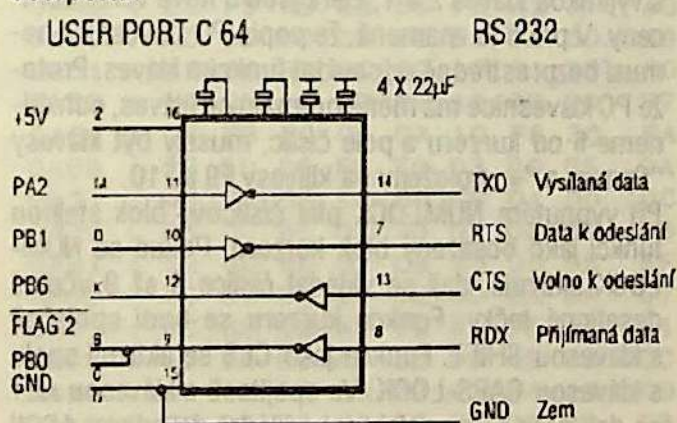
Systém C64 umožňuje i analýzu chyb RS232. Přímou z basicu se dá zkontrolovat status ev. se dá číst adresa \$0297 (dekadicky 663).

Přizpůsobení úrovní

Bohužel C64 má na user portu pouze úroveň TTL, takže v tomto bodě nesplňuje požadavky normy pro provoz RS232. Není však třeba zoufat. Dnes již existují integrované obvody, které ošetří všechny signály podle požadavků normy. Takovým obvodem je MAX 232, který vystačí s napájecím napětím +5V a potřebná symetrická napájení +12V a -12V si vyrobíš sám. Obvod obsahuje napěťový měnič, který připojením externích elektro-

lytických kondenzátorů vyrábí pomocná napětí. Dále jsou v něm čtyři invertory. Dva z nich převádějí signály úrovně TTL na RS232, zatímco zbylé dva jsou plánovány pro obrácený převod.

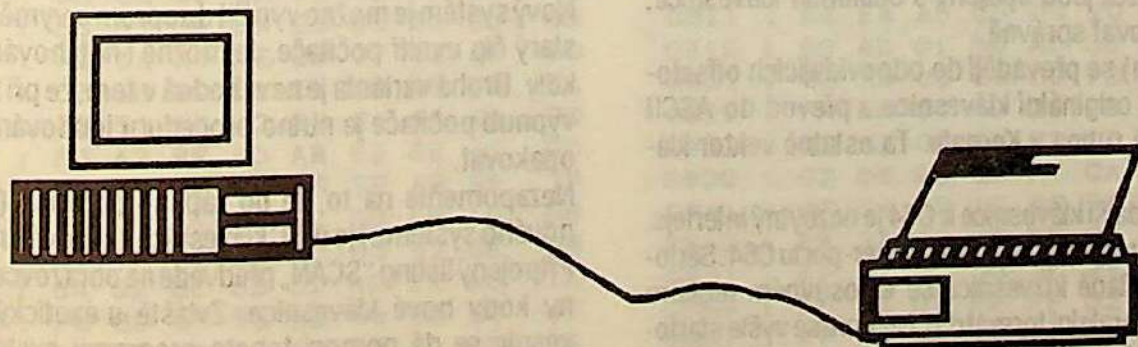
Zapojení obvodu MAX 232 k user portu C64 uvádí obrázek. Obvod se dá zabudovat do krytky konektoru userportu.



Obr. RS232 pomocí obvodu MAX232.

Z obrázku je patrné, že přenos dat je doplněn vzájemným potvrzováním připravenosti periferního zařízení a počítače. Tímto způsobem lze k C64 připojit tiskárny s příslušným interfejsem, dálnopisy a modemy.

(JV)



Zažili jste už úplně ztuhlé prsty a bolesti v zápěstí od několikahodinového fukání na klávesnici C64? Určitě alespoň někteří vědí, co mám na mysli.

Klávesy C64 jsou mimoriádně špatně ergonomicky uspořádány. Pochválit se nedá ani to, že leží vysoko nad podložkou, nemají definovaný zdvih a v sepnutém stavu mají vysoký odpor.

Uživatel C64 může proto jen mlsně pokukovat i po těch nejlevnějších klávesnicích k PC. Bohužel ty se však přímo na C64 provozovat nedají.

Proti klávesnicím C64, které jsou pasivně připojeny k počítači jsou PC klávesnice inteligentní. Vlastní procesor v nich přebírá úlohu osahávání klávesnice a odesílá aktuální hodnoty sériově do připojeného počítače. Pro skutečně účinnou spolupráci PC klávesnice s C64 jsou nutné tedy jak hardwarové tak i softwarové předpoklady.

Nejnápadnější rozdíl mezi velkou (PC) klávesnicí a commodoráckou je ve větším počtu kláves. Nová klávesnice pracuje analogicky jako klávesnice C64, s výjimkou kláves Z a Y, které jsou u nové verze obráceny. V praxi to znamená, že popis PC klávesnice nemusí bezprostředně odpovídat funkcím kláves. Protože PC klávesnice má méně normálních kláves, odhlédneme-li od kurzoru a pole číslic, musejí být klávesy "Space", a "=", položeny na klávesy F9 a F10.

Při vypnutém NUMLOCK plní číslicový blok stejnou funkci jako obsazený blok kurzoru. Pokud se NUMLOCK aktivuje, dají se vkládat číslice 0 až 9 včetně desetinné tečky. Funkce kurzoru se budí společně s klávesou SHIFT. Funkce jako CLS se aktivují spolu s klávesou CAPS-LOCK. Ve spojitosti s klávesou ALT se dají přes numerický blok vkládat dekadicky ASCII znaky. Stiskne se klávesa ALT a vloží se hodnota ASCII. Po uvolnění klávesy ALT se přenesou znak do vyrovnávací paměti klávesnice a tam se vyhodnotí jako každý normální stisk klávesnice.

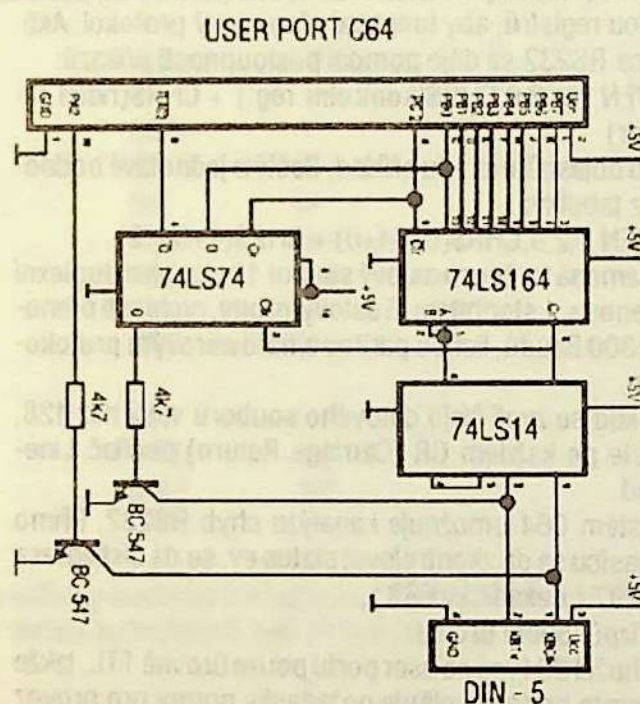
Principiálně existují dvě možnosti připojení:

1. Kódy (scan) vysílané procesorem klávesnice se převádějí do odpovídajících ASCII znaků (podle obsazení klávesnice) a zapisují se do vyrovnávací paměti klávesnice. Tato metoda je komfortní, pokud člověk není limitován originální klávesnicí. Nevýhodou je to, že programy, které jsou spojeny s obsluhou klávesnice, nemusí fungovat správně.

2. Kódy (scan) se převádějí do odpovídajících offsetových hodnot originální klávesnice a převod do ASCII znaků přebírá rutina v Kernalu. Ta ostatně vektor klávesnice volá.

Pro připojení další klávesnice k C64 je nezbytný interfejs. Klávesnice se připojí pomocí nějak user-portu C64. Sériové znaky, vysílané klávesnicí, se v posuvném registru převádějí na paralelní formát dat. Nejprve se vyšle startovací bit, následuje 8 datových bitů a nakonec přijde bit 0.

Ke zpracování startovacího bitu slouží IO 74LS74. Svým výstupem Q zavírá přes tranzistor datový vodič klávesnice. To je nutné zejména proto, že C64 nestíhá přicházející kódy tak rychle zpracovat a tím by mohlo dojít ke ztrátě některých z nich. Čte-li nyní C64 data z portu B, vyrobí CIA na výstupu PC2 impuls (L). Tento vývod je spojen s resetovacími vstupy posuvného registru (74LS164) a klopného obvodu (74LS74). Posuvný registr může nyní přijmout další proud dat a klopný obvod uvolní přes spínací tranzistor klávesnici. Invertor 74LS14 slouží jako tvarovač impulsů a jako invertor tvarovacího signálu. PA2 může uvolnit reset procesoru klávesnice. K tomu účelu se taktovací vedení stahuje na 20 ms přes druhý spínací tranzistor do úrovně L.



Obr. Připojení PC klávesnice k user portu C64.

Opište pomocí MSE1.1 připojený listing a uložte jej před startem vložte do drazvu naformátovanou prázdnou disketu.

Program přečte Kernalu, odstraní z něj rutinu RS232 a přepíše na její místo novou rutinu pro klávesnici. Současně zapíše na disketu změněný Kernalu. Potom okopíruje do paměti pod ROM nový systém.

Nový systém je možno vypálit do eprom a vyměnit jej za starý čip uvnitř počítače. Je možné i natahování z diskety. Druhá varianta je nevhodná v tom, že při každém vypnutí počítače je nutno proceduru loudování znovu opakovat.

Nezapomeňte na to, že po zapnutí počítače (aktivaci nového systému) je u PC klávesnice aktivován mód XT. Připojený listing "SCAN", předvede na obrazovce všechny kódy nové klávesnice. Zvláště u exotických klávesnic se dá pomocí tohoto programu systém lépe přizpůsobit.

(JV)

PROGRAMM : SCAN

C000 C071

C000 : A9 00 8D 03 DD AD 00 DD 74
 C008 : 09 04 8D 00 DD AD 01 DD 82
 C010 : A2 03 A0 00 88 D0 FD CA F9
 C018 : D0 FA AD 00 DD 29 FB 8D 03
 C020 : 00 DD AD 0D DD 29 10 F0 65
 C028 : F9 AD 01 DD C9 46 F0 16 B3
 C030 : C9 81 D0 04 A9 93 D0 08 F9
 C038 : 8D 70 C0 20 4E C0 A9 20 03
 C040 : 20 D2 FF 4C 22 C0 AE 70 16
 C048 : C0 E0 E0 D0 EB 60 48 4A 42
 C050 : 4A 4A 4A 20 59 C0 68 29 E6
 C058 : 0F AA BD 60 C0 4C D2 FF F1
 C060 : 30 31 32 33 34 35 36 37 50
 C068 : 38 39 41 42 43 44 45 46 CD
 C070 : E0 67 4A B7 4B 48 E1 D2 B1

Obr. Výpis rutiny SCAN (MSE 1.1)

PROGRAMM : KERNAL-GEN

0B01 0CF8

0B01 : 0B 08 C8 07 9E 32 30 36 CC
 0B09 : 34 00 00 00 FF 00 FF A9 91
 0B11 : 00 A0 E0 85 5F 84 60 85 F1
 0B19 : 5A 85 5B A0 60 85 58 84 BD
 0B21 : 59 20 BF A3 AD 9D 44 C9 5B
 0B29 : 4B F0 0D A2 90 8E 3C 08 02
 0B31 : E8 8E 41 08 E8 8E 50 08 06
 0B39 : A2 00 BD 60 08 85 02 BD 87
 0B41 : 61 08 F0 15 85 03 A0 00 78
 0B49 : A9 EA 91 02 C8 98 DD 62 99
 0B51 : 08 D0 F5 E8 E8 E8 D0 E2 3B
 0B59 : EA 4C D0 08 EA EA EA 0E 6C
 0B61 : 4D 03 BB 4E 00 BB 4F 00 04
 0B69 : BB 50 02 51 51 04 B8 51 B2
 0B71 : 12 08 52 06 AD 52 1B 09 C9
 0B79 : 54 95 8A 58 03 0A 5E 0E 5C
 0B81 : 72 5E 4A C2 5E 86 00 00 27
 0B89 : 00 00 00 00 00 00 00 0E A6
 0B91 : 4D 03 BB 4E 00 BB 4F 00 34
 0B99 : BB 50 02 4A 51 0B 76 51 30
 0BA1 : 02 B8 51 12 E5 51 02 08 97
 0BA9 : 52 06 25 52 05 66 52 04 67
 0BB1 : A9 52 1F 84 53 07 09 54 16
 0BB9 : 95 0A 5E 0E 54 5E 02 72 D2
 0BC1 : 5E 4A C2 5E 6B 00 00 00 77
 0BC9 : 00 00 00 00 00 00 00 AD 25
 0BD1 : 9D 44 C9 4B D0 58 A2 13 ED
 0BD9 : A0 F7 A9 4C 8D 4E 51 8E 16
 0BE1 : 4F 51 8C 50 51 8D 27 52 C9
 0BE9 : 8E 28 52 8C 29 52 8D 6C E6
 0BF1 : 52 8E 6D 52 8C 6E 52 8D D1
 0BF9 : 88 53 8E 89 53 8C 8A 53 6A
 0901 : A2 1A A0 FE 8D 07 5E 8E 60
 0909 : 08 5E 8C 09 SE A9 D5 8D 2A
 0911 : 78 51 A9 42 8D E4 51 A9 7D
 0919 : F0 A2 BF 8D AB 52 8E AC DD
 0921 : 52 A9 66 8D 55 5E A9 56 2F
 0929 : 8D 65 5E 4C 6C 09 A9 4C D8
 0931 : A2 0D A0 F7 8D 63 52 8E DB
 0939 : 64 52 8C 65 52 A2 1A A0 7A
 0941 : FE 8D 07 5E 8E 08 5E 8C 4F
 0949 : 09 5E A9 90 A2 10 8D 23 25
 0951 : 52 8E 24 52 A2 64 8D A7 11
 0959 : 52 8E A8 52 A9 24 8D 41 DB
 0961 : 51 A9 0C 8D 7E 53 A9 56 11

0969 : 8D 65 5E A2 00 BD D0 09 D8
 0971 : 85 5F BD D1 09 F0 20 85 F3
 0979 : 60 BD D2 09 85 5A BD D3 57
 0981 : 09 85 5B BD D4 09 85 58 38
 0989 : BD D5 09 85 59 8A 18 69 41
 0991 : 06 48 20 BF A3 68 AA D0 85
 0999 : D4 A9 01 A2 08 A8 20 BA 92
 09A1 : FF A9 0B A2 C2 A0 09 20 22
 09A9 : BD FF A2 00 A0 40 86 02 39
 09B1 : 84 03 A0 60 A9 02 20 D8 C8
 09B9 : FF A9 35 85 01 4C F2 0C E1
 09C1 : EA 4D 46 49 49 2D 4B 45 C2
 09C9 : 52 4E 41 4C EA EA EA 20 0E
 09D1 : 0A 3B 0A 96 44 3B 0A E4 DE
 09D9 : 0A 30 4B E4 0A E9 0A 54 2B
 09E1 : 4B E9 0A 67 0B 39 4F 67 17
 09E9 : 0B F9 0B 9B 54 F9 0B 1A 9D
 09F1 : 0C DD 56 1A 0C 1D 0C F9 93
 09F9 : 5D 1D 0C 2B 0C 18 5E 2B 9F
 0A01 : 0C 73 0C BA 5E 73 0C F2 B9
 0A09 : 0C 41 5F 00 A0 00 C0 00 9B
 0A11 : C0 00 40 00 60 00 00 00 E8
 0A19 : 00 00 00 00 00 00 00 20 5A
 0A21 : 50 43 2D 54 41 53 54 41 6B
 0A29 : 54 55 52 20 43 2D 36 34 9F
 0A31 : 20 42 41 53 49 43 20 56 09
 0A39 : 32 20 A0 40 84 CB A9 81 FC
 0A41 : 85 F5 A9 EB 85 F6 A2 FF 43
 0A49 : AD 0D DD 29 10 D0 01 60 66
 0A51 : A9 00 8D 8C 02 AD 01 DD 3D
 0A59 : 10 03 4C 09 F4 A2 06 DD 57
 0A61 : CF F6 F0 05 CA 10 F8 30 FA
 0A69 : 3B BD D6 F6 E0 03 B0 05 0A
 0A71 : 45 F7 85 F7 60 C9 01 D0 0C
 0A79 : 05 2C 8C 02 30 0F 0D 8D C3
 0A81 : 02 8D 8D 02 C9 03 D0 05 EF
 0A89 : AD 91 02 10 0E 60 EA EA E7
 0A91 : 6C 8F 02 A4 CB B1 F5 AA 51
 0A99 : 4C 2A EB AD 18 D0 49 02 DC
 0AA1 : 8D 18 D0 60 C9 01 F0 13 09
 0AA9 : C9 53 D0 1A 2C 8C 02 30 23
 0AB1 : 15 AE 8D 02 E0 06 D0 0E 5F
 0AB9 : 6C FC FF AD 8D 02 D0 04 8D
 0AC1 : A9 7F 85 91 A9 01 A8 4C 9B
 0AC9 : 72 FE EA EA EA EA EA EA 5A
 0AD1 : EA EA EA EA EA EA EA EA D0
 0AD9 : EA EA EA EA EA AC 8E 02 A2
 0AE1 : 8C 8D 02 EA EA EA EA EA 99
 0AE9 : AD 8D 02 29 02 D0 04 4C 52
 0AF1 : 9F FE EA B9 D5 EE 30 F7 87
 0AF9 : C9 2E F0 F3 38 E9 30 AA 7D
 0B01 : A5 F8 0A 0A 18 65 F8 0A 8B
 0B09 : 85 F8 18 8A 65 F8 85 F8 88
 0B11 : EA EA A5 C5 09 20 85 C5 C6
 0B19 : 60 AD 01 DD AD 0D DD A2 4C
 0B21 : 03 A0 00 88 D0 FD CA D0 4F
 0B29 : FA AD 00 DD 29 FB 8D 00 5E
 0B31 : DD A2 00 8E 8D 02 8E 8E 71
 0B39 : 02 86 F8 86 F7 CA 86 91 A0
 0B41 : AD 01 DD 4C 6E FF EA EA D8
 0B49 : EA 37 38 39 FF 34 35 36 E7
 0B51 : FF 31 32 33 30 2E FF FF 50
 0B59 : FF 33 07 FF FF 02 FF 02 C8
 0B61 : FF FF 07 FF 00 00 C9 E0 0B
 0B69 : F0 27 C9 E1 F0 48 29 7F 90
 0B71 : A2 03 DD D2 F6 F0 05 CA 07
 0B79 : 10 F8 30 31 BD D9 F6 C9 52

```

OC11 : 36 38 01 80 40 04 01 01 DE
OC19 : 02 4C EC EE OF 49 4F 51 75
OC21 : 54 57 58 09 12 04 03 FF A5
OC29 : 84 8F AD 8D 02 8D 8E 02 5D
OC31 : B9 DF FE 85 CB 30 38 C9 FD
OC39 : 40 F0 11 98 20 C2 FE A5 A8
OC41 : F7 29 01 OD 8D 02 8D 8D E9
OC49 : 02 6C 8F 02 2C 8C 02 10 F5
OC51 : 3D B9 E5 EE 4C 7B FE 24 A7
OC59 : F7 10 F6 AD 8D 02 4A 90 FF
OC61 : 07 0A 8D 8D 02 4C 99 FE 69
OC69 : B9 D5 EE AA 4C 30 EB 4C AC
OC71 : 66 F4 A2 06 DD D9 FE F0 45
OC79 : 04 CA 10 F8 60 AD 8D 02 B3
OC81 : 09 01 8D 8D 02 60 4C BB EC
OC89 : EE 3C 3E 40 42 48 4B 52 65
OC91 : 3F 38 3B 08 0B 10 13 18 6A
OC99 : 1B 20 23 28 2B 00 FF 3E C1
OCA1 : 09 0E 11 16 0C 1E 21 26 3B
OCA9 : 29 2E 31 01 3A 0A OD 12 A2
OCB1 : 15 1A 1D 22 25 2A 2D 32 1C
OCB9 : 39 OF 30 19 17 14 1F 1C 70
OCC1 : 27 24 2F 2C 37 34 31 3D A0
OCC9 : 3C 40 04 04 05 06 06 44
OCD1 : 03 03 36 35 FF FF 40 40 OC
OCD9 : 40 2B 40 40 40 28 40 40 8E
OCE1 : 40 40 40 FF FF 36 FF FF 03
OCE9 : FF FF FF FF FF FF FF EB
OCF1 : FF 20 EC EE 6C 00 A0 EA 38

```

```

OB89 : 49 FF 2D 8D 02 8D 8D 02 96
OB91 : 60 8D 8C 02 20 BF F6 C9 8B
OB99 : 46 DO OE A9 7F 85 91 A9 BE
OBA1 : F4 48 A9 33 48 08 6C FA F7
OBA9 : FF A8 4C A5 EA C9 01 DO 67
OBB1 : 04 A9 FF 85 91 60 A2 06 ED
OBB9 : 20 BF F6 CA DO FA C9 E1 9F
OBC1 : F0 F4 60 98 A2 06 DD OA 3C
OBC9 : FE F0 04 CA 10 F8 60 BD 5F
OBD1 : 11 FE AA 4C 30 EB C9 02 23
OBD9 : DO AE 49 FF 2D 8D 02 8D B5
OBE1 : 8D 02 A5 C5 29 20 DO 01 6A
OBE9 : 60 A5 C5 29 DF 85 C5 A6 41
OBF1 : F8 A9 00 85 F8 4C 34 EB 09
OBF9 : 4C 87 EA AD OD DD 29 10 FE
OCO1 : F0 F9 AD 01 DD 60 EA EA DC

```

Obr. Výpis programu KERNAL-GEN (MSE 1.1)

Představujeme Vám . . .

CARTRIDGE MAKER 256KB

Cartridge Maker (dále CM) patří do skupiny periferních přístrojů k C64, ve kterých je možno v pamětech eprom uchovávat programy (uživatelské i hry). Vyznačuje se vysokým komfortem obsluhy, velkou kompatibilitou a rychlostí.

Pomocí CM je možno zpracovat programy tak, aby je bylo možno vypálit do eprom. Z vypálených pamětí, vsunutých do připravených objímek lze potom program vyvolat stiskem klávesy. V porovnání s disketovou jednotkou trvá instalace programu sekundy.

CM obsahuje výkonný modulgenerátor. Jeho úlohou je přestrukturování normálních basicovských či strojových programů. Programy se nejprve přetáhnou do paměti C64. Modulgenerátor je automaticky doplní několika kontrolními daty (název, startovací adresa, délka), autostartem a nakonec vyrobí datové soubory, jejichž obsah je možno přímo vypálit do eprom. Modulgenerátorem mohou být zpracovány téměř všechny jednoduché programy psané v basicu či strojovém kódu. Programy jsou do eprom ukládány úsporně. Do čipu s větší kapacitou se může uložit více programů. Možné

jsou i alternativy, kdy delší program přesahuje do dalšího čipu. I v tomto případě se modulgenerátor postará o správné uložení částí programů do jednotlivých čipů.

Software CM (i modulgenerátor) je umístěn odděleně ve zvláštní eprom.

Ve zkratce:

- 4 patice pro eprom typu 2764 až 27512
- celková kapacita programů uložených v eprom činí 256 KB
- automatické rozeznání typu eprom
- řídicí software umístěn ve zvláštní eprom
- malé vnější rozměry
- komfortní menu, pohodlné zacházení s modulgenerátorem
- vlastní directory vypálených programů seřazeno abecedně
- softwarově odpojitelný

Modulgenerátor

Software se nachází v eprom uložené kolmo k objímčím pro ostatní paměti.

Po zapnutí počítače se objeví na obrazovce menu. Obrazovka je rozdělena na levou a pravou polovinu. V levém okně se objeví hlášení. "Žádné programy ... později budou ve stejném poli abecedně seřazeny názvy programů vypálených do epromek 1 až 4. Pro skok do menu modulgenerátoru uposlechněte nápovědy v pravé části obrazovky a stiskněte " ^ " (šipka vzhůru). Po splnění příkazu se dostanete do menu, které vám umožní zpracovat zvolené programy. Stiskem písmen D, K, H, E, B volíte příslušné funkce.

D - vypíše na obrazovku obsah diskety vložené do floppy

K - otevírá kanál floppy pro odeslání příkazů př.: S: název (pro vymazání souboru název)H - po stisku této klávesy se ocitnete zpět ve hlavním menu

E - přerušuje zpracování programů modulgenerátorem a vrací se zpět do menu modulgenerátoru (aktivní mimo základní úroveň menu).

B - start modulgenerátoru pro zpracování programů. Zvolené basicovské či strojové programy budou přepracovány do datových souborů vhodných pro přímé vypálení do eprom. Před započítím práce s modulgenerátorem doporučujeme připravit všechny programy, které chcete převést do cartridge, na jednu disketu. Druhou, prázdnou, ale naprogramovanou disketu přichystejte pro zaznamenání datových souborů k vypálení do eprom. Tato metoda umožní rychlou a pohodlnou práci s modulgenerátorem.

Po stisku klávesy "B" budete vyzváni ke vložení zdrojové diskety s předem připravenými programy. Na obrazovku se vypíše directory diskety. Označte programy, které chcete převést do eprom (J/N). Současně může být zpracováno až 60 programů. Maximální délka programu činí teoreticky 245 bloků.

V dalším kroku vyberte z nabízených kombinací typů pamětí tu, která se vám jeví jako optimální. Současně s nabídkou pouzder pamětí eprom se zobrazí údaj o tom, jakou paměťovou kapacitu nárokují vybrané programy.

Po uzavření této úlohy začne CM přetahovat programy ze zdrojové diskety do paměti C64 a zpracovávat je. Vložte do dráivu druhou, čistou disketu pro záznam dat. Pokud se najednou nezpracují všechny programy, opakujte postup opětovným vkládáním zdrojové a cílové diskety, podle pokynů programu. Program je připraven i na možnost vzniku chyby při práci s dráivem. Nastane-li tato situace, objeví se dotaz, zda se má učinit nový pokus o čtení/zápis či zda se má zpracovávání přerušit. Název nových datových souborů pro vypálení do eprom je velice případný. Označuje typ paměti př. ".256". U paměti 27512 vzniknou dva datové soubory ".512-1" a ".512-2". Rozdělení je nutné, neboť C64 není schopen mít ve své operační paměti (38 kB free) uložen program dlouhý 64 kB.

Umístění eprom

Vypálené paměti eprom mohou být zasunuty do připravených objímek. Před vkládáním pamětí vypněte počítač. Při vlastním vkládání obvodů do patič dbejte správné orientace. Výřez v čele epromky musí směřovat vlevo, stejným způsobem jsou ostatně orientovány i objímky na desce tištěných spojů. Kterou z připravených objímek osadíte, je ponecháno vlastní libovůli.

Výběr programů z CM

Při zapnutí počítače se objeví hlavní menu. V levé části obrazovky jsou abecedně seřazeny názvy programů. Na konci každého z obsazených řádků je uvedeno v hranaté závorce písmeno. V pravé části obrazovky je nápověda usnadňující práci s modulem.

První řádek v seznamu programů je zobrazen inverzně. Pomocí kurzorových kláves či joystickem lze inverzní řádek přesouvat nahoru a dolů.

Abyste mohli přesunout do paměti počítače a odstartovat zvolený program, máte hned několik možností. Najedte na zvolený program inverzním řádkem a stiskněte return. Pokud se vám tento postup jeví jako zdlouhavý, stiskněte písmeno uvedené v hranaté závorce za názvem programu.

Stiskem klávesy " * " opustíte CM a přejdete do basku. Stisk klávesy "CTRL" + restore vede k resetu systému.

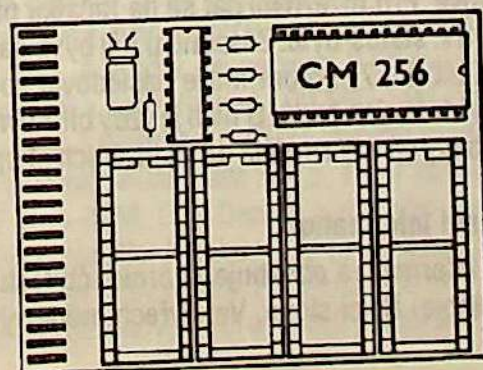
Nastavení I/O

Změna adresy I/O je nutná pouze v případě, že chcete na expanzním portu provozovat vedle CM současně ještě některý jiný modul, který je adresován shodně. Karta je nastavena na I/O 2. Pro nastavení I/O 1 je nutno přerušit propojku mezi dvěma čtvercovými ploškami umístěnými na lícové straně cartridge těsně za konektorem a vodivě propojit střední čtvercovou plošku s ploškou nalevo.

Závěr

Cartridge Maker jsme prakticky vyzkoušeli. Lze jej doporučit všem, kteří často používají určité programy, které není možno pro jejich délku umístit do cartridge s kapacitou 8kB a jejichž neustálé natahování do počítače z kazety či diskety již unavuje. Jistě ji neodmítnou ani ti, kteří si rádi zahrají, neboť ve shodě s předchozím případem odpadá zdlouhavé loudování her. Kladem je i vysoká schopnost modulgenerátoru zpracovat programy.

(JV)



MIDIINTERFEJS PRO C64

Pro profesionály je MIDI pojmem už dost dlouho. Co se však za tímto pojmem skrývá? Jak mohou tuto techniku využít počítačová nadšenci?

MIDI je nejpotřebnější označení pro interfejs k výměně dat mezi hudebními nástroji. Norma pro interfejs MIDI (Musical Instruments Digital Interface) byla zpracována vedoucími výrobci elektronických hudebních nástrojů už na podzim 1983. Byla dána tzv. MIDI specifikací 1.0. Muzikantům byl dán do ruky pomocný nástroj ke smysluplnému kombinování nástrojů rozdílných druhů. MIDIinterfejs řídí přenos dat mezi těmito nástroji. Do řetězu nástrojů se dá připojit i mikropočítač. S pomocí vhodného software (sekvencer, sound-editor) přebírá úlohy řízení. Jednoduchý MIDIinterfejs používá pro připojení 3 pětikolíkové konektory DIN. Podle své funkce jsou nazvány MIDI-In, MIDI-Out a MIDI-Thru. Připojen MIDI-In jsou přijímány vstupní informace. Shodné informace se v nezměněné formě vedou vstupy MIDI-Thru k dalším nástrojům zapojeným do řetězce. Při přenosu vzniká malé časové zpoždění. K vysílání informací slouží výstup MIDI-Out.

Při vzájemném spojení dvou syntetizérů nebo počítačů se propojí do kříže vstup MIDI-In s výstupem MIDI-Out. Toto jednoduché propojení je pro řadu použití pro výměnu dat dostačující. Pokud se chtějí přenášet data z jednoho počítače na řadu dalších nástrojů, propojí výstup MIDI-OUT s jednotlivými vstupy MIDI-In prvních nástrojů. MIDI-Thru prvních přístrojů potom s dalšími vstupy MIDI-In.

Tento způsob zapojení se v elektronické hudbě s oblibou používá. Jeden vysílač (Master-Keyboard nebo Master-Computer) řídí připojené nástroje. Časové zpoždění dané hardwarem při předávání vstupních dat MIDI-In na MIDI-Thru si vynucuje použití speciálního děliče (MIDI-Thru-Box). Pokud by bylo za sebou zapojeno bez ošetření více nástrojů, byly by problémy správného timingu akusticky pozorovatelné. Mezi úderem na klávesu na Master-Keyboard a zvukem syntetizéru na konci přenosového řetězce by došlo k časovým rozdílům. Při přenosu dat mezi mikropočítači se dá zpoždění, mimochodem činí při zařazení desítky počítačů zlomky sekund, akceptovat.

Předtím než člověk začne tvořit program pro MIDI, měl by se seznámit s příslušnou normou. Data se přenášejí seriově. Pro rozeznání dat se na začátku přenosu přenáší tzv. status-byte. Zásadně u něj bývá nastaven nejvyšší bit (bit 7). Potom může následovat libovolné množství datových bajtů. U nich je vždy bit 7 vymazán. Datové bajty se dají rozdělit do tří hlavních skupin:

1. Channel Information

- je řídicí informace a obsahuje v horních čtyřech bitech (horní nibble) řídicí slovo. Ve čtyřech méně význam-

ných bitech následuje číslo kanálu (0 až 15 odpovídá kanálům 1-16). Protože tato informace o kanálu se vyhodnocuje v připojeném přístroji, dá se řídit nezávisle na sobě až 16 nástrojů. Standardně rozumí všechny přístroje s MIDI následujícím příkazům:

\$9X-144+kanál(X) - nota zap

Této informaci následují dva bajty:

1. bajt:\$00 až \$7F(127) - Key

Prostřední „C“ má číslo 60. Všechny klávesy C odpovídají násobení 12.

2. bajt:\$00 až \$7F(127) - Velocity

1=lehce, 127=silně. Při „0“ je nota vypnuta. Klávesnice bez dynamiky vysílá vždy 64.

\$8X-128+kanál(X) - nota vyp

...následují opět dva bajty:

1. bajt:\$00 až \$7F(127) - Key

2. bajt:\$00 až \$7F(127) - Release

...odpovídá rychlosti při uvolnění klávesy.

SAX-160+kanál(X) - stisk klávesy

...také zde následují dva bajty:

1. bajt:\$00 až \$7F(127) - Key

2. bajt:\$00 až \$7F(127) - Aftertouch

... odpovídá síle stisku klávesy. Po stisku klávesy se jím dá ovlivnit síla tónu. Většina klávesnic tento parametr nevysílá.

\$BX-176+kanál(X) - změna parametrů

...opět následují dva bajty:

1. bajt:\$00 až \$7F(127) - příkaz

... je specificky stanoven výrobcem, definovány jsou pouze čtyři hodnoty:

124 dálkové ovládání zap/vyp

125 Omni-mód zap, všechny noty vyp

126 Mono-mód zap, všechny noty vyp

127 Poly-mód zap, všechny noty vyp

2. bajt:\$00 až \$7F(127) - hodnota parametru

Pro příkazové módy je tato hodnota vždy nulová, při dálkovém ovládání znamená 0=dálkově ovládání, 127=interní ovládání.

SCX-192+kanál(X) - změna programu

... následuje jeden bajt:

\$00 až \$7F(127) - číslo programu

SDX-208+kanál - klávesa (mono)

... i zde následuje jeden datový bajt:

\$00 až \$7F(127) - After-touch v režimu mono

SEX-224+kanál(X) - výška tónu (Pitch-Wheel)

...zde následují opět dva bajty

1. bajt:\$00 až \$7F(127) MSB změna
 2. bajt:\$00 až \$7F(127) LSB změna
- Oba bajty dávají dohromady 14 bitovou hodnotu, která zrcadlí polohu kola. Střední poloha je „\$00 \$40“.

2. System Exclusive - \$F0

...jsou specifické příkazy výrobce. Počet datových bajtů zde není předepsán. Řídící slovo (první bajt,\$F0=240) zavádí postup, druhé (0 až 127) obsahuje rozpoznání výrobce. Pokud přijímač není rozpoznán, data se nenačtou (pokud bit 7 není nastaven). Libovolné řídicí slovo (bit 7 nastaven) uzavírá přenos dat.

3. System Realtime

...slouží k řízení počítače, sekvenceru atd. v reálném čase. Bajty představují jednobajtové příkazy. Mohou být vyřízeny vždy, tedy i mezi dvěma datovými bajty. Platí pro všechny kanály.

SF8(248) - Clock

...udává takt 24x za čtvrt noty

SF9(249) - konec

...nahrazuje hodiny na konci sekvence

SFA(250) - start

...objeví se na začátku sekvence. Hodinový impuls musí následovat během 5 milisekund.

SFB(251) - pokračování

...je vyslán při pokračování přerušené sekvence.

SFC(252) - sync

...objevuje se během přestávek a stará se o pokračování timingu.

4. System Common

...platí nezávisle na nastaveném kanále

SF2(242) - takt/číslo sekvence

...následují dva datové bajty

1. bajt:\$00 až \$F7(127)- MSB číslo sekvence

2. bajt:\$00 až \$F7(127)- LSB číslo sekvence

SF3(243) - číslo partitury

...opět následují dva bajty

1. bajt:\$00 až \$F7(127) MSB

2. bajt:\$00 až \$F7(127) LSB

Oba bajty dávají dohromady 14 bitovou hodnotu, která udává číslo následující písně.

SF6(246) - hlas

...způsobí samourčení hlasu připojeného nástroje.

5. System Reset - \$FF

...nastavuje všechny připojené přístroje zpět do výchozího stavu. Při použití tohoto příkazu pracujte s rozmyslem.

Výhody MIDI leží nejen ve spojení různých hudebních nástrojů, ale v možnosti jejich řízení počítačem. Teprve potom může i amatér nechat bez velkých investic zaznít kompletní orchestr. Pozoruhodné je, že celá věc je realizovatelná i na C64. Nezkusíte to i vy? Napište! Rádi vaše programy uveřejníme.

(JV)

AMIGA 600/1200



Obr. Amiga 600

V loňském roce se na trhu objevil nový počítač řady AMIGA, počítač AMIGA 600. Koncem roku pak dodal Commodore do ČSFR první Amigy 1200. Málokdo měl příležitost je vidět, nalož blíže poznat, proto neuškodí, když si o těchto počítačích uvedeme pár podrobností.

AMIGA 600

Amiga 600 je dodávána v provedení bez HD, s HD 20MB a HD 40MB. Hard disky mají průměr 2,5" a rozměr cca 8x8 cm. Jsou připojené na vnitřní konektor počítače, takže zvenku verzi s HD od obyčejné A600 nerozeznáte. A uvažte, že ve verzi 2,5" jsou již dnes dostupné HD s kapacitou až 100 MB! Takt procesoru A600 je zase 7 MHz jako u A500. Počítač je vybaven 1 MB Chip-RAM, AGNUSem 8375, který spolupracuje s 2 MB Chip-RAM, ECS-Denise a interním prostorem pro zabudování Hard disku. Dále se nevyznačuje žádnými zvláštními novinkami. Procesor 68000, nejmenší z rodiny 16/32 bitových procesorů Motorola, je dnes

pro domácí počítač spíše málo výkonný. Jistou výhodou nového počítače je zabudovaný TV modulátor s podstatně vyšší kvalitou než je běžný modulátor A520HF. Lze také připojit monitor s digitálním RGB vstupem. Podstatnou změnou proti řadě A500 je nepřítomnost DMA konektoru, pro který byla u předchozí řady vyvinuta řada periférií, které teď zůstanou bez užitku. Místo tohoto konektoru má A600 na levé boční straně výřez, označený MEMORYCARD. Výřez s konektorem kdesi uvnitř je určen pro připojení paměťové karty 1MB CHIP RAM, kterou plně využívá nový AGNUS. Ostatní konektory jsou vlastně shodné s A500. Jednu podstatnou novinku však hardwaru A600 musíme přiznat. Počítač je konečně vybaven disketovou mechanikou CHINON, která je nesrovnatelně spolehlivější než mechaniky Panasonic-Matsushita, používané u A500+.

Pohled zvenku

Počítač působí vedle Amigy 500/500+ jako drobeček, nemá totiž numerický blok. Díky tomu je pouzdro počítače podstatně kratší a vzhledem k tomu, že deska počítače je osazována metodou povrchové montáže, je i hloubka počítače menší. Uvážíme-li že v provedení HD jde o počítač rovnocenný minimálně PC/XT, je počítač opravdu kompaktní. Takto koncipovaná konstrukce je samozřejmě náchylná k rušení, proto po otevření počítače najdeme složité stínění z perforovaného plechu, kryjící prakticky celou desku počítače. Zdá se, že stínění i proti statické byla u počítače věnována velká pozornost, což potvrzují první uživatelské zkušenosti. Čas ukáže, zda výsledek odpovídá vynaložené snaze.

Malý a kompaktní tvar počítače je podtržen skoro bílou barvou obalu s tmavšími klávesami. Mnoho uživatelů však bude tímto provedením zklamáno, pro mnoho softwarových produktů na A500, které budou chtít na A600 také provozovat sloužil totiž numerický blok k ovládní.

Systémový software

V počítači je 512 KB EPROM s Kickstartem 2.0. Také dodávaný Workbench je poněkud novější verze číslo 2.05. Dlouho očekávaná verze Superworkbench 2.1 se tedy zatím nekoná. Není se co divit, když byla loni v létě teprve ve stádiu Gama-fáze. Nicméně nová tvář Workbenche 2.04 se líbí a verze 1.3 vedle ní začíná vypadat jako chudý příbuzný. Je však pro objektivitu nutno přiznat, že rozlišovací schopnost 768 x 1024 bodů kterou oplývá Amiga 600 tvorbu graficky zdařilého prostředí Workbenche 2.04 značně usnadnila. Prostě, standardních 320 x 400 bodů Workbenche 1.3 je dnes již žalostně málo.

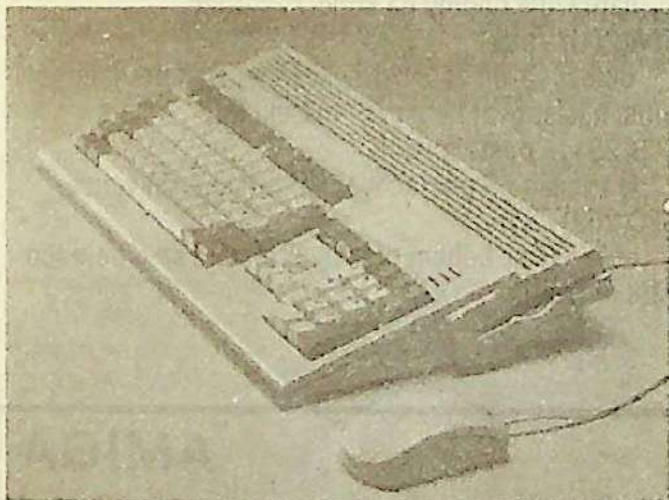
Co se nelíbí

Disketová jednotka 3,5" je na každém PC opatřena pérovou krytkou, omezující přístup prachu a nečistot.

Proč tuto sériovou maličkost Commodore na počítači nezavedl, je těžko pochopitelné. Pohled na "vnitřnosti", výřezem pro disketu nepůsobí dobře z mnoha důvodů. Procesor 68000 7MHz spolu s grafickými módy SUPERHIRES a PRODUCTIVITY jsou nelogickou kombinací. Procesor je v těchto módech zabrzděn na cca 10% a dovedete si představit Amigu 0,7 MHz? Nový typ rozhraní (MEMORYCARD) způsobí pomalý rozvoj přídatných karet, je opravdu škoda, že počítač nemá praktický DMA konektor řady A500, který by umožnil beze změn použít řadu karet řady 500.

Závěr

Amiga 600 byla dlouho očekávána a zvláště ve výkonných parametrech nepřinesla změny, které si od ní nedočkaví fandové slibovali. Kritikové o ní proto prohlásili, že by měla být spíše označena A300. Těm, kteří považují výpočetní výkon A600 za nedostatečný, doporučujeme AMIGU 1200, která přinesla i v tomto bodě proti řadě 500 zlepšení.



AMIGA 1200 - NOVÁ VLAJKOVÁ LOŤ COMMODORE?

Amiga 1200 s procesorem 68020 pracujícím na 14 MHz kmitočtu představuje konečně výkonnější počítač v porovnání s řadou 500. Spolu s rozšířením grafických módů se tak A1200 stává supervýkonnou grafickou stanicí. Dodává se ve dvou provedeních, jednak bez HD, jednak s HD, standardně 40 MBa s výhledem až do 200 MB. Vestavěná disketová jednotka 3,5" dokáže číst a zapisovat kromě disket ve formátu Amiga-DOS také diskety 720 kB pod MS-DOSem. Počítač je standardně vybaven 2MB Chip-RAM. Také dodávaná myš je inovována, minimálně tvarově. Amiga 1200 má na levé straně problematický konektor MEMORYCARD. Snad i pod dojmem kritiků je A1200 vybavena klávesnicí, která odpovídá té, na kterou si uživatelé A500 zvykli, se samostatným numerickým blokem.

Ve vnějším porovnání s A600 je A1200 jen o něco delším příbuzným ve stejné barvě, tvarovém provedení, konektorech. Jen ten numerický blok ji staví v porovnání s A600 do lepšího světla. A praktický provoz

namí prozradí, že počítač je neporovnatelně rychlejší, než A500, což ocení zvláště uživatelé, pracující s grafikou.

Závěr

Novou řadou počítačů A600/1200 zvoní staré dobré Amize 500 umíráček. Commodore se i v tomto případě zachoval tradičně. Pouze částečná kompatibilita nové a staré řady, problémy kolem Kickstartu 1.3 a 2.0, opožděné uvedení WORKBENCHE 2.1 a vlastně provi-

kompatibility programů. Uživatelé A600 se domáhají software pro A600, který vlastně neexistuje a jsou nešťastní, že jim nikdo neporadí, který software se pro ně hodí a který nikoliv. A co teprve majitelé A1200! Ale tak je to vždy při zavádění novinek. Teprve léta praktického provozu ukáží, nakolik byl záměr techniků firmy Commodore uvážení a životaschopný. Byla by škoda, kdyby se měla opakovat historie řady C16/P4.

(JK)

COMOTRONIC NEWS

KONEČNĚ MOŽNOSTI HODNOTNÉ VÝHRÝ!

Slosování zákazníků

S platností od 1. 1. 1993 budou každé čtvrtletí slosování všichni zákazníci, kteří u firmy Comotronic nakoupili v daném období zboží za více než 500,- Kč. Výsledky slosování budou vždy uveřejněny v časopise FUN with Commodore, ceny budou zákazníkům předány osobně, pokud nebudou moci přijet zašleme je poštou.

1. cena: DISKETOVÁ JEDNOTKA VC 1541-II
2. cena: CARTRIDGE MAKER 256kB (EPROM DISK)
3. cena: MYŠ VC 1351 pro C64/128

Slosování předplatitelů FUNu

Obdobné slosování čeká předplatitele FUNu, kde budou slosování všichni celoroční předplatitelé vždy 2x za rok (červen, prosinec) s následujícími cenami:

1. cena: MYŠ VC 1351 pro C64/128
2. cena: FINAL CARTRIDGE III
3. cena: 2x JOYSTICK PYTHON 1M

Výsledky slosování budou uveřejněny v časopise FUN a ceny zaslány výhercům poštou, pokud si nebudou moci vyzvednout výhry osobně.

SUPERCENY - OD 01.03.93

Konečně se ceny pořádně hnuly a dokonce směrem dolů!

Po prvním šoku, který vyvolalo zavedení DPH, nových pojišťovacích zákonů a odvodů ze zisku, jsme i my museli sáhnout k úpravám cen. Tak došlo k obecnému zvýšení cen našich výrobků o 8% s výjimkou těch,

kteří byly ovlivněny změnou cen našich dodavatelů, kteří dovozové zboží většinou slevňovali. Výsledkem dnes již dvouměsíčního pohybu cen je nakonec pronikavé snížení cen základního hardware a některých dalších výrobků. Základní přehled Vám dá následující tabulka. Samozřejmě bude snahou firmy dosáhnout loňské cenové úrovně i u ostatních výrobků, lze totiž očekávat, že i naši dodavatelé součástek a tiskovin postupně ceny sníží na loňskou úroveň.

| | |
|-----------------------|------------|
| Commodore 64-II | 4130,- Kč |
| Video Game Set | 4030,- Kč |
| Magnetofon 1530 | 875,- Kč |
| Disk 1541-II | 4880,- Kč |
| Tisk. MPS 1230 | 7600,- Kč |
| Monitor 1085S | 10170,- Kč |
| Monitor 1084S | 10900,- Kč |
| Amiga 500 | 13500,- Kč |
| Amiga 600 | 13990,- Kč |
| Amiga 600HD20 | 20250,- Kč |
| Amiga 600HD40 | 23920,- Kč |
| Amiga 1200 | 22250,- Kč |
| CDTV, 1MB RAM + 1085S | 24240,- Kč |
| Kabel Centronics - UP | 200,- Kč |
| Zdroj C64 | 864,- Kč |
| Kabel Centronics A500 | 99,- Kč |
| Plexi kryt C64 | 260,- Kč |
| Kazeta C60 RAKS | 22,- Kč |

NOVINKY V SORTIMENTU

Diskety

Byla rozšířena nabídka disket pro C64 i Amigu. Jedním z důvodů bylo zrušení výroby disket Verbatim Verex a silící poptávka po jiných typech levnějších disket. Proto je v současné době nabízen následující výběr disket:

| | |
|-----------|-------|
| NoName | 100,- |
| Commodore | 180,- |
| Nashua | 175,- |

Diskety 3,5" DD po 10 kusech

| | |
|-----------|----------------------|
| NoName | 210,- |
| V.Verex | 255,- (do vyprodání) |
| Commodore | 260,- |

Joysticky

V našem katalogu nabízený joystick SV 124 není dodáván již více než půl roku. Místo něj byl dosud dodáván joystick QuickShot 2Plus, provedením na 90% shodný. I u toho však dochází v poslední době k problémům s dodávkami, neboť výroba se údajně zastavila. Proto jsme hledali náhradu za uvedené mikrosplnačové joysticky a přitom jsme diskutovali otázku joystick mikrosplnačový-gumový. Zástupcem výrobce jsme byli přesvědčováni, že joysticky s kontakty z vodivé gumy jsou dnes již stejně spolehlivé (trvanlivé) jako mikrosplnačové, proto jsme nabídku doplnili o tento typ. V současné době nabízíme tento sortiment:

| | | |
|--------------------|----------------|----------|
| Joystick Apache 1 | (guma) | 205,- Kč |
| Joystick QS-2Plus | (mikro) | 259,- Kč |
| | (do vyprodání) | |
| Joystick QS 111A | (mikro) | 280,- Kč |
| Joystick Python 1M | (mikro) | 320,- Kč |

Konečně ekonomický HD pro Amigu!

Pro majitele Amigy 500/500+ máme příjemnou zprávu, že je pro ně vyřešeno připojení Hard disků IDE (AT bus) o kapacitě 40 - 210MB na jejich počítač. Tím se snižuje cena HD pro A500 prakticky o 50%! Zázrakem, který tuto možnost přináší je nového řadič C211, který je dodatečný od března 1993. Řadič je v plastové krabici, do které se umístí i vlastní HD, připojuje se na rozšiřovací konektor bez nutnosti rozebírat počítač. Spolupracuje s počítači, používajícími Kickstart 1.3 a 2.0. Cena řadiče s kabely, systémovou disketou a návodem je stanovena na 4099,- Kčs.

(JK)

OBJEDNÁVKA PŘEDPLATNÉHO ČASOPISU FUN 1993

Předplatné časopisu minimálně na 5 čísel Vám umožní ušetřit na jednom výtisku minimálně 5,- Kč.

Pro předplatitele je cena stanovena na 10,- Kč za 1 výtisk včetně poštovného. Pro ostatní zájemce je cena jednoho výtisku 15,- Kč.

Z organizačních důvodů zachováváme i letos shodnou formu placení předplatného. Předplatné se platí složenkou. Kontrolní ústřížek případně jeho kopii přiložte prosím k objednávce. S ohledem na zavedený způsob evidence žádáme předplatitele, aby respektovali tento postup a nevyžadovali telefonické zajištění předplatného.

Adresa redakce: Dolnomlýnská 2, 787 01 Šumperk

Autoři čísla: Jiří Kouřil a Jaroslav Vančura

Podávání poštovních novinových zásilek povoleno Oblastní správou pošt v Ostravě č.j.: 2882/92-P1 ze dne 14.12.1992

FUN - časopis Comotronic klubu pro uživatele počítačů Commodore 64/128. Druhý ročník, první číslo.

Fotosazba J. Kotinský • Tisk: Grafotyp Šumperk

OBSAH

| | |
|------------------------------|------------------|
| Úvodní slovo | 1 |
| Dopisy čtenářů | 2 |
| Basic | 3 |
| Assembler na C64 | 6 |
| Tipy a triky | 10 |
| DYCP grafika | 12 |
| RS 232 | 13 |
| PC klávesnice pro C64 | 16 |
| Představujeme vám: | |
| Cartridge maker 256 kB | 18 |
| MIDIinterfejs pro C64 | 20 |
| Amiga 600 a 1200 | 21 |
| Comotronic News | 23 |
| Inzerce | 3. strana obálky |

CO BUDE V PŘÍŠTÍM ČÍSLE?

Basic - pokračování

Assembler - pokračování

Tipy a triky - pokračování

Centronics na C64

DYSP - grafika

Dopisy čtenářů, inzerce aj.

Představujeme:

FCIII a ty ostatní

Jednoduché přístroje - R/C metr

Superfloppy pro C64

Comotronic News

COBAM

CO BUDE V PŘÍŠTĚM ČÍSLE



COMOTRONIC

Vydává Comotronic klub Šumperk